

# Resumen

El análisis de imagen multiespectral es una técnica que implica capturar imágenes en diferentes longitudes de onda, incluyendo aquellas que están más allá de lo visible para el ojo humano. Esto proporciona una vista más completa del entorno, revelando patrones y cambios que pueden no ser evidentes a simple vista.

La segmentación semántica es un método que permite dividir una imagen en segmentos o regiones distintas según su contenido, lo cual puede lograrse utilizando algoritmos de aprendizaje profundo entrenados con conjuntos de datos grandes. Esta técnica proporciona una comprensión detallada del entorno, con cada segmento ofreciendo información específica sobre los objetos y características dentro de él, como diferentes tipos de vegetación o cuerpos de agua. Una de las ventajas más significativas de esta técnica es su capacidad para detectar el entorno. Al comparar imágenes tomadas en diferentes momentos, es posible identificar cambios en la cobertura de tierra, la salud de la vegetación, la integridad de los objetos y la contaminación, entre otras cosas. Al identificar tendencias y problemas potenciales como la deforestación o la contaminación, esta información puede ayudar en la toma de decisiones relacionadas con la gestión ambiental, incluyendo la identificación de áreas que requieren esfuerzos de protección o conservación. Este enfoque también ayuda a apoyar el desarrollo sostenible y permite mejores procesos de toma de decisiones.

La escalabilidad del análisis de imagen multiespectral y la segmentación semántica es otra ventaja. Por ejemplo, con la disponibilidad de imágenes de satélite multiespectrales de alta resolución, ahora es posible monitorizar grandes áreas del entorno simultáneamente, lo que facilita la identificación de tendencias y patrones que pueden no ser evidentes a escala local.

Esta Tesis Doctoral contribuye al avance en el campo de la monitorización medioambiental mediante el análisis de imagen multiespectral y la segmentación semántica de múltiples formas:

- Se lleva a cabo una revisión exhaustiva de los trabajos más relevantes y recientes en el área del análisis de imagen multiespectral y la segmentación semántica con el fin de conocer el estado del arte y las tendencias actuales en el campo de la monitorización medioambiental.

- 
- Se presenta y valida un método para identificar distintos tipos de cultivos a partir de imágenes aéreas. También se producen y publican dos conjuntos de datos inéditos para esta tarea. Uno con imágenes obtenidas por aviones (UOPNOA) y otro con imágenes de satélite (UOS2).
  - Se desarrolla y verifica un método para localizar campos que han sido abonados recientemente. Asimismo, se genera y comparte un nuevo conjunto de datos.
  - Se diseña y comprueba un método para encontrar defectos subsuperficiales en láminas de polímero con fibra de carbono. Igualmente, se crea y publica un nuevo conjunto de datos.
  - Se elabora y confirma un método para detectar, ubicar y caracterizar emisiones fugitivas en plantas industriales usando cámaras de videovigilancia.

En resumen, esta Tesis Doctoral desarrolla y valida un enfoque innovador de análisis de imagen multiespectral y segmentación semántica para diversas aplicaciones reales de monitorización medioambiental. Esta técnica proporciona información precisa y detallada sobre el entorno, facilitando una mejor toma de decisiones y garantizando la eficiencia y la sostenibilidad de los recursos naturales. Además, se crean y publican nuevos conjuntos de datos públicos para el entrenamiento y evaluación de algoritmos, lo que impulsa el avance científico y tecnológico en el campo del análisis de imagen multiespectral y la segmentación semántica en estos sectores.

# Abstract

Multi-spectral imaging is a technique that involves capturing images at different wavelengths, including those beyond what is visible to the human eye. This provides a more comprehensive view of the environment, revealing patterns and changes that would otherwise not be evident.

Semantic segmentation is a method of dividing an image into distinct segments or regions based on their content, which can be accomplished using deep learning algorithms trained on large datasets. This technique provides a detailed understanding of the environment, with each segment offering specific information about the objects and features within it, such as different types of vegetation or bodies of water. One of the most significant advantages of this technique is its ability to detect the environment. By comparing images taken at different times, it is possible to identify changes in land cover, vegetation health, the integrity of objects, and pollution, among other things. By identifying trends and potential problems like deforestation or pollution, this information can inform decisions related to environmental management, including identifying areas that require protection or conservation efforts. This approach also helps to support sustainable development and enables better decision-making processes.

The scalability of multi-spectral imaging and semantic segmentation is another advantage. For example, with the availability of high-resolution multi-spectral satellite images, it is now possible to monitor large areas of the environment simultaneously, making it easier to identify trends and patterns that may not be apparent at a local scale.

This Doctoral Thesis contributes to the advancement in the field of environmental monitoring through multispectral image analysis and semantic segmentation in multiple ways:

- A comprehensive review of the most relevant and recent works in the area of multispectral image analysis and semantic segmentation is carried out in order to know the state of the art and the current trends in the field of environmental monitoring.
- A method for identifying different type of crops from aerial images is presented and validated. Two new datasets for this task are also produced and published. One with images obtained by airplanes (UOPNOA) and another with satellite images (UOS2).

- 
- A method for locating fields that have been recently fertilized is developed and verified. Likewise, a new dataset is generated and shared.
  - A method for finding subsurface defects in polymer laminates with carbon fiber is designed and tested. Similarly, a new dataset is created and published.
  - A method for detecting, locating and characterizing fugitive emissions in industrial plants using video surveillance cameras is developed and confirmed.

In summary, this Doctoral Thesis develops and validates an innovative approach of multispectral image analysis and semantic segmentation for various real applications of environmental monitoring. This technique provides precise and detailed information about the environment, facilitating a better decision making and ensuring the efficiency and sustainability of natural resources. In addition, new public datasets are created and published for the training and evaluation of algorithms, which drives the scientific and technological advancement in the field of multispectral image analysis and semantic segmentation in these sectors.

# Índice general

Índice de figuras	IX
Índice de tablas	XIII
<b>1 Introducción</b>	<b>1</b>
1.1 Motivación y objetivos . . . . .	3
1.2 Contexto . . . . .	4
<b>2 Análisis, procesamiento y segmentación de imágenes multiespectrales</b>	<b>7</b>
2.1 Análisis de imagen multiespectral . . . . .	8
2.1.1 Sensores multiespectrales . . . . .	9
2.1.2 Satélites de imagen multiespectral . . . . .	10
2.1.3 Preprocesamiento de los datos . . . . .	11
2.2 Clasificación de regiones . . . . .	12
2.2.1 Métodos tradicionales . . . . .	12
2.2.1.1 <i>K-Nearest Neighbors</i> . . . . .	12
2.2.1.2 <i>Naive Bayes</i> . . . . .	13
2.2.1.3 <i>Random Forest</i> . . . . .	14
2.2.1.4 <i>Support-Vector Machines</i> . . . . .	15
2.2.1.5 Redes Neuronales Artificiales . . . . .	15
2.2.2 Métodos basados en aprendizaje profundo . . . . .	16
2.2.2.1 Detección de objetos . . . . .	17
2.2.2.2 Segmentación semántica . . . . .	18
2.2.2.3 Segmentación de instancias . . . . .	19
2.2.2.4 Segmentación panóptica . . . . .	20
2.3 Diseño y configuración de las redes de segmentación semántica . . . . .	21
2.3.1 Entrenamiento . . . . .	21
2.3.2 Métricas . . . . .	22
2.3.3 Hiperparámetros . . . . .	24
2.3.4 Arquitecturas . . . . .	25
2.3.4.1 UNet . . . . .	25
2.3.4.2 DeepLabV3+ . . . . .	27
2.4 Monitorización medioambiental . . . . .	28
2.4.1 Agricultura de precisión . . . . .	28
2.4.2 Detección de defectos . . . . .	30

2.4.3	Detección y estimación de emisiones fugitivas contaminantes	31
<b>3</b>	<b>Experimentación e interpretación de resultados</b>	<b>33</b>
3.1	Agricultura de precisión	34
3.1.1	Reconocimiento de cultivos	35
3.1.1.1	Resultados y discusiones	39
3.1.2	Detección de campos recientemente abonados	51
3.1.2.1	Resultados y discusiones	59
3.2	Defectos subsuperficiales	65
3.2.1	Generación del conjunto de datos	68
3.2.1.1	Termografía infrarroja mediante calentamiento gradual	68
3.2.1.2	Preprocesado de los datos	71
3.2.1.3	Conjuntos de datos	72
3.2.2	Resultados y discusiones	74
3.2.2.1	<i>Random Forest</i> y <i>Support Vector Machines</i>	74
3.2.2.2	UNet	78
3.2.2.3	DeepLabV3+	82
3.2.3	Comparación de resultados	85
3.2.3.1	Validación de los resultados	86
3.3	Emisiones fugitivas en plantas industriales	91
3.3.1	Detección de emisiones	94
3.3.1.1	Emisiones diurnas	94
3.3.1.2	Emisiones nocturnas	102
3.3.1.3	Cámaras infrarrojas	103
3.3.1.4	Resultados y discusiones	103
3.3.2	Estimación de la opacidad	108
3.3.2.1	Solución propuesta	108
3.3.2.2	Algoritmos de estimación	109
3.3.2.3	Resultados y discusiones	115
<b>4</b>	<b>Conclusiones</b>	<b>121</b>
4.1	Trabajo futuro	122
<b>5</b>	<b>Contribuciones</b>	<b>125</b>
5.1	Publicaciones: JCR (Journal Citation Reports)	125
5.1.1	Evaluation of Semantic Segmentation Methods for Land Use with Spectral Imaging Using Sentinel-2 and PNOA Imagery	125
5.1.2	Semantic segmentation for non-destructive testing with step-heating thermography for composite laminates	159

5.1.3	Detection and localization of fugitive emissions in industrial plants using surveillance cameras . . . . .	175
5.1.4	Fully automated method to estimate opacity in stack and fugitive emissions: A case study in industrial environments	190
5.1.5	Remote sensing for detecting freshly manured fields . . . . .	203
5.2	Publicaciones en revisión: JCR (Journal Citation Reports) . . . . .	220
5.2.1	Evaluation of remote sensing spectral indices for manure application in pasture fields . . . . .	220
5.3	Publicaciones: JCI ( <i>Journal Citation Indicator</i> ) . . . . .	220
5.3.1	Satellite imagery dataset of manure application on pasture fields . . . . .	220
5.4	Otras contribuciones . . . . .	231
5.4.1	UOPNOA and UOS2 datasets for aerial crop classification [Conjuntos de datos] . . . . .	231
5.4.2	Dataset for semantic segmentation in NDT with step-heating thermography for CFRP laminates [Conjuntos de datos] . . . . .	231
5.4.3	Satellite imagery dataset of manure application on pasture fields [Conjuntos de datos] . . . . .	231
<b>A</b>	<b>Índices multiespectrales de vegetación</b>	<b>233</b>
	<b>Bibliografía</b>	<b>239</b>





# Índice de figuras

2.1	Ejemplo de <i>K-Nearest Neighbors</i> con $K$ igual a tres para dos clases.	13
2.2	Ejemplo de <i>Naive bayes</i> para dos clases. . . . .	13
2.3	Estructura de <i>Random Forest</i> . . . . .	14
2.4	Ejemplo de <i>Support-Vector Machines</i> para dos clases. . . . .	15
2.5	Estructura de una Red Neuronal Artificial con dos capas ocultas.	16
2.6	Ejemplo de detección de objetos. . . . .	17
2.7	Ejemplo de segmentación semántica. . . . .	18
2.8	Ejemplo de segmentación de instancias. . . . .	19
2.9	Ejemplo de segmentación panóptica. . . . .	20
2.10	Verdadero positivo, falso positivo, verdadero negativo, y falso negativo. . . . .	23
2.11	Arquitectura de UNet. . . . .	26
2.12	Arquitectura de DeepLabV3+. . . . .	28
3.1	Imagen recortada y comprobación de la máscara objetivo para UOPNOA. (a) imagen recortada de $256 \times 256$ píxeles de una imagen PNOA, (b) visor SIGPAC de la misma región para verificar la máscara, (c) máscara objetivo. . . . .	35
3.2	Tamaño en píxeles de cada clase de UOPNOA. . . . .	36
3.3	Imagen recortada y comprobación de la máscara de referencia para UOS2. (a) imagen recortada de $256 \times 256$ píxeles de una imagen de Sentinel-2, (b) visor SIGPAC de la misma región para verificar la máscara, (c) máscara de referencia. . . . .	37
3.4	Tamaño en píxeles de cada clase de UOS2. . . . .	38
3.5	Visualización de los resultados para DeepLabV3+ evaluado en UOPNOA (Experimento Base). (1 <sup>a</sup> col.) Imágenes originales, (2 <sup>a</sup> col.) máscaras de referencia, (3 <sup>a</sup> col.) predicciones, (4 <sup>a</sup> col.) superposición con máscaras de referencia, (5 <sup>a</sup> col.) superposición con predicciones. . . . .	45
3.6	Visualización de los resultados para UNet en UOS2 (Experimento Base). (1 <sup>a</sup> col.) Imágenes originales, (2 <sup>a</sup> col.) máscaras de referencia, (3 <sup>a</sup> col.) predicciones, (4 <sup>a</sup> col.) superposición con máscaras de referencia, (5 <sup>a</sup> col.) superposición con predicciones. . . . .	46

3.7	Visualización de los resultados para UNet en UOS2 para cuatro clases simplificadas (Experiment Base). ( <b>1<sup>a</sup> col.</b> ) Imágenes originales, ( <b>2<sup>a</sup> col.</b> ) máscaras de referencia, ( <b>3<sup>a</sup> col.</b> ) predicciones, ( <b>4<sup>a</sup> col.</b> ) superposición con máscaras de referencia, ( <b>5<sup>a</sup> col.</b> ) superposición con predicciones. . . . .	47
3.8	Rendimiento del procesamiento de imagen en diversas plataformas. . . . .	49
3.9	Rendimiento de coste por infraestructura para diez imágenes. Los precios y la potencia de cálculo están sujetos a cambios (abril de 2021). . . . .	50
3.10	Ejemplo de etiquetado (P-VG1). . . . .	54
3.11	Ejemplo de máscaras de referencia (P-VG1). . . . .	54
3.12	Respuesta de los índices de vegetación antes y después de la aplicación de estiércol. . . . .	56
3.13	Valores de intensidad de las bandas de Sentinel-2 para la serie temporal de las parcelas P-CLGT, P-FNFR, P-LNDRS1, P-SNVCNT2, P-TGL2, y P-VG1. . . . .	61
3.14	Detecciones del conjunto de pruebas. La columna izquierda es la ubicación de la verdad terrestre y la columna derecha es la máscara de detección. P-CLGT en la primera fila, P-LNDRS1/2/3/4 en la segunda fila y P-LLT y P-MT en la tercera fila. . . . .	64
3.15	Fotografías del espécimen 1. . . . .	66
3.16	Dimensiones y localización de los defectos del espécimen 1. . . . .	67
3.17	Configuración para la grabación. . . . .	68
3.18	Señales de intensidad de calentamiento y enfriamiento para la secuencia temporal. . . . .	69
3.19	Píxeles de referencia con la muestra rotada 0°. . . . .	70
3.20	Píxeles de referencia con la muestra rotada 120°. . . . .	70
3.21	Ejemplo de una imagen de 30 canales. Las imágenes con etiqueta roja se obtienen de la secuencia de calentamiento. Las imágenes con etiqueta azul se obtienen de la secuencia de enfriamiento. . . . .	73
3.22	La visualización de los resultados predichos por <i>Random Forest</i> . ( <b>1ra col.</b> ) Imágenes originales, ( <b>2da col.</b> ) máscaras de la verdad terrena, ( <b>3ra col.</b> ) predicciones, ( <b>4ta col.</b> ) imágenes originales y máscaras de la verdad terrena, ( <b>5ta col.</b> ) imágenes originales con las predicciones. . . . .	76
3.23	Visualización de los resultados predichos por <i>Support Vector Machines</i> . ( <b>1ra col.</b> ) Imágenes originales, ( <b>2da col.</b> ) máscaras de la verdad terrena, ( <b>3ra col.</b> ) predicciones, ( <b>4ta col.</b> ) imágenes originales con máscaras de la verdad terrena, ( <b>5ta col.</b> ) imágenes originales con predicciones. . . . .	77

3.24	Visualización de los resultados predichos para UNet evaluado con 3 canales. ( <b>1ra col.</b> ) Imágenes originales, ( <b>2da col.</b> ) máscaras de verdad terreno, ( <b>3ra col.</b> ) predicciones, ( <b>4ta col.</b> ) imágenes originales con máscaras de verdad terreno, ( <b>5ta col.</b> ) imágenes originales con predicciones. . . . .	80
3.25	Visualización de los resultados predichos para UNet evaluado con 30 canales. Solo se muestran los primeros tres canales en la imagen, que consisten en los componentes primero, tercero y cuarto, utilizando exactamente los mismos canales que las imágenes de 3 canales. ( <b>1<sup>a</sup> col.</b> ) Imágenes originales, ( <b>2<sup>a</sup> col.</b> ) máscaras de verdad terreno, ( <b>3<sup>a</sup> col.</b> ) predicciones, ( <b>4<sup>a</sup> col.</b> ) imágenes originales con máscaras de verdad terreno, ( <b>5<sup>a</sup> col.</b> ) imágenes originales con predicciones. . . . .	81
3.26	Visualización de los resultados predichos para DeepLabv3+. ( <b>1ra columna</b> ) Imágenes originales, ( <b>2da columna</b> ) máscaras de la verdad terrena, ( <b>3ra columna</b> ) predicciones, ( <b>4ta columna</b> ) imágenes originales con máscaras de la verdad terrena, ( <b>5ta columna</b> ) imágenes originales con predicciones. . . . .	84
3.27	<b>F<sub>1</sub>-Score</b> de de cada método. . . . .	85
3.28	Configuración del espécimen 2. . . . .	88
3.29	Configuración del espécimen 3. . . . .	89
3.30	Especímenes 2 (arriba) y 3 (abajo). . . . .	90
3.31	Estudio de umbrales (Plant1). . . . .	100
3.32	Estudio de umbrales (Plant2). . . . .	101
3.33	Mejor clasificación binaria de Plant1. . . . .	104
3.34	Mejor clasificación binaria de Plant2. . . . .	105
3.35	Mejor clasificación binaria de Plant3. . . . .	106
3.36	Mejor clasificación binaria de emisiones nocturnas con UNet para imágenes RGB. . . . .	107
3.37	Obtención de las regiones de interés (Cielo, emisión, y edificio) . . . . .	108
3.38	Escala de Ringelmann. . . . .	109
3.39	Modelo físico de <i>DOM</i> ( <i>modelo de transmisión</i> ). . . . .	110
3.40	División de la emisión en regiones horizontales. . . . .	110
3.41	Puntos de referencia para el método de <i>Yuen et al.</i> . . . . .	112
3.42	Modelo físico del método de Transmisión. . . . .	114
3.43	Extracción de la emisión utilizando la banda azul. . . . .	115
3.44	Emisiones de alta opacidad (Plant1). . . . .	115
3.45	Emisiones de baja opacidad (Plant1) . . . . .	116
3.46	Emisiones con alta luminosidad (Plant1) . . . . .	116
3.47	Emisiones de alta opacidad (Plant2). . . . .	117



# Índice de tablas

3.1	Clases de SIGPAC. . . . .	34
3.2	Clases del conjunto de datos UOPNOA. . . . .	36
3.3	Clases del conjunto de datos UOS2. . . . .	38
3.4	Métricas globales de cada método para el conjunto de datos UOP- NOA. . . . .	40
3.5	Métricas de cada clase de UNet en UOPNOA. . . . .	40
3.6	Métricas de cada clase de DeepLabV3+ en UOPNOA. . . . .	41
3.7	Comparativa del uso de la clase “Otros” en UOPNOA y UOS2. . . . .	41
3.8	Métricas globales para diferentes conjuntos de bandas con UNet en UOS2. . . . .	42
3.9	Métricas de cada clase para el mejor experimento de UNet en UOS2 (Base y Multiuso). . . . .	43
3.10	Métricas globales para clases simplificadas con UNet en UOPNOA. . . . .	44
3.11	Métricas de clase para clases simplificadas con UNet en UOS2. . . . .	44
3.12	Parcelas en el conjunto de datos. . . . .	53
3.13	Descripción de todos los conjuntos de características. . . . .	58
3.14	Resultados de cada conjunto de características. $mP=Precision$ media, $mR=Recall$ media, $mF_1=F_1-Score$ medio. . . . .	59
3.15	Resultados por método de clasificación para el experimento BA- 102-VI. $mP=Precision$ media, $mR=Recall$ media, $mF_1=F_1-Score$ medio. . . . .	62
3.16	Resultados por clase para el método de clasificación de Análisis Discriminante del experimento BA-102-VI. . . . .	62
3.17	Métricas para los experimentos con <i>Random Forest</i> y <i>Support Vector Machines</i> . . . . .	76
3.18	Parámetros de UNet. . . . .	78
3.19	Parámetros de entrenamiento para UNet. . . . .	78
3.20	Métricas para los experimentos con UNet. . . . .	79
3.21	Parámetros de DeepLabV3+. . . . .	82
3.22	Parámetros de entrenamiento para DeepLabV3+. . . . .	83
3.23	Métricas para el experimento con DeepLabV3+. . . . .	83
3.24	Métricas para todos los métodos. . . . .	85
3.25	Métricas para el espécimen 2. . . . .	90
3.26	Métricas para el espécimen 3. . . . .	90

3.27	Proporciones para los diferentes conjuntos de datos. (emisión:sin emisión) . . . . .	92
3.28	Hiperparametros para DeepLabV3+. . . . .	93
3.29	Métricas para los experimentos de clasificación multi-clase. . . . .	94
3.30	Métricas para los experimentos de clasificación binaria. . . . .	95
3.31	Métricas para los experimentos de clasificación binaria con ponderación de clases personalizada. . . . .	96
3.32	Métricas para los experimentos de proporción. . . . .	97
3.33	Métricas para los experimentos de test cruzado. . . . .	98
3.34	Métricas para el conjunto de entrenamiento reducido. . . . .	98
3.35	Métricas para los experimentos de transferencia de aprendizaje. . . . .	99
3.36	Métricas para la alarma. . . . .	101
3.37	Métricas de UNet y DeepLabV3+ para Día o Noche. . . . .	102
3.38	Métricas de Día y de Noche para UNet entrenado con el conjunto de datos Día-Noche. . . . .	102
3.39	Métricas para la comparación entre imágenes RGB y RGBI en Día y Noche. . . . .	103
3.40	$F_1$ -Score de los métodos para Plant1. . . . .	119
3.41	$F_1$ -Score de los métodos para Plant2. . . . .	119
A.1	Índices de vegetación mas usados en la agricultura de precisión y su orden en las imágenes multiespectrales generadas. . . . .	233

# Capítulo 1

## Introducción

En la actualidad, la conservación del medio ambiente es un tema de gran relevancia debido a la creciente preocupación por cuestiones como el cambio climático [2], la contaminación [15] y la disminución de la biodiversidad [18]. En este contexto, la tecnología puede jugar un papel importante para contribuir a la solución de estos problemas [60].

La monitorización y la gestión ambiental son áreas críticas para prevenir la degradación ambiental y conservar los recursos naturales [19]. Tradicionalmente, la monitorización del medio ambiente ha sido realizada de forma manual mediante métodos de muestreo y análisis de laboratorio, lo que resulta en una evaluación limitada debido a que los datos se recolectan en puntos específicos y, por lo tanto, no representan una imagen completa del área que se quiere monitorizar [129]. Además, los costes asociados con este método son altos y la velocidad de obtención de datos es baja. Por lo tanto, se requiere una solución que permita una evaluación más rápida y una toma de decisiones más informada.

En los últimos años, el uso de tecnologías basadas en la Inteligencia Artificial (IA) ha crecido de manera exponencial [139]. Hay varios factores que han contribuido a este aumento. Uno de ellos es la mejora en el rendimiento del hardware, especialmente en las GPUs (unidades de procesamiento gráfico) [119], que han permitido el entrenamiento de modelos de IA de manera más rápida y eficiente. Otro factor que ha impulsado el crecimiento de la IA es la aparición de nuevas técnicas y usos [3]. En los últimos años, se han desarrollado nuevas técnicas de aprendizaje máquina (*machine learning*, ML) y redes neuronales que han permitido la aplicación de la IA en una amplia variedad de campos, desde el análisis de datos hasta la automatización de procesos industriales [12], pasando por el reconocimiento facial [58] y la conducción autónoma [85].

La IA sigue manteniendo una trayectoria ascendente debido a su potencial para mejorar la eficiencia y la productividad en diferentes ámbitos. Esto ha llevado a que se invierta cada vez más en la investigación y el desarrollo de tecnologías basadas en IA [71]. También ha comenzado a integrarse en la vida cotidiana de las personas [138]. Desde asistentes virtuales hasta sistemas de recomendación en plataformas de compra en línea, la IA está cada vez más presente en

nuestra vida diaria. Esto ha contribuido a un aumento en popularidad e interés, impulsando su crecimiento.

El uso de la IA tiene una gran cantidad de aplicaciones en el campo medioambiental [115, 133, 134]. Por ejemplo, pueden ser utilizadas en la evaluación de la calidad del aire, la identificación de cambios en el uso del suelo, la detección de plagas y enfermedades en plantaciones, la identificación de cuerpos de agua, la identificación de especies vegetales y animales, y la reducción de residuos mediante mejoras en los procesos de fabricación. Además, el uso de estas técnicas permite obtener una evaluación más precisa, permitiendo a los operadores identificar los problemas más rápidamente y tomar medidas para abordarlos.

En particular, dos técnicas que están ganando cada vez más relevancia son el análisis de imagen multiespectral [87, 110] y la segmentación semántica [82, 135]. El análisis de imagen multiespectral consiste en la obtención de imágenes en diferentes longitudes de onda, lo que permite la extracción de información adicional de una imagen, como la reflectancia y la absorción de la radiación electromagnética de los diferentes materiales. Esto permite mejorar la visualización de diferentes propiedades de los elementos de una imagen, facilitando tareas de detección y localización de elementos de interés. La segmentación semántica es una técnica de procesamiento de imágenes basada en redes neuronales que consiste en identificar y separar elementos en una imagen. Esta segmentación se realiza a nivel de píxel, lo que permite una clasificación más precisa de los elementos en la imagen. Estas tecnologías son especialmente útiles para facilitar tareas que antes solo podían ser realizadas por humanos, como el análisis de imágenes para la monitorización de procesos industriales.

Este trabajo tiene como objetivo utilizar técnicas de análisis de imágenes para mejorar la monitorización medioambiental. Para ello, se emplean técnicas de segmentación semántica en imágenes multiespectrales para la detección y clasificación de elementos del entorno, con el fin de automatizar tareas beneficiosas para el medio ambiente. En este sentido, el trabajo evalúa la eficacia y precisión de la segmentación semántica frente a técnicas clásicas de aprendizaje máquina, como las *support-vector machines* (SVM) [24] o *random forest* (RF) [1], en aplicaciones medioambientales. La hipótesis que se va a evaluar es que la segmentación semántica aplicada a imágenes multiespectrales es una herramienta eficaz y precisa para el análisis de elementos del entorno y es adecuada para su monitorización constante. Su uso en aplicaciones medioambientales puede contribuir significativamente al conocimiento y la toma de decisiones en el ámbito medioambiental.

Para llevar a cabo la investigación, se han seleccionado varias tareas o aplicaciones concretas que podrían beneficiarse del análisis de imagen multiespectral y la segmentación semántica para su automatización, y así permitir su monitorización constante sin la intervención de un operador humano.



- El reconocimiento de diferentes tipos de cultivos mediante satélite. Esta tarea posibilita la monitorización y optimización del uso de recursos, lo que puede contribuir a reducir el impacto ambiental de la agricultura.
- La detección de campos recientemente abonados mediante satélite. Esto permite la monitorización y control del uso de fertilizantes, lo que puede contribuir a reducir la contaminación del agua y del suelo.
- La detección de defectos subsuperficiales en láminas de polímero reforzado con fibra de carbono (*Carbon fiber-reinforced polymers*, CFRP) mediante el uso de cámaras termográficas. Esta tarea ayuda a garantizar la seguridad estructural de los componentes, lo que puede contribuir a reducir el impacto ambiental de los desechos y la necesidad de reparaciones y reemplazos.
- La detección y estimación de emisiones fugitivas contaminantes en plantas industriales mediante el uso de cámaras de videovigilancia. Lo que permite la pronta detección y actuación para la corrección de problemas en procesos industriales, contribuyendo a reducir la contaminación del aire y el agua.

Se han utilizado datos de imágenes multiespectrales captadas por satélites y cámaras, y se han aplicado técnicas de segmentación semántica y aprendizaje máquina para evaluar la precisión y la eficiencia de estas técnicas.

## 1.1. Motivación y objetivos

El objetivo principal de esta tesis es evaluar el uso de la segmentación semántica y el análisis de imagen multiespectral en aplicaciones medioambientales y demostrar su valor y relevancia en el ámbito medioambiental. Para ello, es necesario un estudio previo del estado del arte. De esta forma se identifican aplicaciones medioambientales que puedan beneficiarse de estas nuevas tecnologías y así poder evaluar los resultados obtenidos. Finalmente, se obtienen conclusiones sobre la efectividad y precisión de la segmentación semántica en cada una de las aplicaciones estudiadas y se analizan las tendencias y futuros retos en el campo. Por estos motivos, en esta tesis se plantean los siguientes objetivos:

- Realizar una revisión de los trabajos más relevantes y recientes en el área de la segmentación semántica y el análisis de imagen multiespectral para conocer el avance y las tendencias actuales en el campo.
- Identificar aplicaciones medioambientales que puedan beneficiarse de la utilización de la segmentación semántica, ya sea por la mejora en el rendimiento en comparación con sistemas existentes, la posibilidad de realizar

tareas que antes no eran posibles o la automatización de procesos que anteriormente requerían la intervención manual de operarios.

- Evaluar los resultados obtenidos en diferentes aplicaciones medioambientales y compararlos con resultados previos para discutir las mejoras y debilidades de cada técnica utilizada.
- Concluir sobre la efectividad y precisión de la segmentación semántica en cada una de las aplicaciones estudiadas y su potencial para contribuir al conocimiento y la toma de decisiones en el ámbito medioambiental.
- Analizar las tendencias y retos futuros en el campo de la segmentación semántica y el análisis de imagen multiespectral para proponer nuevas direcciones de investigación y establecer el panorama futuro de esta tecnología.

## 1.2. Contexto

Esta tesis se enmarca en diversos proyectos de investigación, que han proporcionado una base sólida para su desarrollo. Asimismo, se ha contado con el privilegio de la Beca Severo Ochoa, la cual ha respaldado la labor investigadora en las últimas fases de la investigación.

- Análisis del uso de terrenos aplicando técnicas de visión por computador.
  - Entidad financiadora: SERESCO, S.A.
  - Referencia: FUI-018-20
  - Plazo de ejecución: 17 de Enero de 2020 a 28 de Febrero de 2021.
  - Descripción: Esta investigación trata el problema del reconocimiento de diferentes tipos de cultivos mediante imágenes captadas por satélite. Se desarrolla una solución capaz de monitorizar el suelo y distinguir más de 6 tipos de cultivo.
- Detección y segmentación de emisiones en imágenes.
  - Entidad financiadora: ArcelorMittal Innovación, Investigación e Inversión, S.L.
  - Referencia: FUI-149-21
  - Plazo de ejecución: 13 de Julio de 2021 a 12 de Noviembre de 2021.

- Descripción: Esta investigación trata el problema de la detección y localización de las emisiones fugitivas en las plantas industriales mediante cámaras de videovigilancia. Se desarrolla un sistema de monitorización que permite localizar la fuente de la contaminación en tiempo real.
- Categorización de emisiones en imágenes nocturnas y multicanal.
  - Entidad financiadora: ArcelorMittal Innovación, Investigación e Inversión, S.L.
  - Referencia: FUI-21-280
  - Plazo de ejecución: 28 de Septiembre de 2021 a 27 de Marzo de 2022.
  - Descripción: Esta investigación trata el problema de la categorización de las distintas severidades de las emisiones fugitivas en las plantas industriales mediante cámaras de videovigilancia. Se desarrolla una solución que permite notificar el nivel de severidad de la contaminación de forma veloz para prevenir riesgos.
- *Assessment of using remote sensing to ensure compliance with nitrate directive.*
  - Entidad financiadora: *Water and Marine Resources Unit (D.2)* del *Joint Research Centre (JRC)* de la *European Commission (EC)*
  - Referencia: CT-EX2022D560006-101
  - Plazo de ejecución: 01 de Abril de 2022 a 31 de Julio de 2022.
  - Descripción: El objetivo de esta colaboración es la evaluación del uso de la teledetección para garantizar el cumplimiento de la normativa e investigar posibles casos de infracción relacionados con la contaminación por nitratos. La investigación trata el problema de la aplicación de estiércol durante el invierno, donde existe un alto riesgo de que los nitratos se filtren a los arroyos y ríos cercanos causando contaminación.



## Capítulo 2

# Análisis, procesamiento y segmentación de imágenes multiespectrales

En este capítulo se presenta una revisión detallada del estado del arte del análisis de imagen multiespectral y la clasificación de regiones en imágenes. En particular, se enfoca en la segmentación semántica, una técnica que permite la clasificación a nivel de píxel, para la que se describen las arquitecturas y técnicas más utilizadas. También se explora el estado del arte monitorización medioambiental, analizando las aplicaciones más relevantes y las tendencias actuales. Este capítulo proporciona una visión general de las últimas investigaciones y desarrollos en el campo estableciendo el contexto para la tesis y proporcionando un punto de referencia para el trabajo presentado.

## 2.1. Análisis de imagen multiespectral

Las imágenes multiespectrales contienen entre 3 y 20 bandas de longitud de onda y proporcionan información sobre diferentes propiedades [49]. Cada banda espectral es registrada como un canal en la imagen multiespectral, y cada canal puede proporcionar información distinta sobre la escena o objeto representado. Por ejemplo, una imagen multiespectral puede incluir bandas que midan la reflectancia verde, la reflectancia roja y la reflectancia infrarroja para obtener información sobre la vegetación, la temperatura y la humedad de la superficie terrestre. Además, el análisis de imágenes multiespectrales se utiliza en una gran variedad de aplicaciones [44, 68, 86, 132], incluyendo: análisis urbano, la monitorización de la biodiversidad y el medio ambiente, defensa y seguridad, la monitorización de desastres naturales, geología y geomorfología, detección de defectos en piezas, evaluación del uso del suelo, la monitorización de la calidad del aire y del agua, identificación de áreas verdes y construcciones, y protección de infraestructuras críticas.

Las ondas electromagnéticas abarcan un amplio espectro de frecuencias o longitudes de onda y su clasificación depende principalmente de su origen, sin embargo, no existe una delimitación precisa en dicha clasificación [92]. Las clasificaciones usadas por ISO 20473:2007 [39] son las siguientes.

- Ultravioleta (UV) (1 - 380 nm): para medir la concentración de ozono y otros gases, para identificar compuestos químicos y en la observación de la atmósfera terrestre y otros cuerpos celestes.
- Luz visible (380 - 780 nm): para imágenes de la atmósfera, calidad del aire, vegetación y agua.
- Infrarrojo Cercano (*Near Infrared*, NIR) (780 - 3 000 nm): para materiales orgánicos, vegetación, y agua.
- Infrarrojo Medio (*Mid infrared*, MIR) (3 000 - 50 000 nm): para gases y aerosoles en la atmósfera, vegetación, contenido de humedad del suelo y incendios forestales.
- Infrarrojo Lejano (*Far infrared*, FIR) (50 000 - 1 000 000 nm): para la detección de emisiones térmicas de la superficie terrestre, la monitorización de la temperatura del agua y la identificación de gases y aerosoles en la atmósfera.
- Radar y tecnologías relacionadas (1 000 000 nm +): para modelado de terrenos, humedad, y detección de objetos.

### 2.1.1. Sensores multiespectrales

Los sensores multiespectrales son capaces de medir la energía en diferentes bandas del espectro electromagnético, lo que les permite distinguir entre diferentes características. A continuación se describen diferentes tipos de sensores multiespectrales utilizados para la captura de imágenes [11, 43, 93, 97, 118].

- Cámaras multiespectrales: estas cámaras tienen la capacidad de capturar imágenes en varias bandas espectrales. Las aplicaciones para estas cámaras incluyen el análisis de la vegetación, la detección de contaminación, el estudio del suelo y la calidad del agua, y en agricultura y silvicultura.
- Espectrómetros: son dispositivos que miden la radiación electromagnética en función de la longitud de onda, lo que permite obtener un espectro de emisión o absorción. Los espectrómetros se utilizan para analizar la composición química de los objetos.
- Escáneres LiDAR: estos sistemas utilizan pulsos de radiación láser para medir la distancia de un objeto y crear mapas 3D de la superficie terrestre. También se utilizan para la detección y medición de la altura de objetos.
- Radiómetros: son instrumentos que miden la radiación electromagnética en una banda específica. Los radiómetros se utilizan para analizar la radiación solar y terrestre.
- Cámaras de termografía: estas cámaras detectan la radiación infrarroja emitida por los objetos y la convierten en una imagen visible en tiempo real con el objetivo de medir la temperatura de los objetos. Las cámaras de termografía se utilizan en diversas aplicaciones, como la inspección de instalaciones eléctricas y mecánicas, la evaluación de la temperatura en procesos industriales, y la búsqueda y rescate de personas en situaciones de emergencia.
- Satélites de imagen multiespectral: estos satélites tienen cámaras multiespectrales y otros sensores que orbitan la Tierra para recopilar datos de la superficie terrestre en múltiples bandas espectrales. Se utilizan en una amplia variedad de aplicaciones, como la observación del clima, el seguimiento de cambios en la cobertura de la tierra, la evaluación de la calidad del agua, y la detección de incendios forestales, entre otros.
- Vehículos aéreos no tripulados (*unmanned aerial vehicle*, UAV): estos vehículos están equipados con cámaras multiespectrales y otros sensores. Por ejemplo, los drones se utilizan para la obtención de imágenes de alta resolución de la superficie terrestre y la recopilación de datos en áreas de difícil acceso.

### **2.1.2. Satélites de imagen multiespectral**

En el análisis de imágenes captadas por satélite, la condición de la atmósfera y la presencia de nubes puede ser un desafío importante [99]. La atmósfera y las nubes pueden afectar la calidad de la imagen y alterar los valores de reflectividad registrados en los diferentes canales multiespectrales. Esto es debido a la presencia de partículas, como polvo y gases, que pueden dispersar y absorber la luz. Además, la atmósfera puede causar una distorsión de la forma y posición de los objetos en la imagen debido a la refracción atmosférica. Por estas razones, es importante realizar una corrección adecuada de la atmósfera y las nubes antes de llevar a cabo el análisis de imagen multiespectral. Hay varios métodos y herramientas disponibles para corregir estos efectos, como la corrección atmosférica basada en modelos, la corrección por reflectancia espectral y la corrección por masa de aire.

Algunas características de los satélites de imagen multiespectral son [84]:

- **Resolución espacial.** La resolución espacial de un satélite multiespectral se refiere a la capacidad de distinguir objetos en la superficie terrestre. Cuanto mayor sea la resolución espacial, mayor será la cantidad de píxeles para una misma superficie. Esto mejora el detalle que se puede observar en las imágenes.
- **Resolución espectral.** La resolución espectral de un satélite multiespectral se refiere a la capacidad de distinguir diferentes longitudes de onda del espectro electromagnético. Cuanto mayor sea la resolución espectral, mayor será la capacidad de distinguir diferentes características de la superficie terrestre.
- **Ancho de banda.** El ancho de banda de un satélite multiespectral se refiere al rango de longitudes de onda del espectro electromagnético que se pueden capturar. Cuanto mayor sea el ancho de banda, mayor será la cantidad de información que se puede obtener de las imágenes.
- **Frecuencia de captura de imágenes.** Los satélites multiespectrales pueden capturar imágenes con diferentes frecuencias, dependiendo de la misión y los objetivos. Algunos satélites pueden capturar imágenes diarias, mientras que otros pueden tardar varias semanas o incluso meses en volver a capturar imágenes de la misma zona.

Dos de los satélites de imagen multiespectral más conocidos son Landsat8 [5] y Sentinel-2 [6]. Sentinel-2 es operado por la Agencia Espacial Europea y tiene un sensor multiespectral de alta resolución con una resolución espacial de entre 10 y 60 metros con una frecuencia de captura de 5 días. Landsat 8 es operado por la NASA y tiene dos sensores que pueden capturar imágenes en 9 bandas



espectrales con una resolución espacial de entre 30 y 60 metros con una frecuencia de 16 días. Ambos satélites se utilizan para la gestión de recursos naturales, la agricultura, la planificación urbana, la gestión de desastres y otros campos. La elección de utilizar uno u otro dependerá de las necesidades específicas de la aplicación y de las características de los satélites.

### 2.1.3. Preprocesamiento de los datos

La preparación adecuada de los datos es esencial para el procesamiento exitoso de la información. En el caso de los datos de imágenes multiespectrales, hay varios aspectos a tener en cuenta [4].

- Normalizar los datos, ya que cada banda de la imagen puede tener un rango muy dispar.
- Índices multiespectrales, para facilitar la visualización de las bandas y para resaltar ciertas propiedades de las imágenes. Los índices multiespectrales combinan múltiples bandas espectrales para crear una nueva imagen que destaca ciertas características de la imagen original. Estos índices pueden ser utilizados para identificar diferentes características. Aquellos índices multiespectrales destinados a la agricultura de precisión se les conoce como índices de vegetación [115]. Algunos ejemplos incluyen el Índice de Vegetación Normalizado (*Normalized difference vegetation index*, NDVI), o el Índice de Vegetación Ajustado al Suelo (*Soil Adjusted Vegetation Index*, SAVI).
- Reducir el número de canales de las imágenes multiespectrales de entrada mediante la combinación de los canales en una representación tridimensional con técnicas como el Análisis de Componentes Principales (*Principal Component Analysis*, PCA) [50, 91], o mediante la selección de los canales más informativos mediante una selección de características con técnicas como la Eliminación de Características Recursiva (*Recursive Feature Elimination*, RFE) o Boruta [61, 104, 140].
- Balancear las clases para que el conjunto de datos sea representativo de la población real.
- Suavizar las imágenes y eliminar el ruido, destacar elementos de interés, o escalar las diferentes bandas para que todas posean la misma resolución y localización.

## 2.2. Clasificación de regiones

La segmentación en imágenes es un campo de la visión por computador que trata de clasificar diferentes regiones de una imagen según su contenido o significado. Su utilidad se extiende a todos aquellos datos que pueden ser expresados en forma de imagen por lo que sus aplicaciones son muy variadas pasando desde la edición de imágenes hasta el reconocimiento facial. Existen una gran cantidad de técnicas para segmentar imágenes [103, 116] como pueden ser la umbralización, *watershed* o transformación divisoria, o algoritmos de agrupamiento. Para ello se basan en la información de características como el color, la intensidad o la textura de la imagen.

Esta Tesis Doctoral se centrará en aquellos algoritmos de la segmentación de imágenes enfocados a la clasificación de regiones. El objetivo es el de detectar, localizar y clasificar diferentes elementos de la imagen, y no solamente obtener los bordes o textura.

### 2.2.1. Métodos tradicionales

En esta sección se describirán algoritmos de clasificación tradicionales basados en aprendizaje automático. Estos algoritmos han sido ampliamente utilizados en el pasado y siguen siendo populares en la segmentación de imágenes debido a su simplicidad y facilidad de implementación [89]. Aunque pueden no ser tan precisos como los modelos de aprendizaje profundo, ofrecen una buena opción para problemas con limitaciones en términos de recursos computacionales o cuando no se pueden obtener grandes conjuntos de datos, ya que en el aprendizaje profundo se requieren conjuntos de datos enormes para lograr resultados óptimos.

#### 2.2.1.1. *K-Nearest Neighbors*

*K-nearest neighbors* (KNN) [45] es un algoritmo de aprendizaje no supervisado que se basa en la idea de asignar una muestra a la clase de mayor frecuencia entre sus  $K$  vecinos más cercanos en el espacio de características (ver Figura 2.1). Se utiliza comúnmente en la clasificación y la regresión, y su principal ventaja es su simplicidad y rapidez de ejecución.

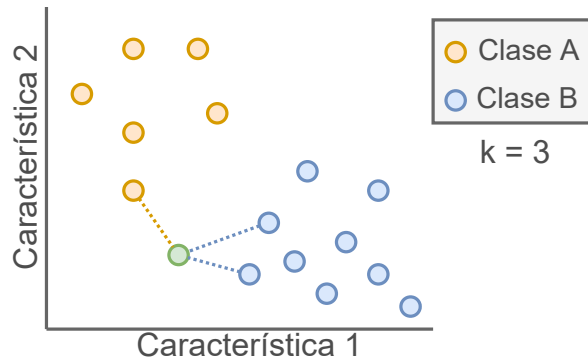


Figura 2.1: Ejemplo de *K-Nearest Neighbors* con  $K$  igual a tres para dos clases.

### 2.2.1.2. *Naive Bayes*

*Naive bayes* (NB) [108] se basa en el cálculo de la probabilidad de cada clase y la probabilidad condicional de cada característica dada una clase. Una vez entrenado con un conjunto de datos etiquetados, se aplica la clasificación por votación Bayesiana para determinar la clase más probable para un nuevo conjunto de características. La Figura 2.2 muestra un ejemplo de clasificación NB.

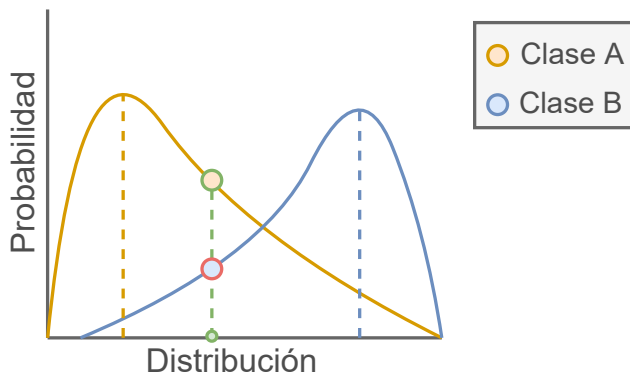


Figura 2.2: Ejemplo de *Naive bayes* para dos clases.

### 2.2.1.3. *Random Forest*

*Random Forest* (RF) [94] es un algoritmo de aprendizaje automático de tipo ensamblado, que se basa en la construcción de una gran cantidad de árboles de decisión y utiliza el promedio o la moda de las predicciones de cada árbol para obtener una predicción final (ver Figura 2.3). Para obtener este efecto se utiliza la técnica de *Bagging*, que consiste en la selección aleatoria de los elementos del conjunto de datos para evitar que existan dos árboles iguales. Es utilizado en una variedad de tareas de aprendizaje automático, incluyendo la clasificación y la regresión. En el caso de la segmentación de imágenes, se utiliza para clasificar cada píxel de la imagen en una determinada clase.

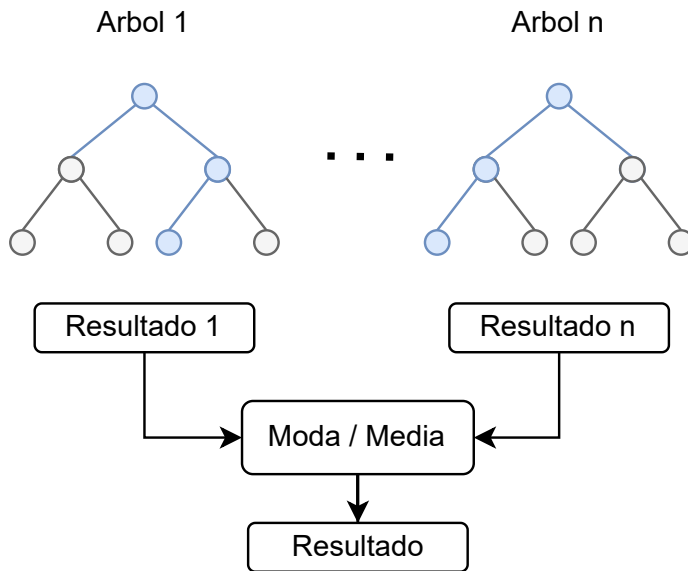


Figura 2.3: Estructura de *Random Forest*.

#### 2.2.1.4. Support-Vector Machines

*Support-Vector Machines* (SVM) [107] es un algoritmo de aprendizaje automático de tipo no paramétrico que se utiliza para clasificar datos en dos o más categorías. Se basa en la idea de encontrar el espacio que separe los datos en diferentes categorías con el mayor margen posible (ver Figura 2.4). En el caso de la segmentación de imágenes, se utiliza para clasificar cada píxel de la imagen en una determinada clase.

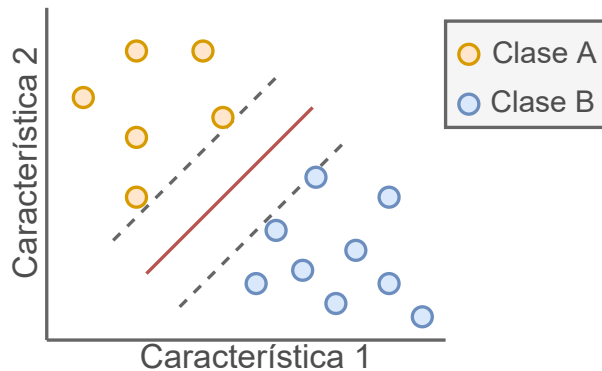


Figura 2.4: Ejemplo de *Support-Vector Machines* para dos clases.

#### 2.2.1.5. Redes Neuronales Artificiales

Las Redes Neuronales Artificiales (*Artificial Neural Network*, ANNs) [30] son modelos matemáticos de aprendizaje automático que se basan en la estructura y el funcionamiento de las redes neuronales biológicas. Estos modelos se componen de una serie de nodos interconectados, cada uno de los cuales representa una unidad de procesamiento y puede realizar cálculos simples (ver Figura 2.5). La utilización de las ANNs se extiende a una amplia gama de aplicaciones, incluyendo la visión por computador, el procesamiento del lenguaje natural y la detección de patrones en tablas de datos. En el análisis de imágenes multispectrales a menudo se utilizan ANNs de baja complejidad.

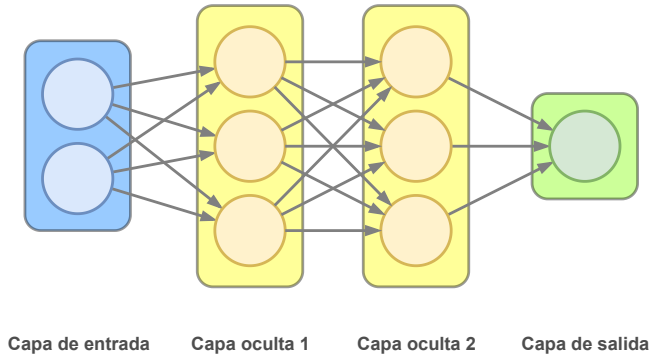


Figura 2.5: Estructura de una Red Neuronal Artificial con dos capas ocultas.

### 2.2.2. Métodos basados en aprendizaje profundo

Actualmente, existen técnicas mucho más complejas y avanzadas que se aprovechan del aprendizaje profundo (*deep learning*, DL) basándose en redes convolucionales neuronales (*convolutional neural network*, CNN) [80]. Sin embargo, este enfoque tiene la desventaja de que se necesitan grandes conjuntos de datos debido a la necesidad de los modelos de aprendizaje profundo de exponerse a una amplia variedad de ejemplos para aprender a generalizar adecuadamente y evitar el sobreajuste. Un conjunto de datos demasiado pequeño puede llevar a un modelo que se ajuste demasiado a los datos de entrenamiento y no sea capaz de generalizar a nuevos datos. Sin embargo, la adquisición y etiquetado de un gran conjunto de datos puede conllevar un costo económico y una inversión temporal significativa, además de aumentar los requerimientos de recursos computacionales. Por lo tanto, es importante encontrar un tamaño adecuado para el conjunto de datos que permita un buen rendimiento sin ser demasiado grande ni demasiado pequeño.

Los cuatro métodos de segmentación de imágenes basados en aprendizaje profundo más conocidos son: detección de objetos, segmentación semántica, segmentación de instancias y segmentación panóptica. En general, se puede observar que las redes neuronales utilizadas en la segmentación de imágenes están en constante evolución y mejora, y se están desarrollando nuevas técnicas y arquitecturas para superar las limitaciones existentes y mejorar la precisión en la clasificación.

### 2.2.2.1. Detección de objetos

La detección de objetos [31] se refiere a la tarea de encontrar y localizar objetos específicos dentro de una imagen o vídeo. Uno de los algoritmos más utilizados para esta tarea es YOLO (*You Only Look Once*) [106], que utiliza una red neuronal para clasificar y localizar los objetos en una sola pasada. YOLOv8 es la última versión de este algoritmo, que es capaz de detectar objetos con alta precisión y velocidad. Esta técnica permite contar el número de objetos de un determinado tipo presentes en una imagen. Sin embargo, la detección de objetos no puede detectar conceptos abstractos, y sólo puede segmentar regiones rectangulares, lo que puede limitar su precisión (ver Figura 2.6). Tiene aplicaciones como el reconocimiento facial o la videovigilancia.

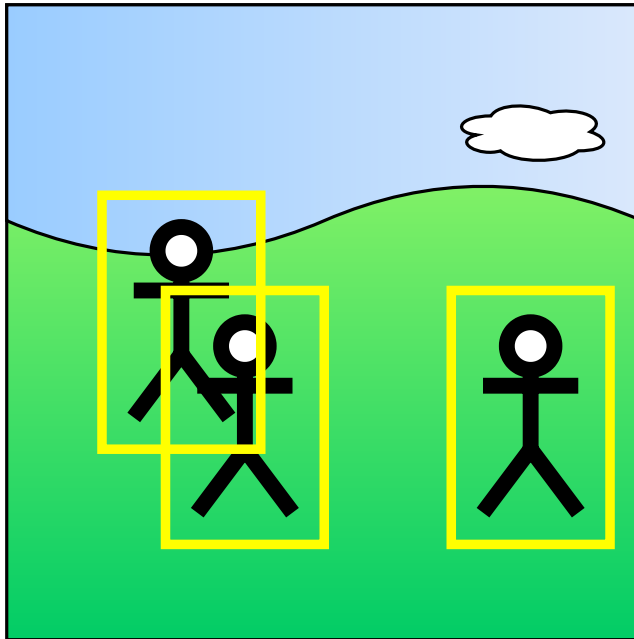


Figura 2.6: Ejemplo de detección de objetos.

### 2.2.2.2. Segmentación semántica

La segmentación semántica [46] es la tarea de clasificar cada píxel de una imagen según su significado o contenido. Uno de los algoritmos más utilizados para esta tarea es la red neuronal UNet [109], que combina tanto características de baja y alta resolución para lograr una mejor precisión en la segmentación. Este algoritmo es muy popular debido a su simplicidad, bajo coste computacional, y facilidad para alterar su arquitectura para adaptarse a una tarea concreta. Otro algoritmo es DeepLabv3+ [21], desarrollado por Google. La segmentación semántica es capaz de detectar objetos en formas irregulares y conceptos abstractos. Sin embargo, no permite contar el número de objetos de un determinado tipo ya que solamente asigna una clase a cada píxel (ver Figura 2.7). Tiene aplicaciones en ámbitos como el análisis de imágenes aéreas y la detección de anomalías en imágenes médicas.

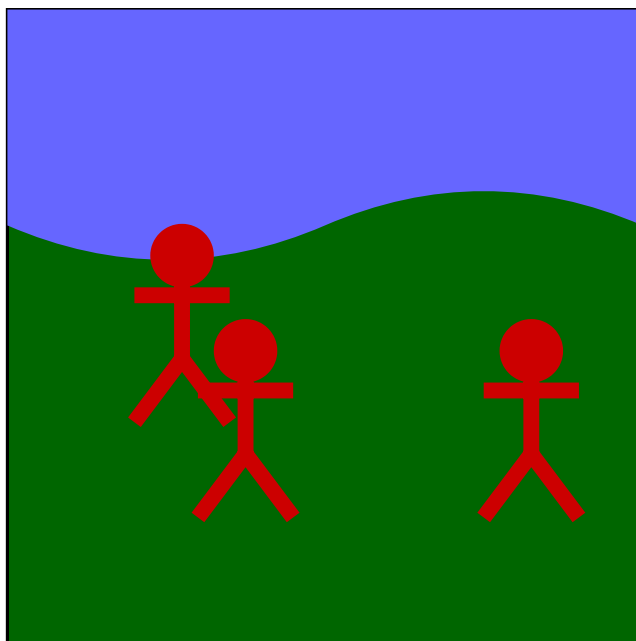


Figura 2.7: Ejemplo de segmentación semántica.



### 2.2.2.3. Segmentación de instancias

La segmentación de instancias [48] trata de segmentar de forma individual a cada objeto o elemento. De esta forma, se mejora la precisión de la detección de objetos al no tener que depender de una región rectangular. Existen versiones de YOLO adaptadas a la segmentación de instancias [55]. Sin embargo, de igual forma que con la detección de objetos, no tiene la capacidad de clasificar objetos incontables o abstractos como “cielo” o “carretera” (ver Figura 2.8).

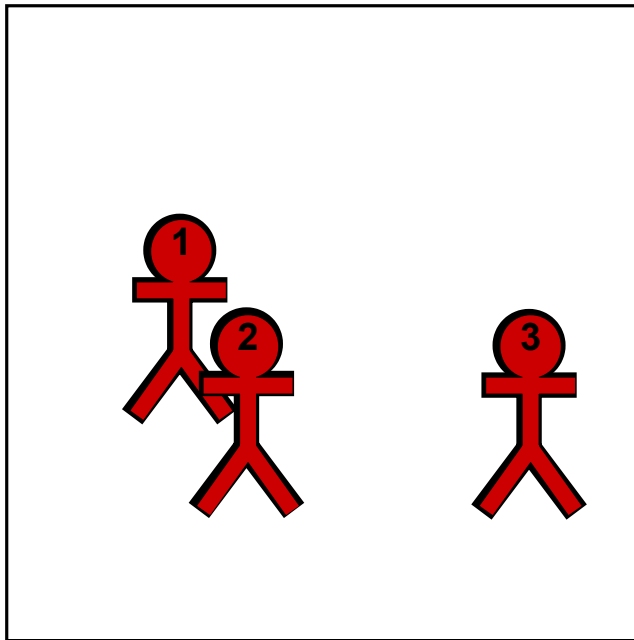


Figura 2.8: Ejemplo de segmentación de instancias.

#### 2.2.2.4. Segmentación panóptica

La segmentación panóptica [34] es la tarea que trata de combinar las ventajas de la segmentación semántica y la detección de objetos. De esta forma, es capaz de detectar objetos de forma similar a la segmentación de instancias pero, además, puede clasificar conceptos incontables como “mar” o “suelo” (ver Figura 2.9). Esta técnica es la más reciente, desarrollada en 2017 mediante el algoritmo UPSNet [128], aunque ya existen otros algoritmos como EfficientPS [83], o variaciones de DeepLab que implementan la segmentación panóptica [22]. Sin embargo, la segmentación panóptica es la más costosa computacionalmente y la creación de conjuntos de datos para su aprendizaje tiene una mayor complejidad debido a la necesidad de clasificar cada píxel de la imagen y asignar cada objeto individualmente, separando las clases contables de aquellas que sean incontables. Por este motivo su utilidad se ve reducida considerablemente en casos de uso reales. Una de sus aplicaciones más conocidas es la conducción autónoma.

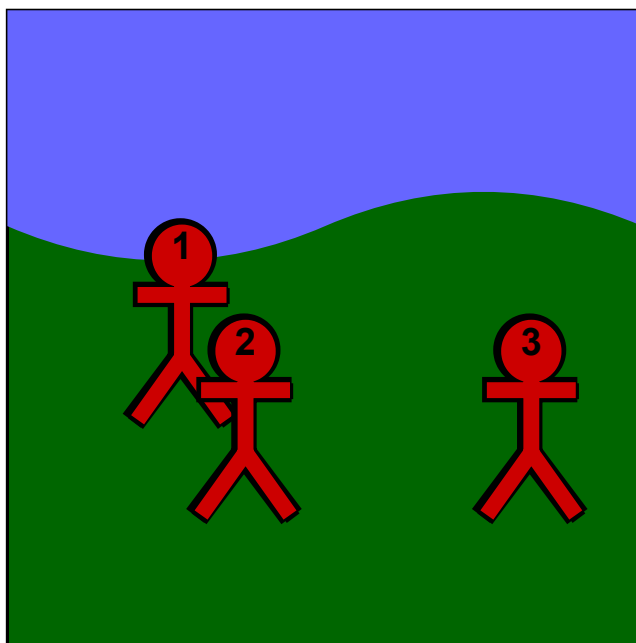


Figura 2.9: Ejemplo de segmentación panóptica.

## 2.3. Diseño y configuración de las redes de segmentación semántica

La segmentación semántica ofrece la posibilidad de detectar regiones de interés independientemente de su significado, es decir contables o incontables. Esto permite una amplia variedad de aplicaciones que necesitan de una localización precisa o el área que ocupan diferentes agentes contaminantes permitiendo la monitorización constante y la implementación de alarmas según su riesgo. En estos casos no se necesita de un conteo de elementos, por lo que se ha decidido utilizar la segmentación semántica como punto de partida para mejorar el rendimiento en diversas tareas para preservar el medio ambiente. De esta forma se reduce la complejidad del etiquetado de datos respecto a otras técnicas como la segmentación de instancias o segmentación panóptica, facilitando su posible adopción.

En esta sección se describen las redes neuronales convoluciones de segmentación semántica del estado del arte: UNet y DeepLabV3+. Estas redes son modelos capaces de obtener resultados de gran precisión en aplicaciones de visión por computador para la segmentación de imágenes. Además, se detallará el proceso necesario para entrenar y evaluar estos modelos, incluyendo las métricas más utilizadas y la configuración de los hiperparámetros.

### 2.3.1. Entrenamiento

En esta sección se describirá el proceso para entrenar un modelo. Es importante señalar que el entrenamiento de un modelo es un proceso complejo que requiere del estudio y tratamiento del conjunto de datos de forma previa.

El primer paso en el proceso de entrenamiento es dividir el conjunto de datos en tres partes: conjunto de entrenamiento, conjunto de prueba y conjunto de validación. El conjunto de entrenamiento se utiliza para entrenar el modelo, mientras que el conjunto de prueba se utiliza para ajustar los hiperparámetros del modelo y evaluar el rendimiento en datos no vistos previamente. Por último, el conjunto de validación se utiliza para evaluar la capacidad predictiva del modelo y su capacidad para generalizar a datos nuevos. Es importante tener en cuenta que, para obtener resultados realistas, es necesario utilizar conjuntos de datos que sean independientes. Además, este proceso se puede repetir variando los conjuntos. A este proceso se le conoce como validación cruzada y permite obtener unos resultados más robustos.

Un problema común en el entrenamiento de modelos es el sobreajuste. El sobreajuste ocurre cuando el modelo aprende patrones específicos del conjunto de entrenamiento y no generaliza bien a nuevos datos. Para evitar el sobreajuste, se pueden utilizar técnicas como la regularización, la parada temprana, el aumento

de datos o la configuración de hiperparámetros.

Para configurar los hiperparámetros del modelo existen varias técnicas [29,53,69]:

- Exploración manual: esta técnica se enfoca en el ajuste de hiperparámetros de forma manual. Aunque puede resultar efectiva en modelos pequeños, el proceso puede ser tedioso y requerir mucho tiempo.
- Exploración aleatoria: este método consiste en probar combinaciones de hiperparámetros seleccionados al azar para evaluar su desempeño. Aunque no es la técnica más eficiente, puede ser útil para encontrar combinaciones de hiperparámetros inesperadas que produzcan buenos resultados.
- *Grid Search*: esta técnica consiste en evaluar de forma sistemática diferentes combinaciones de hiperparámetros a través de una rejilla de valores posibles. Es eficiente para ajustar un número limitado de hiperparámetros, pero puede ser menos efectiva en modelos complejos.
- Métodos Bayesianos: utilizan la teoría de probabilidades para determinar qué combinaciones de hiperparámetros son las más prometedoras, acelerando la búsqueda. Son útiles para ajustar un gran número de hiperparámetros.
- Métodos evolutivos: usan técnicas inspiradas en la selección natural para generar y evaluar diferentes combinaciones de hiperparámetros. Son efectivos para encontrar combinaciones inesperadas de hiperparámetros, pero pueden requerir muchos cálculos.
- AutoML (Aprendizaje Automático Automatizado): es una técnica que busca automatizar todo el proceso de aprendizaje automático, incluyendo la configuración de hiperparámetros. Puede ser útil para mejorar la eficiencia en la selección de hiperparámetros.

### 2.3.2. Métricas

En esta sección se describe brevemente el conjunto de métricas más utilizadas para evaluar el rendimiento de los modelos entrenados [37]. Las métricas incluyen verdaderos positivos (*True Positive*, TP), verdaderos negativos (*True Negative*, TN), falsos positivos (*False Positive*, FP) y falsos negativos (*False Negative*, FN). En la Figura 2.10 se muestra como calcular estas métricas utilizando la predicción respecto al objetivo.

		Objetivo	
		Positivo	Negativo
Predicción	Positivo	Verdadero Positivo (TP)	Falso Positivo (FP)
	Negativo	Falso Negativo (FN)	Verdadero Negativo (TN)

Figura 2.10: Verdadero positivo, falso positivo, verdadero negativo, y falso negativo.

*Overall accuracy* (OA): un porcentaje que representa cuántos píxeles están correctamente clasificados del total. Esta métrica puede ser engañosa si las clases no están equilibradas. Por ejemplo, dadas dos clases, si una de ellas representa el 99 % de los píxeles en el conjunto de datos y la otra representa el 1 % restante, incluso si todos los píxeles de la segunda clase se clasifican incorrectamente como píxeles de la primera clase, esta métrica aún obtendrá un 99 % de acierto.

*Precision* (P) mide el porcentaje de píxeles clasificados correctamente sobre el total de predicciones para una clase determinada. Una alta *Precision* indica que los píxeles que se clasifican como de una clase particular están clasificados de forma correcta. Sin embargo, no refleja el porcentaje de píxeles clasificados sobre el total.

$$P = \frac{TP}{TP + FP} \quad (2.1)$$

*Recall* (R) mide el porcentaje de píxeles clasificados correctamente sobre el total de píxeles para una clase determinada. Una alta *Recall* indica que se clasifican la mayoría de píxeles de esa clase. Sin embargo, no refleja el porcentaje de acierto.

$$R = \frac{TP}{TP + FN} \quad (2.2)$$

*F<sub>1</sub>-Score* (F<sub>1</sub>) combina la *Precision* y la *Recall* facilitando la comparación de modelos. Un buen modelo debe tener un equilibrio entre la *Precisión* y *Recall*. Esta métrica es equivalente al coeficiente de Dice con dos clases.

$$F_1 = \frac{2 \times P \times R}{P + R} \quad (2.3)$$

*Intersection-Over-Union* (IoU) mide la similitud entre el objetivo y la predicción. Esta métrica es equivalente al índice de Jaccard.

$$IoU = \frac{\text{Área de intersección}}{\text{Área de Unión}} = \frac{TP}{TP + FN + FP} \quad (2.4)$$

### 2.3.3. Hiperparámetros

En esta sección se explicarán los hiperparámetros comunes que se utilizan en la segmentación semántica que influyen en el proceso de entrenamiento [123].

- *Input size* o tamaño de entrada: la resolución y número de canales de las imágenes de entrada. A mayor resolución y número de canales mayor es el coste computacional. Si el coste es demasiado alto es posible que no pueda ejecutarse debido a limitaciones en la memoria de video (VRAM).
- Clases: número de clases que la red podrá detectar.
- *Class balancing* o equilibrio de clases: el método utilizado para evitar un entrenamiento sesgado cuando el conjunto de datos está desequilibrado. Los más comunes son el Peso de Frecuencia Inversa (*Inverse Frequency Weighting*, IFW) [25] y el Peso de Frecuencia Mediana (*Median Frequency Weighting*, MFW) [33].
- *Solver* o solucionador: algoritmo que calcula el gradiente al entrenar la red. Los más utilizados son *Root Mean Square Propagation* (RMSProp), *Stochastic Gradient Descent with Momentum* (SGDM) y *Adaptive Moment Estimation* (Adam) [28].
- *Epochs* o ciclos: número de veces que se utiliza el conjunto de datos completo en el entrenamiento. Dependiendo del tamaño del conjunto de datos, un número muy elevado puede llegar al sobreajuste del modelo.
- *Batch size* o tamaño de lote: número de imágenes tras el que actualizar el modelo durante el entrenamiento. Un valor pequeño mejora la generalización pero aumenta la variabilidad, mientras que un valor grande suaviza la variabilidad pero puede converger a mínimos locales. Este valor también afecta al coste computacional del entrenamiento ya que aumenta o disminuye el número de actualizaciones de los pesos de la red así como la cantidad de datos que cargar en memoria en cada iteración.
- *Learning rate* o tasa de aprendizaje: parámetro que controla cómo se ajustan los pesos de la red con respecto al gradiente.
- *Gradient clipping* o recorte del gradiente: limita el valor máximo del gradiente para evitar el problema del gradiente explosivo al entrenar.

- Regularización L2: técnica para reducir la complejidad de un modelo penalizando la función de pérdida. Como resultado, se reduce el sobreajuste.
- *Data augmenting* o aumento de datos: generación de nuevos datos a partir de los datos originales con el objetivo de obtener un dataset mayor y más variado para reducir el sobreajuste del modelo.
- *Shuffle* o barajar: el conjunto de datos se baraja en cada epoch para evitar que el orden del entrenamiento sea el mismo y así reducir el sobreajuste del modelo.

### 2.3.4. Arquitecturas

Esta sección se centra en dos de las arquitecturas más populares en el campo de la segmentación semántica: UNet y DeeplabV3+. Ambas arquitecturas han demostrado ser efectivas para la segmentación de imágenes y tienen su uso en una amplia gama de aplicaciones. Otras redes relevantes como Mask-RCNN [51] y SegNet [7] han sido evaluadas en el ámbito de segmentación de imágenes. Sin embargo, estas arquitecturas se encuentran en desuso en comparación con las más modernas y efectivas, Unet y DeeplabV3+.

#### 2.3.4.1. UNet

UNet es una arquitectura de redes neuronales utilizada en el campo de la visión por computador y el procesamiento de imágenes. Fue desarrollada en 2015 y se ha utilizado con éxito en una amplia variedad de tareas de segmentación de imágenes, incluyendo la segmentación de órganos en imágenes médicas y la segmentación de objetos en imágenes de satélite.

UNet se basa en la arquitectura de redes neuronales convolucionales *encoder-decoder* (ver Figura 2.11), lo que significa que consta de dos partes: un codificador y un decodificador. El codificador se encarga de extraer características de las imágenes de entrada a través de la aplicación de múltiples filtros de convolución, mientras que el decodificador se encarga de utilizar esas características para generar una salida segmentada.

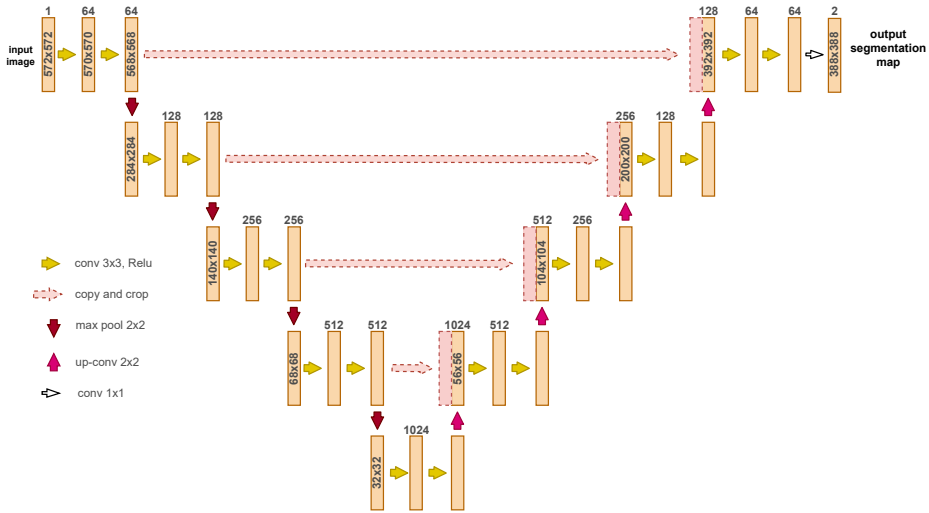


Figura 2.11: Arquitectura de UNet.

Una de las características distintivas de UNet es que utiliza una conexión de “saltos” entre el codificador y el decodificador, lo que permite que la información se transfiera entre ambas partes de la red y se utilice para mejorar la precisión de la segmentación. Su simplicidad permite realizar entrenamientos de forma veloz sin necesidad de modelos pre-entrenados. Además, esto permite una mayor flexibilidad a la hora de modificar su arquitectura, dando lugar a una gran variedad de redes basadas en UNet como pueden ser WideUNet, UNet++ o SwanUNet.

Unet tiene hiperparámetros particulares para su arquitectura además de los mencionados en la sección de Hiperparámetros.

- **Profundidad:** número de capas de *max pooling* en la arquitectura de UNet. Sirve para aumentar la capacidad de abstracción del modelo, lo que permite que el modelo sea capaz de detectar características más complejas en las imágenes de entrada, aunque puede provocar un sobreajuste del modelo.
- **Filtros en la primera capa:** número de filtros en la primera convolución de la arquitectura de UNet. Este valor se multiplica por 2 en cada nivel de profundidad. Afecta a la capacidad de la red de generar características y a su coste computacional. Si se aumenta indefinidamente puede causar un sobreajuste del modelo.



#### 2.3.4.2. DeepLabV3+

DeepLabV3+ es una arquitectura de redes neuronales utilizada en el campo de la visión por computador y el procesamiento de imágenes para tareas de segmentación de imágenes. Fue desarrollada por Google en 2018 y es una versión mejorada de la arquitectura DeepLabV3 original [20].

DeepLabV3+ utiliza una arquitectura *encoder-decoder* (ver Figura 2.12) similar a la utilizada por otras arquitecturas de redes neuronales de segmentación de imágenes, como UNet. El codificador se encarga de extraer características de las imágenes de entrada a través de la aplicación de múltiples filtros de convolución, mientras que el decodificador se encarga de utilizar esas características para generar una salida segmentada.

A diferencia de la arquitectura original de UNet, y al igual que otras redes de gran complejidad, este algoritmo se aprovecha de la arquitectura de redes de clasificación de imágenes como Resnet [52] o Xception [23]. Esta parte de la red se le denomina *Deep Convolutional Neural Network* (DCNN) como se puede apreciar en la Figura 2.12. Esto sirve no solo para obtener una buena base, sino también para reutilizar los pesos de modelos entrenados de estas redes y reducir el tiempo de entrenamiento de forma considerable.

Una de las características distintivas de DeepLabV3+ es que utiliza un módulo conocido como *Atrous Spatial Pyramid Pooling* (ASPP) para mejorar la precisión de la segmentación. Este módulo trata de escalar la imagen para expandir el campo de visión de la red y así capturar información a diferentes niveles y obtener una segmentación más precisa.

DeepLabV3+ tiene hiperparámetros particulares para su arquitectura además de los mencionados en la sección Hiperparámetros. Estos hiperparámetros se indican a continuación:

- **DCNN:** red de clasificación utilizada como parte de la arquitectura base de una red más compleja, muy común en las arquitecturas de segmentación semántica. Se utiliza para extraer características de las imágenes de entrada. Las más utilizadas en DeepLabV3+ son Resnet50, MobileNetV3 y Xception71.
- **Output stride:** la división entre la resolución de la imagen de entrada y el mapa de características final. Por ejemplo, si la imagen de entrada tiene una resolución de  $256 \times 256$  y el mapa de características final  $32 \times 32$ , el paso es de 8. Sirve para controlar la separación entre convoluciones. Un valor más alto ofrece un mayor campo de visión para la red y reduce su coste computacional. Generalmente los valores pequeños obtienen resultados mejores.
- **Batch normalization:** es un parámetro que permite entrenar las capas de normalización por lotes en lugar de utilizar las ya pre-entrenadas. Esto

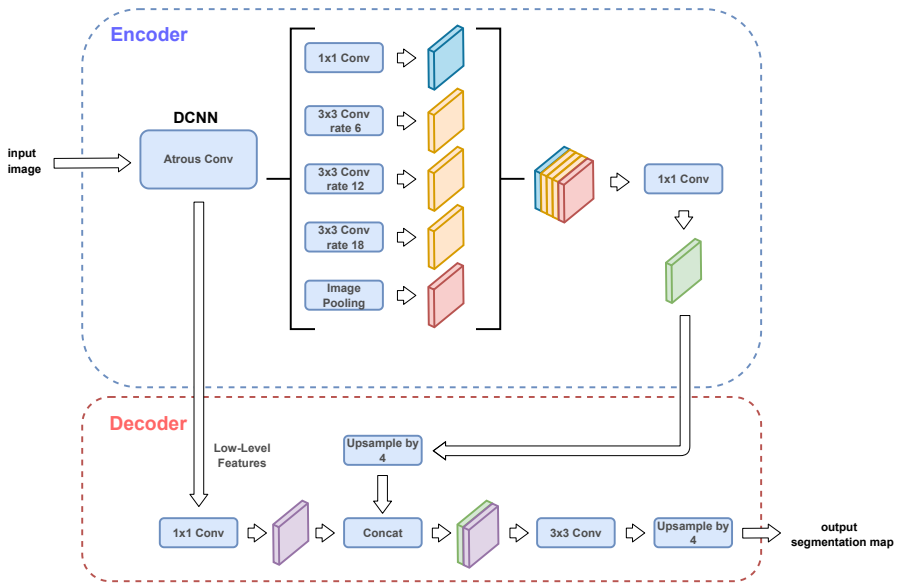


Figura 2.12: Arquitectura de DeepLabV3+.

puede ser útil para adaptar el modelo a un conjunto de datos específico y mejorar el rendimiento de la segmentación, aunque aumenta su coste computacional.

## 2.4. Monitorización medioambiental

En esta sección se describirán las aplicaciones medioambientales más relevantes de la segmentación de imágenes: agricultura de precisión, detección de anomalías o defectos, y análisis de emisiones contaminantes.

### 2.4.1. Agricultura de precisión

La agricultura de precisión trata de reducir costes, mejorar la gestión de cultivos, y reducir el impacto medioambiental de la agricultura. Para ello se hace uso de información geográfica ya sea de imágenes multispectrales de UAVs o de satélites.

Las aplicaciones más comunes para las imágenes captadas por satélite [115] son: el reconocimiento del terreno, la estimación del rendimiento de cultivos, la detección de enfermedades en plantaciones, y el cálculo de características rele-

vantes como índices de clorofila o cantidad de agua. En estos ámbitos se utilizan algoritmos de aprendizaje automático clásicos como *Random Forest* [121]. Sin embargo, recientemente se está empezando a integrar el aprendizaje profundo para tareas como el reconocimiento del terreno. En concreto, en el reconocimiento de diferentes tipos de cultivo [66]. A pesar de esto, arquitecturas más complejas como las vistas en la segmentación semántica aún son poco comunes. La segmentación semántica es una técnica con gran potencial en este área y aún en proceso de ser explotada. En la actualidad, la clasificación del uso del suelo a partir de imágenes aéreas sigue siendo un proceso manual que consume tiempo y dinero. La automatización reduce los costes y aumenta la velocidad de procesamiento. Esto permite nuevas aplicaciones que requieren una respuesta en milisegundos, como la clasificación de cada fotograma de un vídeo. La automatización de tareas de ubicación en imágenes aéreas abre la posibilidad de explorar nuevos servicios, como la monitorización de cultivos, que resultan de gran interés a las empresas.

Los fertilizantes utilizados en la agricultura pueden contaminar el medio ambiente. Existen varios estudios que analizan el impacto ambiental del uso de fertilizantes como el estiércol [64, 70, 124]. El problema más discutido es el impacto del lixiviado de nitrógeno en cuerpos de agua cercanos, incluyendo las aguas subterráneas. Este problema se acentúa con la lluvia, que transporta el nitrógeno del estiércol a las aguas subterráneas y causa una grave contaminación, dañando el medio ambiente y a las personas que usan esa agua. Recientemente se han encontrado grandes cantidades de peces muertos debido a la falta de oxígeno en el agua por la contaminación causada por el lixiviado de fertilizantes de cultivos cercanos [35, 42, 63]. En la literatura también cubren otros problemas, como el uso excesivo de suplementos para el crecimiento del ganado. Cuando estos suplementos se sobredosifican, los animales excretan el exceso, contaminando el estiércol con metales. Con el tiempo, esto contamina el suelo y su entorno [17].

De acuerdo con [38], la cantidad total de estiércol depositada en terrenos agrícolas en todo el mundo es de 116 millones de toneladas. El 50% de esta cantidad representa el exceso de nitrógeno, lo que equivale a 58 millones de toneladas. De esta cantidad, 23 millones de toneladas de nitrógeno se volatilizan en la atmósfera, principalmente como gas de amoníaco, lo cual tuvo un impacto significativo en la calidad del aire de la región. La lixiviación causó la pérdida de aproximadamente 35 millones de toneladas de exceso de nitrógeno en el agua, lo que posteriormente contaminó las vías fluviales y finalmente los mares costeros.

Las soluciones propuestas incluyen el uso de períodos cerrados en los que los agricultores no pueden utilizar fertilizantes con alto contenido de nitrógeno, pero su aplicación es difícil de monitorizar en áreas extensas [124]. La monitorización mediante satélites puede ser una solución viable, pero aún no se han desarrollado métodos específicos para detectar el uso de fertilizantes en diferentes tipos de terrenos utilizando esta tecnología. En la detección de campos recientemente

abonados se tiende a utilizar UAVs y otros datos in situ. Por lo que la detección mediante el uso de imágenes de satélite aun esta sin explotar. Lo que podría mejorar este campo en gran medida gracias a la posibilidad de automatizar una monitorización en tiempo real posibilitando la gestión del fertilizante y fomentando leyes que prohíban su uso en temporadas de riesgo.

No existe literatura científica que aborde la detección del uso de abonos en diferentes tipos de campos utilizando tecnologías de aprendizaje profundo en imágenes de satélites multispectrales. El estudio más parecido a este problema es el de [130], en el que se utilizan árboles de decisión para distinguir entre abonos orgánicos y fertilizantes químicos en dos tipos de cultivos diferentes. En este caso, las imágenes se adquirieron desde aviones y tienen una resolución espacial de 2 m<sup>2</sup>. Las imágenes obtenidas desde aviones tienen mayor resolución que las imágenes obtenidas por satélite, pero su frecuencia es escasa para un sistema de monitorización. En [27] se evalúa la relación de la presencia de estiércol en tierras arables con la intensidad de diferentes índices multispectrales (*Multispectral Index*, MI) utilizando el satélite Sentinel-2 (S2), pero no estudia ningún método de clasificación. Los fertilizantes son una fuente importante de contaminación del agua y del aire, por lo que es importante poder detectar y controlar su uso y distribución. Los satélites son una herramienta valiosa para monitorizar la aplicación de fertilizantes y evaluar su impacto en el medio ambiente.

### 2.4.2. Detección de defectos

El control de calidad es de gran interés en la industria, lo que provoca esfuerzos continuos para mejorar los métodos anteriores. La inspección por medios no destructivos (*Non-Destructive Testing*, NDT) son un conjunto de métodos de análisis para inspeccionar, probar y evaluar materiales, componentes o sistemas sin dañar el objeto. Este enfoque tiene ventajas sobre las pruebas destructivas (DT): se puede usar para analizar cada artículo en lugar de uno por lote ya que las pruebas no dañan los productos, lo que conduce a un coste más bajo y reduce los residuos generados, ya que los artículos no deben ser reemplazados después de la prueba. Además, como el objeto no se daña durante la prueba, las pruebas NDT también se pueden aplicar para detectar problemas como método de mantenimiento durante la vida útil del producto, mejorando el uso a largo plazo y la seguridad [56].

Los métodos de NDT se pueden dividir en aquellos que son de Contacto y No Contacto. Los métodos de contacto consisten fundamentalmente en pruebas ultrasónicas, pruebas de corriente de Foucault, pruebas magnéticas y pruebas de penetración. Los métodos sin contacto consisten fundamentalmente de ultrasonido acoplado al aire, pruebas de radiografía, termografía, shearografía e inspección visual [41].

El análisis más común de defectos se realiza manualmente por un experto

en el campo. Los resultados de las inspecciones NDT sin contacto, y también de algunas técnicas de contacto automatizadas, se representan generalmente a través de imágenes. En estas situaciones, los expertos suelen utilizar técnicas de posprocesamiento de imágenes para hacer su trabajo más rápido. Aun así, este enfoque sigue siendo costoso y consume mucho tiempo en comparación con el potencial de una solución basada en el aprendizaje profundo. Actualmente se está empezando a integrar el uso de algoritmos de aprendizaje automático para analizar imágenes o datos recogidos por sensores [36, 54, 72]. Sin embargo, existe una oportunidad de mejora mediante el uso de técnicas más recientes de aprendizaje profundo como la segmentación semántica.

### 2.4.3. Detección y estimación de emisiones fugitivas contaminantes

Las emisiones fugitivas se refieren a la liberación accidental o intencional de sustancias contaminantes en el medio ambiente desde equipos, tuberías, válvulas o fugas en procesos industriales o sistemas de transporte. Estas emisiones pueden ocurrir durante la producción, almacenamiento, transporte o distribución de diferentes tipos de productos. Las emisiones fugitivas pueden contribuir de manera significativa a la contaminación del aire, agua y suelo, y pueden tener efectos negativos en la salud humana y en el medio ambiente en general. Por lo tanto, es importante implementar medidas de control y monitorización para minimizar la liberación de emisiones fugitivas.

La detección y estimación de emisiones fugitivas contaminantes [67] se refiere al proceso de detectar, localizar y estimar la contaminación en el aire emitida por diferentes fuentes, como fábricas, centrales térmicas, vehículos, etc. La prevención de la contaminación es una prioridad si se quiere preservar el medio ambiente. Las emisiones fugitivas contaminan el aire, poniendo en peligro la vida de las personas y los animales que viven cerca. Además, dependiendo de la composición de las emisiones, pueden contribuir al efecto invernadero. Por ejemplo, las emisiones de metano de las industrias del petróleo y el gas tienen un impacto 25 veces mayor que el dióxido de carbono [117].

En el estado del arte actual, existen varias técnicas y tecnologías utilizadas para la detección y estimación de emisiones contaminantes [57, 88, 122]. Algunas de las técnicas más comunes incluyen:

- **Medida directa:** mediante sensores y equipos de medición para medir directamente las emisiones de contaminación en el aire. Estos pueden ser equipos fijos instalados en estaciones de monitorización o móviles que se utilizan para medir las emisiones en diferentes puntos.
- **Inversión atmosférica:** Utiliza modelos matemáticos y mediciones de concentraciones de contaminación en el aire para estimar las emisiones de

contaminación de diferentes fuentes.

- **Técnicas de diagnóstico:** Utilizan técnicas como la espectroscopia, la termografía y la tomografía para detectar y medir las emisiones contaminantes.

Sin embargo, el costo de las soluciones tradicionales de detección de emisiones puede constituir un desafío para su adopción en entornos industriales. Por lo tanto, es necesario explorar opciones innovadoras y eficientes para mejorar la detección y la monitorización de emisiones en este contexto. La segmentación semántica, en particular, tiene un gran potencial en este campo, ya que permite clasificar imágenes a nivel de píxel. Esto puede ser útil a la hora de detectar emisiones anómalas o irregulares con mayor precisión. Esta propuesta tiene el potencial de abaratar costes ya que las cámaras tienen un bajo costo y cubren una gran cantidad de terreno. Aunque ya existen modelos de aprendizaje profundo capaces de detectar emisiones de humo y fuego mediante imágenes [95], la literatura científica en este campo es limitada, especialmente en su aplicación en entornos industriales.

## **Capítulo 3**

# **Experimentación e interpretación de resultados**

El objetivo de este capítulo es presentar y analizar la experimentación realizada a lo largo de esta Tesis Doctoral, así como explicar la metodología empleada y los criterios de evaluación utilizados. Se describirán los diferentes escenarios y casos de estudio considerados, así como las herramientas y técnicas aplicadas para el desarrollo y validación de los modelos propuestos. Finalmente, se interpretarán y discutirán los resultados obtenidos.

### 3.1. Agricultura de precisión

En esta sección se evalúa el potencial de las redes de segmentación semántica contra los métodos clásicos de aprendizaje máquina para tareas de agricultura de precisión. Para ello se compara el uso de imágenes obtenidas por UAV contra imágenes obtenidas por satélite, así como la utilidad del uso de índices multiespectrales de vegetación.

Para descubrir qué zonas son aptas para esta investigación, se utiliza una base de datos del uso del suelo conocida como “Sistema de Información Geográfica de Parcelas Agrícolas” o SIGPAC. Es una base de datos gratuita proporcionada por el gobierno español que permite identificar geográficamente las parcelas declaradas por los agricultores. Tiene definidas hasta treinta clases diferentes (ver Tabla 3.1). De esta forma se obtienen y descartan las regiones de interés para este estudio.

Tabla 3.1: Clases de SIGPAC.

Clase	Descripción	Clase	Descripción
CF	Cítricos	IM	Improductivo
CS	Peladuras de cítricos	IV	Invernaderos
CV	Viñedos de cítricos	OV	Olivar
FF	Frutales - Frutales de cáscara	OF	Olivar - Frutales
OC	Olivar - Cítricos	PS	Pastizal
CI	Cítricos	PR	Pasto arbustivo
AG	Cursos y superficies de agua	PA	Pastos con arbolado
ED	Edificaciones	TA	Tierras arables
EP	Elemento del paisaje	CA	Viales
FO	Bosques	VI	Viñedos
FY	Frutales	VF	Viñedo - Frutal
FS	Frutos secos	VO	Viñedo - Olivar
FL	Frutos secos y olivar	ZV	Área censurada
FV	Frutos secos y viñedo	ZC	Zona no incluida
TH	Huerta	ZU	Zona urbana

Una vez detalladas las regiones de interés, se obtienen las imágenes a través del satélite Sentinel-2 o del Plan Nacional de Ortofotografía Aérea (PNOA) [98]. El proceso de etiquetado se lleva a cabo utilizando las coordenadas de las parcelas de SIGPAC para colorear las regiones de las imágenes y generar las máscaras objetivo. Como las imágenes están georreferenciadas, este proceso se puede hacer de forma automática.



### 3.1.1. Reconocimiento de cultivos

Para evaluar el rendimiento del reconocimiento de diferentes tipos de cultivos mediante imágenes aéreas se evalúan y comparan las redes de segmentación semántica UNet y DeepLabV3+ contra Random Forest y Support Vector Machines. Para ello se crean varios conjuntos de datos de imágenes aéreas e información del suelo y se comparan los resultados obtenidos por imágenes obtenidas por UAVs y por satélite.

PNOA es una base de datos de ortofotografías aéreas digitales georeferenciadas que son accesibles de forma gratuita. La georreferencia es importante ya que permite fusionar datos de diferentes fuentes como SIGPAC. Cada ortofotografía tiene una resolución espacial (GSD) de 0,25 m/píxel y cubre una región equivalente a una página del Mapa Topográfico Nacional de España (MTN50). Una de las principales ventajas de PNOA, aparte de su gran resolución espacial, es que las imágenes no tienen nubes ni otros defectos. Sin embargo, tiene una gran desventaja para la monitorización periódica de cultivos ya que estos datos solo se actualizan una vez al año.

Usando las imágenes de PNOA se crea un nuevo conjunto de datos. A este conjunto se le llamará UOPNOA. UOPNOA consiste en 33.699 imágenes de  $256 \times 256$  píxeles. Estas imágenes son recortes de las imágenes PNOA que cubren una región equivalente a una página MTN50. Para comprobar que se ha generado correctamente la máscara objetivo, se compara con los datos oficiales del visor SIGPAC. El visor SIGPAC separa cada parcela pero como sólo se necesita el uso del suelo, no es necesario separar parcelas con el mismo tipo. En la Figura 3.1 se muestra una imagen de ejemplo así como la comprobación de su máscara.

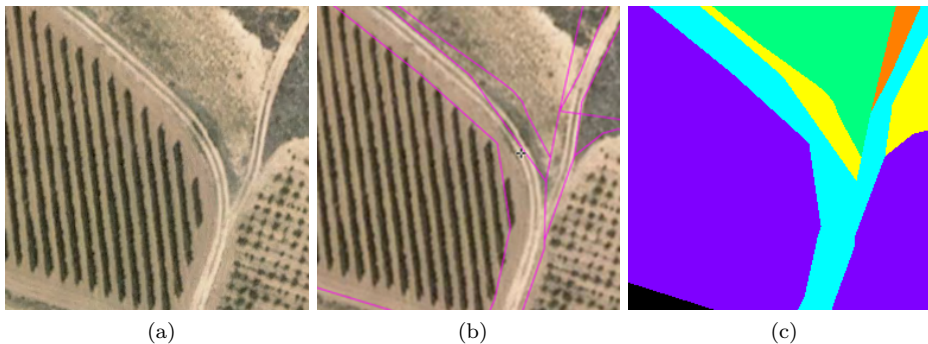


Figura 3.1: Imagen recortada y comprobación de la máscara objetivo para UOPNOA. (a) imagen recortada de  $256 \times 256$  píxeles de una imagen PNOA, (b) visor SIGPAC de la misma región para verificar la máscara, (c) máscara objetivo.

Según los requisitos de los posibles usuarios de este tipo de producto, no todas las clases son relevantes. De SIGPAC se extraen las clases mostradas en la Tabla 3.2. El número de parcelas de SIGPAC utilizadas para crear las máscaras objetivo de cada clase se presentan en la Tabla 3.2 junto con el número de píxeles del conjunto de datos UOPNOA. Una representación visual de la relación de los píxeles entre clases se muestra en la Figura 3.2. Se puede observar una gran diferencia entre la clase AR y el resto de las clases. Esta clase suele tener parcelas de mayor tamaño y ser más común que el resto de las clases.

Tabla 3.2: Clases del conjunto de datos UOPNOA.

Clase		Parcelas	Píxeles
UN	Improductivo	410	8.095.351
PA	Pastizales	1.883	46.738.269
SH	Pastos arbustivos	3.467	168.342.415
FO	Bosque	472	73.980.816
BU	Edificios y Zona urbana	125	7.562.696
AR	Tierra arables	4.935	968.106.602
GR	Pastos con árboles	220	93.208.409
RO	Carreteras y caminos	943	67.552.872
WA	Agua	340	19.776.364
FR	Frutas y nueces	93	4.150.529
VI	Viñedo	1.759	163.774.218

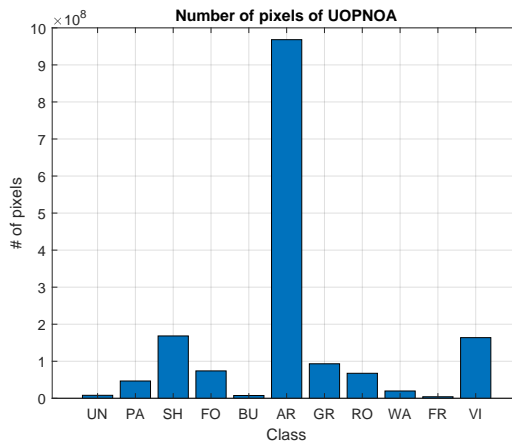


Figura 3.2: Tamaño en píxeles de cada clase de UOPNOA.

Otro conjunto de imágenes generado para esta investigación es el denominado UOS2, que consta de 1.958 imágenes de  $256 \times 256$  píxeles tomadas en julio de 2020 por el satélite Sentinel-2. Estas imágenes se combinan con los datos de SIGPAC para generar las máscaras objetivo.

Para obtener imágenes válidas de Sentinel-2, se buscan manualmente imágenes que no contengan nubes ni ningún otro defecto. Cuando una región de interés incluye algo que podría comprometer su calidad, se busca otra fecha en la que la imagen no tenga defectos. El proceso de anotación se llevó a cabo de manera similar al conjunto de datos UOPNOA, utilizando las coordenadas de las parcelas de SIGPAC para pintar la máscara correspondiente. La Figura 3.3 muestra un ejemplo de una de estas imágenes y su máscara. Para verificar que la máscara objetivo se ha generado correctamente, se compara con los datos oficiales del visor SIGPAC. El número de parcelas de SIGPAC utilizadas para crear las máscaras objetivo de cada clase se presenta en la Tabla 3.3 junto con el número de píxeles resultante. Dado que el conjunto de datos UOS2 cubre la misma región que el conjunto de datos UOPNOA, también se observa una gran diferencia entre la clase AR y el resto de las clases. En la Figura 3.4 se muestra una representación visual de la relación de píxeles entre clases.

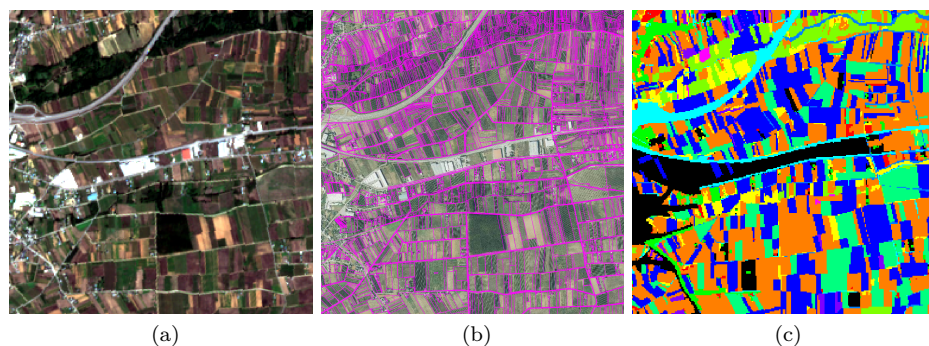


Figura 3.3: Imagen recortada y comprobación de la máscara de referencia para UOS2. (a) imagen recortada de  $256 \times 256$  píxeles de una imagen de Sentinel-2, (b) visor SIGPAC de la misma región para verificar la máscara, (c) máscara de referencia.

Tabla 3.3: Clases del conjunto de datos UOS2.

Clase	Parcelas	Píxeles
UN Improductivo	26.912	1.455.995
PA Pastizales	152.359	5.591.140
SH Pastos arbustivos	265.046	16.465.578
FO Bosque	48.977	10.746.146
BU Edificios y Zona urbana	207.839	1.047.900
AR Tierra arables	772.427	60.046.203
GR Pastos con árboles	28.649	4.494.801
RO Carreteras y caminos	73.250	3.719.742
WA Agua	16.802	1.733.641
FR Frutas y nueces	6.306	566.355
VI Viñedo	87.071	4.126.803

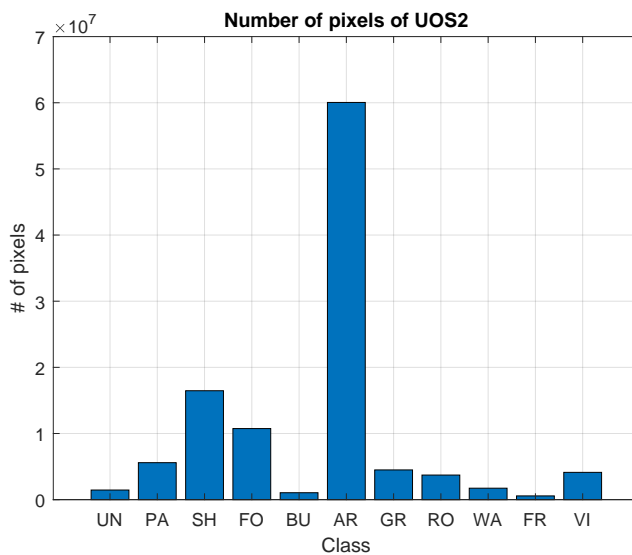





Figura 3.4: Tamaño en píxeles de cada clase de UOS2.

Los píxeles que no corresponden con ninguna de estas clases serán asignados a una clase genérica llamada “OT - Otros” o no serán utilizados, dependiendo del experimento (Base vs Multiuso). Esta clase no es relevante para el estudio y su único propósito es proporcionar una predicción realista que permita clasificar aquellos píxeles que no pertenecen a ninguna otra clase.

Para probar un menor número de clases más genéricas, se crean nuevas clases a partir de combinaciones de las anteriores. Esto permite experimentos más simples para estudiar como afecta la complejidad de las clases al entrenamiento de los modelos.

-  PASHGR—Todos los pastos
  
-  BURO—Toda la infraestructura
  
-  ARVI—Tierras arables y viñedos

#### 3.1.1.1. Resultados y discusiones

Para realizar una comparación justa y observar el rendimiento de los métodos propuestos, deben compararse con los métodos más comunes utilizados anteriormente: *Random Forest* y *Support Vector Machine*. Ambos métodos necesitan muchos menos datos que una red neuronal común. Para este trabajo de investigación se utilizó un conjunto de datos reducido de UOPNOA con 1,000 muestras (píxeles) por clase, con un total de 12.000 muestras. *Random Forest* logró una precisión general del 0,074% *Support Vector Machine* logró una precisión general del 0,073%. En la Tabla 3.4 se muestran los resultados en detalle. Como se puede observar, la OA de ambos métodos es similar, obteniendo un valor casi idéntico de 0,07%.

Las características utilizadas para estos experimentos consisten en los valores rojo, verde y azul (RGB) de cada píxel. Una experimentación adecuada con estos métodos debe incluir una ingeniería de características. Esto no es necesario con las redes neuronales, ya que generan sus propias características. Esta es una gran ventaja, aunque se necesita de un conjunto de datos de gran tamaño. Sin embargo, incluso con una ingeniería de características adecuada, los resultados de RF y SVM tienden a ser inferiores a los de una red neuronal. Siempre y cuando el conjunto de datos utilizado sea lo suficientemente grande y variable [14, 120, 131].

De esta forma, se observa que tanto UNet como DeepLabV3+ son capaces de reconocer los diferentes cultivos del conjunto de datos UOPNOA (ver Tabla 3.4). Aunque, en este caso, DeepLabV3+ supera a UNet por más de un 23% de  $F_1$ -Score. Parece que la mayor complejidad de la red supone una ventaja en el entrenamiento.

Tabla 3.4: Métricas globales de cada método para el conjunto de datos UOPNOA.

Experimentos	OA	R	P	IoU	F <sub>1</sub>
RF	0.074	0.114	0.130	0.034	0.121
SVM	0.073	0.138	0.130	0.035	0.134
UNet	0.830	0.618	0.473	0.473	0.536
DeepLabV3+	0.898	0.781	0.758	0.637	0.769

A continuación, se vuelven a repetir los experimentos de UNet y DeepLabV3+ con el conjunto de datos UOPNOA añadiendo la clase Otros. A partir de ahora, a este experimento de se conocerá como Experimento Multiuso. El experimento original pasará a llamarse Experimento Base.

En la Tabla 3.5 se presentan los resultados desglosados de UNet para cada clase. Se puede observar que hay un aumento considerable en la precisión cuando no se utiliza la clase de propósito general Otros. Esta clase tiene los peores resultados. El rendimiento del resto de las clases disminuye notablemente. Esto puede ser debido a la gran variabilidad de esta clase, dificultando su capacidad de discriminación del resto de clases.

Tabla 3.5: Métricas de cada clase de UNet en UOPNOA.

Clase	Experimento Base			Experimento Multiuso		
	R	P	IoU	R	P	IoU
UN	0.45	0.29	0.21	0.40	0.23	0.17
PA	0.54	0.25	0.20	0.36	0.21	0.15
SH	0.69	0.62	0.49	0.64	0.51	0.39
FO	0.21	0.78	0.20	0.52	0.63	0.40
BU	0.59	0.92	0.56	0.71	0.62	0.49
AR	0.93	0.96	0.90	0.88	0.75	0.68
GR	0.62	0.70	0.49	0.67	0.59	0.46
RO	0.77	0.51	0.45	0.79	0.37	0.34
WA	0.60	0.43	0.34	0.63	0.30	0.26
FR	0.45	0.92	0.43	0.54	0.70	0.44
VI	0.90	0.97	0.88	0.95	0.76	0.74
OT	-	-	-	0.15	0.47	0.13

En el caso de DeepLabV3+ (ver Tabla 3.6) también se reduce el rendimiento del resto de las clases al añadir una clase general como Otros. Es evidente que en la mayoría de los casos una clase general solo introduce confusión. Esto puede deberse a que esta clase es demasiado genérica y no posee patrones específicos. Sin embargo, con DeepLabV3+, y al contrario que con UNet, esta clase obtiene los mejores resultados de todas las clases. Pero este comportamiento no es de interés ya que lo que se busca es mejorar las clases objetivo.

Tabla 3.6: Métricas de cada clase de DeepLabV3+ en UOPNOA.

Clase	Experimento Base			Experimento Multiuso		
	R	P	IoU	R	P	IoU
UN	0.56	0.66	0.43	0.65	0.63	0.47
PA	0.79	0.78	0.65	0.55	0.48	0.27
SH	0.58	0.53	0.38	0.38	0.39	0.30
FO	0.84	0.84	0.73	0.63	0.68	0.49
BU	0.85	0.86	0.76	0.78	0.75	0.62
AR	0.94	0.97	0.92	0.76	0.82	0.65
GR	0.75	0.89	0.69	0.81	0.87	0.72
RO	0.84	0.60	0.54	0.82	0.75	0.65
WA	0.77	0.47	0.41	0.75	0.52	0.44
FR	0.66	0.73	0.53	0.46	0.56	0.34
VI	0.96	0.95	0.92	0.62	0.77	0.53
OT	-	-	-	0.86	0.86	0.76

A modo de conclusión, se añade la comparación de las métricas globales de los experimentos Base contra los experimentos Multiuso en la Tabla 3.7. Además, se añade una prueba con el conjunto de datos UOS2. Este conjunto solamente puede ser evaluado con UNet, ya que DeepLabV3+ solamente puede ejecutarse con tres canales y en este caso se utilizan 10 (se descartan aquellas bandas de 60 metros por píxel por su baja resolución). Esta prueba confirma el patrón seguido hasta ahora. La clase Otros resulta contraproducente.

Tabla 3.7: Comparativa del uso de la clase “Otros” en UOPNOA y UOS2.

Experimento	OA	R	P	IoU	F <sub>1</sub>
UNet Base	0.830	0.618	0.473	0.473	0.536
UNet Multiuso	0.641	0.607	0.391	0.391	0.476
DeepLabV3+ Base	0.898	0.781	0.758	0.637	0.769
DeepLabV3+ Multiuso	0.750	0.678	0.678	0.524	0.678
UNet UOS2 Base	0.647	0.569	0.428	0.323	0.489
UNet UOS2 Multiuso	0.527	0.521	0.364	0.259	0.429

Hasta ahora, se ha demostrado que el reconocimiento de cultivos con imágenes de PNOA es posible. Sin embargo, estas imágenes tienen una frecuencia demasiado baja como para poder ser utilizadas para un sistema de monitorización. Por este motivo, se profundiza en el estudio sobre el conjunto de datos UOS2 basado en imágenes de Sentinel-2. De esta forma, se puede comparar como afecta la disminución de la resolución espacial y el aumento de la resolución espectral a los modelos de entrenamiento.

Para evaluar la utilidad del aumento de la resolución espectral se realizan los siguientes experimentos: el experimento “RGB” para probar cómo se desempeña el conjunto de datos solo con bandas RGB y servir de comparación con PNOA, el experimento “ME”, que utiliza solo tres bandas multiespectrales (B8 NIR, B12 SWIR, B6 VRE) para el espectro infrarrojo con el visible, y dos experimentos adicionales utilizando seis y trece bandas para observar como afecta el aumento de información al modelo. Dado que los experimentos “Base” y “Multiuso” utilizan diez bandas, se puede hacer una comparación entre tres, seis, diez y trece bandas. Todos los experimentos excepto el experimento “Base” utilizan la clase “Otros”. Los resultados de estos experimentos se pueden observar en la Tabla 3.8.

Al utilizar solamente tres bandas, tanto los experimentos “RGB” como “ME” obtienen unos resultados insuficientes, con una *Recall* y *Precision* cercanas al 10%. Además, hay poca diferencia entre utilizar las bandas RGB o las tres bandas multiespectrales seleccionadas.

Los experimentos con seis, diez y trece bandas tienen resultados similares. Esto demuestra que se necesitan seis bandas y que tres bandas no proporcionan suficientes datos para diferenciar entre las clases en esta resolución espacial.

Una comparación entre el experimento “Base” de UNet en UOPNOA (0,61 R y 0,47 P) y el mejor experimento de UNet en UOS2 (0,56 R y 0,42 P) revela que UOPNOA tiene mejores resultados. Por lo tanto, la resolución espacial es el factor más importante de un conjunto de datos de imágenes aéreas. El siguiente factor más importante es tener un rango espectral de mayor tamaño.

Tabla 3.8: Métricas globales para diferentes conjuntos de bandas con UNet en UOS2.

Experimento	OA	R	P	IoU	F <sub>1</sub>
Base	0.647	0.569	0.428	0.323	0.489
Multiuso	0.527	0.521	0.364	0.259	0.429
RGB	0.585	0.090	0.079	0.053	0.084
ME	0.570	0.091	0.109	0.052	0.099
6 bandas	0.569	0.480	0.373	0.252	0.420
13 bandas	0.589	0.463	0.371	0.261	0.412



La Tabla 3.9 muestra el desglose de clases para los experimentos con mejores resultados: Base y Multiuso. Se puede apreciar que hay ciertas clases con muy buenos resultados como AR, VI, BU, y FO. Sin embargo, hay clases con métricas muy bajas como FR, RO, WA, y UN. Esta disparidad es algo que llama la atención e invita a valorar la complejidad de las clases, así como el número de muestras de cada una.

Tabla 3.9: Métricas de cada clase para el mejor experimento de UNet en UOS2 (Base y Multiuso).

Clase	Experimento Base			Experimento Multiuso		
	R	P	IoU	R	P	IoU
UN	0.56	0.22	0.19	0.51	0.22	0.18
PA	0.42	0.28	0.20	0.46	0.21	0.17
SH	0.36	0.67	0.30	0.37	0.54	0.28
FO	0.59	0.60	0.43	0.52	0.48	0.34
BU	0.81	0.50	0.45	0.74	0.39	0.34
AR	0.77	0.94	0.73	0.69	0.85	0.62
GR	0.62	0.37	0.30	0.58	0.28	0.23
RO	0.36	0.16	0.12	0.40	0.13	0.11
WA	0.61	0.22	0.19	0.63	0.17	0.16
FR	0.34	0.10	0.08	0.45	0.08	0.07
VI	0.78	0.59	0.51	0.76	0.56	0.48
OT	-	-	-	0.09	0.38	0.08

Para reducir la complejidad de las clases, se ha decidido fusionar las clases pertenecientes a los pastos (PA+SH+GR), las infraestructuras y caminos (BU+RO), y las tierras arables con los viñedos (AR+VI). El resto de clases se descartan por su bajo número de muestras. De esta forma, el conjunto de datos resultante está más equilibrado y tiene una menor complejidad. De esta forma, se repitieron los experimentos “Base” y “Multiuso” para comparar con experimentos anteriores. Los resultados globales se pueden observar en la Tabla 3.10. Estos resultados coinciden con experimentos anteriores en los que el experimento Base supera al experimento Multiuso. La fusión y reducción del número de clases mejora drásticamente la precisión de las predicciones, obteniendo una mejora de casi el 22% y el 25% en *Recall* y *Precision*, respectivamente, para los mejores experimentos.

Al observar los resultados por clase de la Tabla 3.11, se pueden ver una *Recall* y *Precision* excepcionalmente altas para cada clase excepto BURO.

Tabla 3.10: Métricas globales para clases simplificadas con UNet en UOPNOA.

<b>Experimento</b>	<b>OA</b>	<b>R</b>	<b>P</b>	<b>IoU</b>	<b>F<sub>1</sub></b>
Base	0.822	0.786	0.677	0.576	0.727
Multiuso	0.650	0.639	0.578	0.402	0.607

Tabla 3.11: Métricas de clase para clases simplificadas con UNet en UOS2.

<b>Clase</b>	<b>Experimento Base</b>			<b>Experiment Multiuso</b>		
	<b>R</b>	<b>P</b>	<b>IoU</b>	<b>R</b>	<b>P</b>	<b>IoU</b>
PASHGR	0.82	0.81	0.69	0.71	0.52	0.43
BURO	0.70	0.25	0.23	0.70	0.17	0.16
ARVI	0.83	0.95	0.80	0.77	0.85	0.68
OT	-	-	-	0.36	0.75	0.21

Para evaluar los resultados de una forma visual, se incluyen las detecciones realizadas por los mejores experimentos. La Figura 3.5 incluye las detecciones de UOPNOA con DeepLabV3+. La Figura 3.6 muestra las detecciones de UNet en UOS2 para todas las clases objetivo. Finalmente, la Figura 3.7 muestra los resultados de UNet en UOS2 para las clases simplificadas. De estas imágenes, se destaca que, especialmente en UOS2 con UNet, los caminos naturales y los lindes de las parcelas se detectan como caminos. Estas detecciones se presentan en las métricas como errores, ya que no están registrados como tal en la base de datos SIGPAC. Sin embargo, esto puede ser un comportamiento deseado que supera a las máscaras objetivo. Esto explica los resultados reducidos de esta clase. Además, este fenómeno también provoca una bajada en el resto de clases ya que se incluyen algunos píxeles de otras clases.

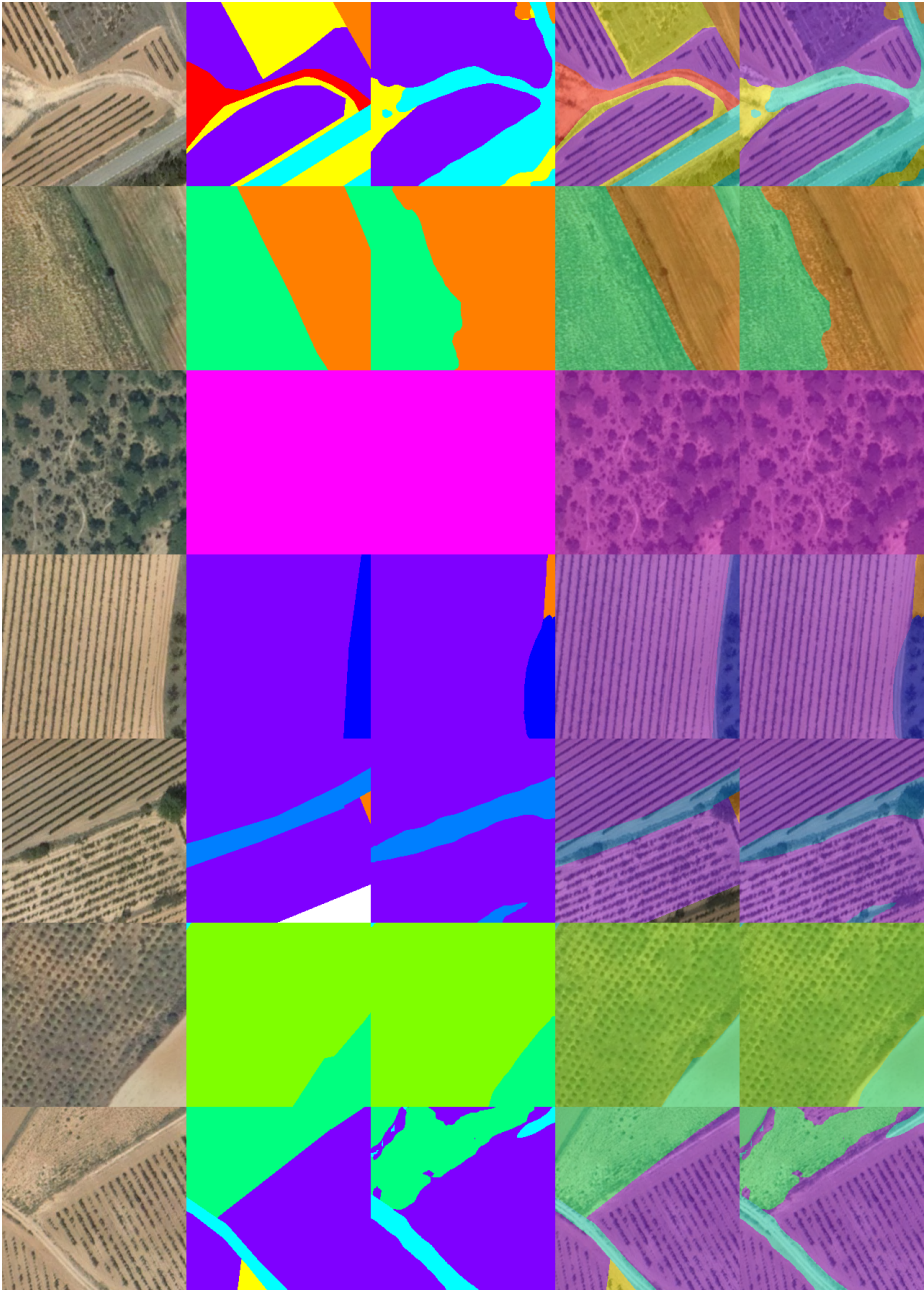


Figura 3.5: Visualización de los resultados para DeepLabV3+ evaluado en UOP-NOA (Experimento Base). (**1<sup>a</sup> col.**) Imágenes originales, (**2<sup>a</sup> col.**) máscaras de referencia, (**3<sup>a</sup> col.**) predicciones, (**4<sup>a</sup> col.**) superposición con máscaras de referencia, (**5<sup>a</sup> col.**) superposición con predicciones.

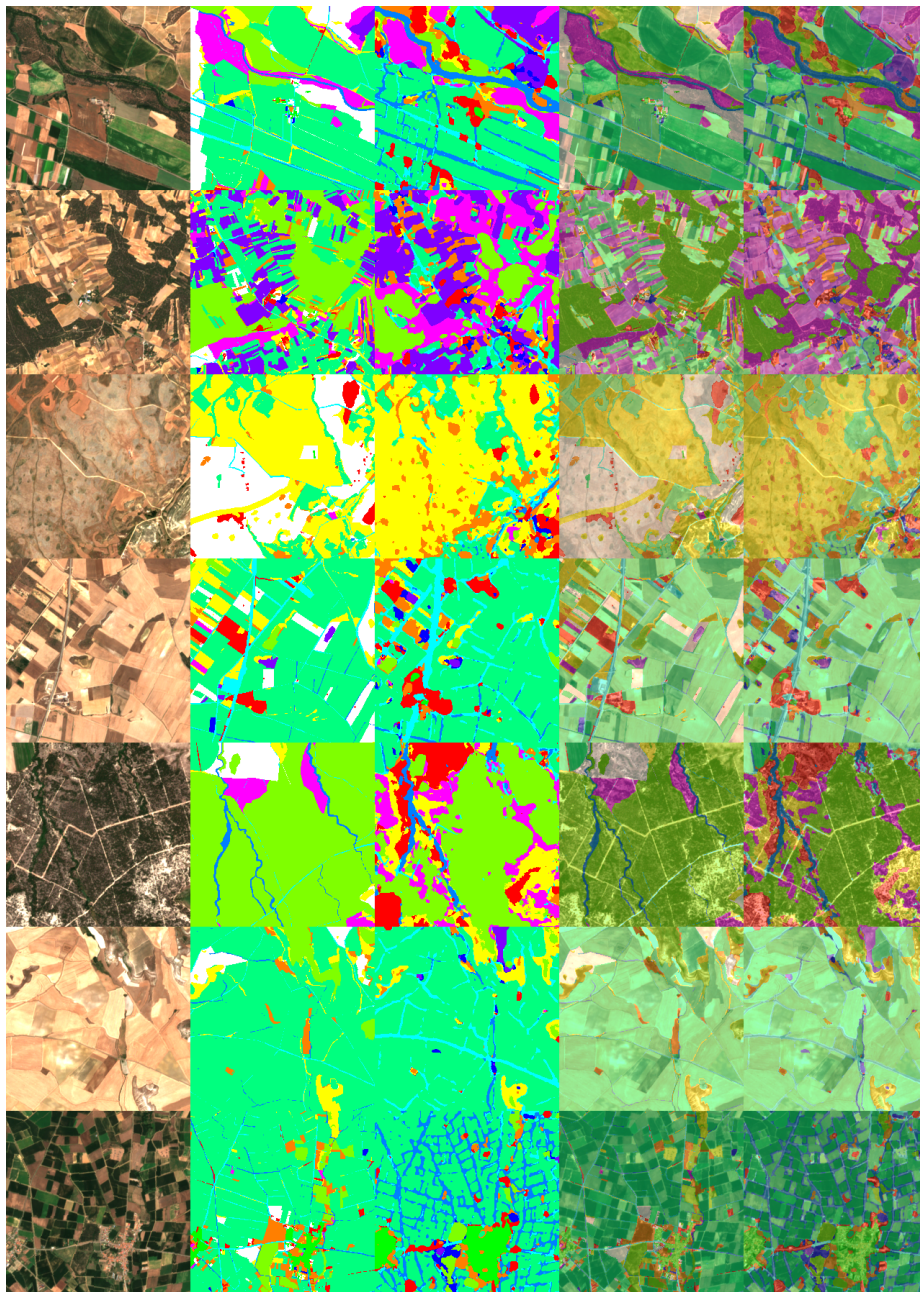


Figura 3.6: Visualización de los resultados para UNet en UOS2 (Experimento Base). (1<sup>a</sup> col.) Imágenes originales, (2<sup>a</sup> col.) máscaras de referencia, (3<sup>a</sup> col.) predicciones, (4<sup>a</sup> col.) superposición con máscaras de referencia, (5<sup>a</sup> col.) superposición con predicciones.

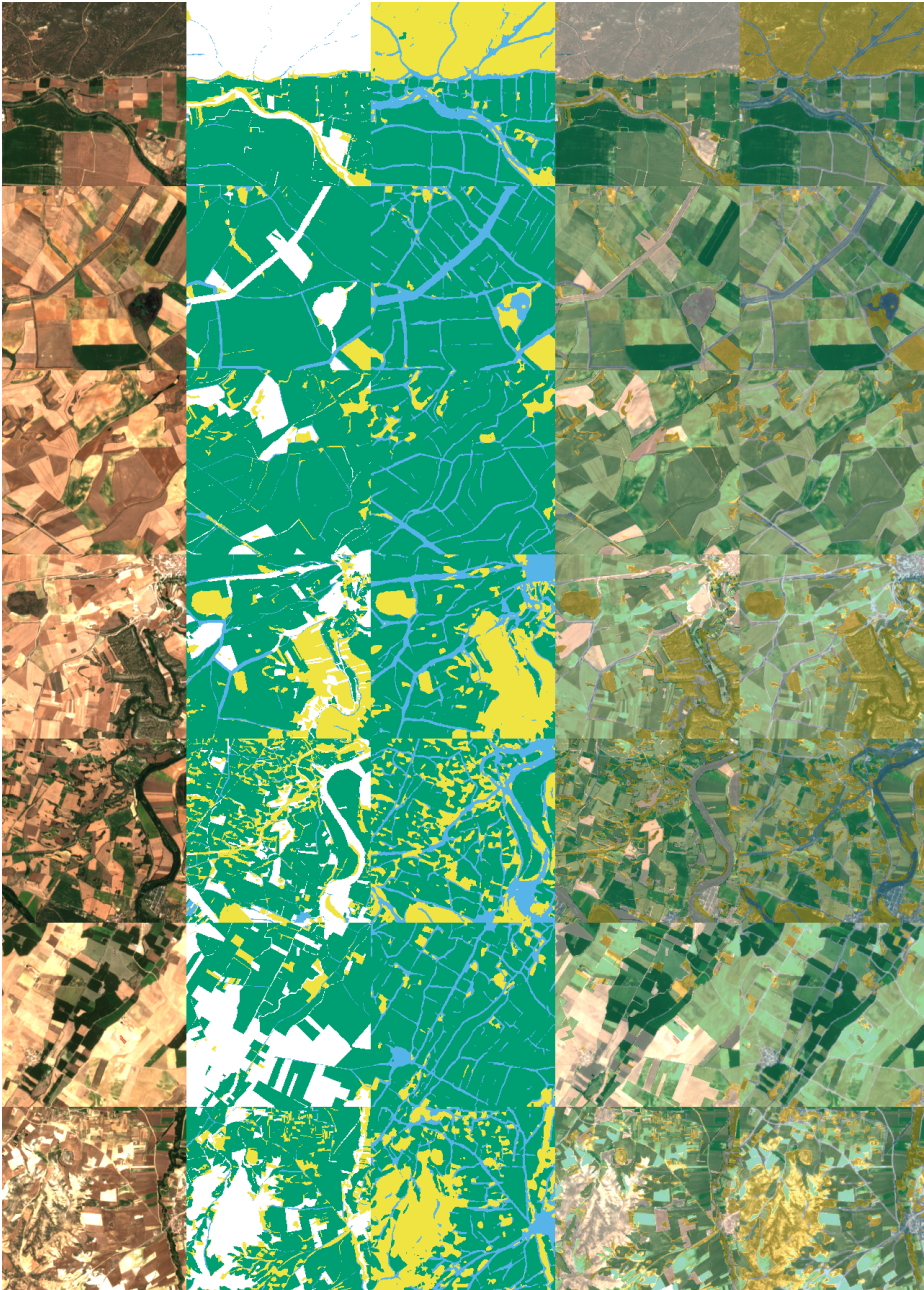


Figura 3.7: Visualización de los resultados para UNet en UOS2 para cuatro clases simplificadas (Experiment Base). (1<sup>a</sup> col.) Imágenes originales, (2<sup>a</sup> col.) máscaras de referencia, (3<sup>a</sup> col.) predicciones, (4<sup>a</sup> col.) superposición con máscaras de referencia, (5<sup>a</sup> col.) superposición con predicciones.

Después de entrenar los modelos, se crea un prototipo para la inferencia con una arquitectura de microservicios que sigue los estándares modernos usando Flask. Para asegurarse de que se cumplen todas las dependencias requeridas y para hacer más fácil el despliegue, se usa Docker para contenerizar el microservicio. Luego, se despliega en diferentes infraestructuras para probar el rendimiento. El modelo seleccionado para este despliegue es la arquitectura UNet con clases simplificadas del experimento Base. La herramienta de aprendizaje profundo utilizada para la inferencia es PyTorch.

Las redes de segmentación semántica solo pueden operar con la misma resolución utilizada para el entrenamiento cuando se realiza la inferencia. Esto significa que la única forma de predecir imágenes más grandes es recortándolas y luego fusionando las predicciones. Cada imagen recortada tomará el mismo tiempo porque la predicción se realiza píxel a píxel, por lo que se realizan el mismo número de predicciones para cada imagen. Por esta razón, el método utilizado para probar el rendimiento se basa en el número de imágenes solicitadas por petición. Estas imágenes tienen la misma resolución que las utilizadas para el entrenamiento.

La mayoría de las herramientas utilizadas en los Sistemas de Información Geográfica están diseñadas para trabajar con *geojson* ya que este formato es más fácil de trabajar que una máscara. Para agregar realismo al servicio, se agrega un proceso para convertir la salida de la red a un *geojson* georreferenciado con cada región y tipo de uso de la tierra predicho en él. Esto agrega muchas computaciones al prototipo ya que debe poligonizar la máscara de la salida de la inferencia y convertirla en un *geojson*. Para limitar el tamaño del *geojson*, se ejecuta el algoritmo de *Ramer-Douglas-Peucker* (RDP) [102] para simplificar el número de puntos que definen los polígonos. Este proceso se adapta para aprovechar múltiples núcleos de CPU.

Para la evaluación del rendimiento, se definen los tiempos de la siguiente manera: “Carga” es el tiempo necesario para acceder a las bibliotecas, cargar las imágenes en la memoria, etc. “Predicción” es el tiempo necesario para que el modelo realice las predicciones. “Resultados” es el tiempo necesario para convertir las predicciones en formato *geojson*. Para esta tarea, se mide el tiempo de poligonización de las máscaras de predicción, la simplificación de los resultados utilizando el algoritmo RDP y la generación de los *geojsons*. “Red” es el tiempo necesario para cargar las imágenes a ser procesadas y el tiempo necesario para descargar las predicciones y sus *geojsons*.

La Figura 3.8 presenta el rendimiento para una sola imagen. Para comparar el rendimiento entre infraestructuras, se muestra la relación de latencia en la parte superior de cada barra apilada. Esta relación se calcula como el tiempo total de un experimento dado dividido por el tiempo total del mejor experimento. En la Figura 3.9 se muestran las métricas de coste-rendimiento para diez imágenes. “Local:A-B” son las máquinas utilizadas para entrenar las redes en este trabajo.

Estas máquinas constan de un procesador Intel® Core™ i9-11900K y de una tarjeta de procesamiento NVIDIA GeForce RTX 2080 Ti, además, están conectadas a través de una LAN de 1 Gbit. El resto de las infraestructuras son proporcionadas por Microsoft Azure. Se evalúan las implementaciones IaaS, CaaS y FaaS de Azure. En todos los casos, el cliente es una máquina local desde fuera de la red de Azure con una conectividad de aproximadamente 250 Mbps.

En la Figura 3.8, se puede observar una mejora mínima al utilizar GPU. Los tiempos de “Carga” son despreciables. “Predicción” depende de la velocidad de un solo núcleo y del número de núcleos. Sin embargo, el uso de una GPU siempre es más rápido. “Resultados” tiene el mayor impacto en el tiempo, dependiendo principalmente de la velocidad de un solo núcleo. “Red” no se ve afectada por la infraestructura de cálculo utilizada, excepto en el caso de “Local:A-B”, donde ambas máquinas están en la misma red local.

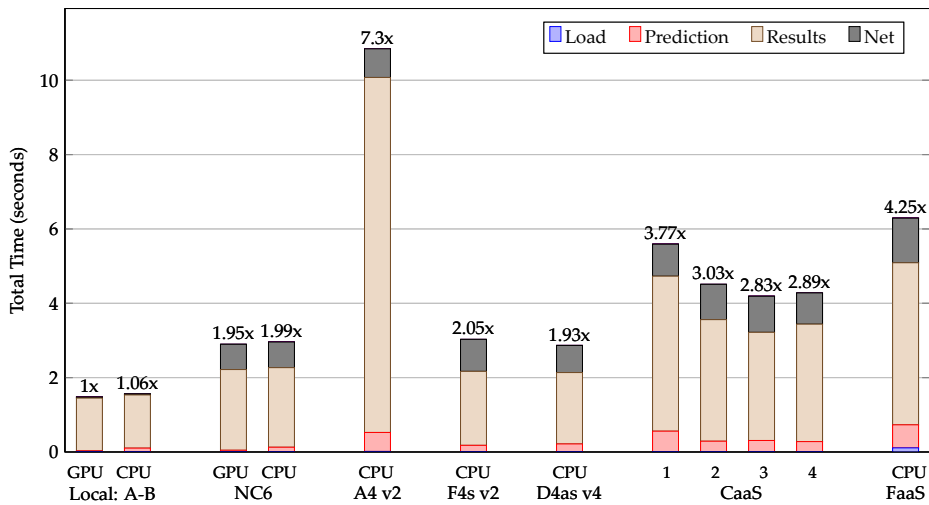


Figura 3.8: Rendimiento del procesamiento de imagen en diversas plataformas.

La Figura 3.9 muestra una mejora casi nula al utilizar una GPU, pero el coste se multiplica. Debe hacerse un compromiso entre el coste y el rendimiento, ya que CaaS es más económico, pero las soluciones IaaS como “D4as v4” y “F4s v2” tienen un mejor rendimiento. “Local:A-B” parece ser la mejor opción incluso cuando se tienen en cuenta los costes de electricidad y la amortización a lo largo de cinco años. La configuración y el mantenimiento de este tipo de infraestructura pueden ser excesivamente complejos. Sin embargo, las opciones en la nube reciben actualizaciones de hardware periódicamente. El coste de FaaS se calcula como si hubiera una petición en ejecución durante toda la hora. Esto significa que el coste es cero si no hay peticiones, pero sería mayor si hubiera múltiples peticiones simultáneas que provocaran la ejecución de múltiples instancias de FaaS al mismo tiempo, multiplicando su coste. El resto de las infraestructuras se cotizan por disponibilidad y no por uso: pueden procesar múltiples peticiones al mismo coste.

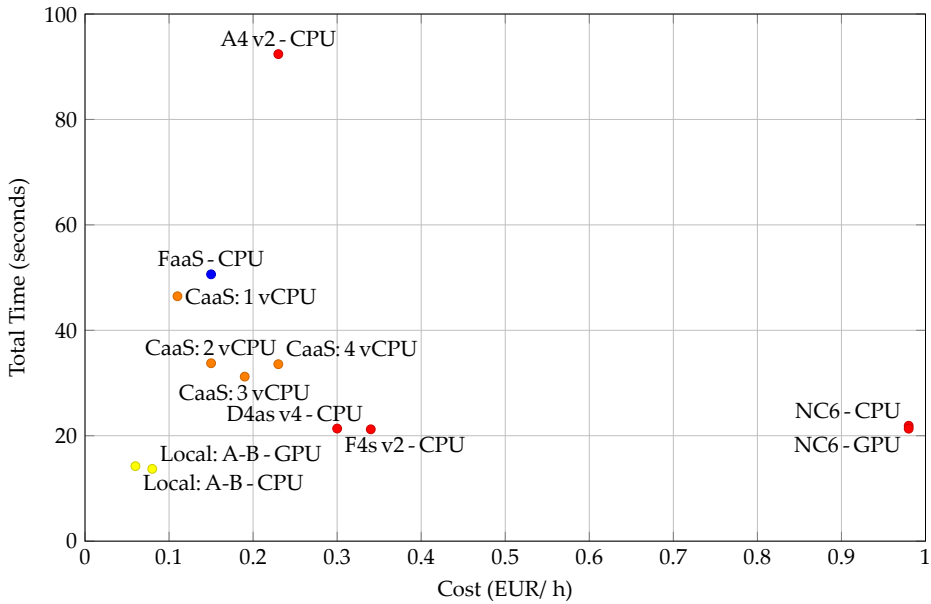


Figura 3.9: Rendimiento de coste por infraestructura para diez imágenes. Los precios y la potencia de cálculo están sujetos a cambios (abril de 2021).



### 3.1.2. Detección de campos recientemente abonados

El uso de fertilizantes en la agricultura puede contaminar el medio ambiente y afectar la calidad del agua y del aire. La lixiviación del nitrógeno en cuerpos de agua cercanos es uno de los problemas más discutidos. Se han propuesto soluciones, como el uso de períodos cerrados en los que los agricultores no pueden utilizar fertilizantes con alto contenido de nitrógeno, pero su cumplimiento es difícil de monitorizar en áreas extensas. El uso de satélites para la detección de campos recientemente abonados parece una solución viable. Sin embargo, aún no se han desarrollado métodos específicos para detectar el uso de fertilizantes en diferentes tipos de terrenos utilizando esta tecnología. La detección del uso de fertilizantes en diferentes tipos de campos mediante el uso de imágenes de satélite multiespectrales podría mejorar la monitorización y gestión del fertilizante en tiempo real, y fomentar leyes que prohíban su uso en temporadas de riesgo.

No se encontraron conjuntos de datos públicos en la literatura que pudieran ser utilizados para este estudio. Por esta razón, se generó manualmente un conjunto de datos. El satélite elegido para las imágenes es Sentinel-2, debido a su rango espectral, resolución espacial y, sobre todo, su corto tiempo de revisita de 5 días.

Para aprovechar la información temporal de Sentinel-2, se decidió utilizar, además de la imagen inmediatamente después de la aplicación del abono, la imagen inmediatamente anterior. De esta manera, el cambio entre las dos imágenes puede aportar información significativa sobre el suelo y mejorar los resultados. Esto duplica el número de características por píxel.

El conjunto de datos contiene imágenes del satélite Sentinel-2 y máscaras objetivo que señalan las áreas donde se aplicó abono. Las imágenes incluyen canales con diferentes bandas espectrales e índices multiespectrales de vegetación. Cada canal de la imagen es una característica para cada píxel. Para generar la mayor cantidad posible de características relevantes, se estudian los índices de vegetación más comunes en la literatura de agricultura de precisión (ver Tabla A.1 en Apéndice A). Las máscaras se han generado manualmente, mediante una investigación in situ para confirmar que la parcela había sido abonada recientemente y para observar las dimensiones reales de la fertilización. Las máscaras también han sido procesadas para eliminar píxeles innecesarios, como carreteras o edificios mediante el uso de la base de datos SIGPAC. Además, se han añadido contraejemplos para entrenar modelos de detección binaria. El conjunto de datos se ha dividido en dos clases: “Aplicación de abono” y “Otros”.

En este estudio, se visitaron 38 parcelas recién abonadas en tierras dedicadas a la ganadería en el norte de España, y se validaron manualmente mediante investigaciones in situ. Ocho de las parcelas fueron descartadas por no ser adecuadas para el estudio, ya sea por su pequeño tamaño, la falta de una fecha específica de aplicación de abono o porque las imágenes estaban demasiado nubladas en ese período. Se dispone de imágenes de 30 parcelas, que tienen un área de aproximadamente  $1.700 \times 1.700$  m. La Tabla 3.12 muestra la identificación de cada parcela, la fecha de aplicación de abono, su área en  $m^2$ , y sus coordenadas geográficas en longitud y latitud. El conjunto de datos completo consiste de 225,94 hectáreas, de las cuales 31,48 pertenecen a la clase “Aplicación de abono” y 195,46 a la clase Otros, que sirve como clase contraejemplo. Cada píxel representa 0,01 hectáreas. El conjunto de datos está dividido por parcelas en lugar de por píxeles para evitar que los conjuntos de entrenamiento y prueba tengan píxeles cercanos entre sí. Esto reduce la posibilidad del sobreajuste del modelo. Para el entrenamiento, se utilizan el 70 % de las parcelas disponibles (21 de las 30 parcelas), con un total de 180,79 hectáreas (22,28 hectáreas de la clase objetivo Aplicación de abono y 158,51 hectáreas para la clase Otros). Para evaluar el rendimiento de los modelos, se utiliza el 30 % restante de las parcelas (9 de las 30 parcelas), con un total de 46,15 hectáreas (9,20 hectáreas de la clase objetivo y 36,95 hectáreas de la clase Otros). El conjunto de datos esta disponible para su descarga <sup>1</sup>.

Para ubicar cada parcela, se realizaron investigaciones in situ para confirmar que las parcelas habían sido recientemente abonadas y observar las dimensiones reales de su fertilización (ver Figura 3.10a). Luego, utilizando Google Earth Engine, se anotó cada parcela de acuerdo con las dimensiones observadas, como se muestra en la Figura 3.10b. La anotación se utilizó luego para generar la máscara objetivo en formato raster utilizando la imagen georreferenciada de Sentinel-2 de base, como se muestra en la Figura 3.11.

---

<sup>1</sup>DOI del conjunto de datos: [10.17632/fbvvf55kp.1](https://doi.org/10.17632/fbvvf55kp.1)

Tabla 3.12: Parcelas en el conjunto de datos.

<b>Parcela</b>	<b>Fecha (AAAA/MM/dd)</b>	<b>Área (m<sup>2</sup>)</b>	<b>Long./Lat.</b>
P-BLD	2022/05/26	8.900	-4.2018, 43.3973
P-BLLT1	2022/05/16	21.200	-4.0840, 43.4309
P-BLLT2	2022/05/26	3.300	-4.0840, 43.4310
P-Cardana	2022/02/24	6.500	8.6580, 45.8592
P-CBRCS1	2022/05/26	6.700	-4.2005, 43.3897
P-CBRCS2	2022/05/26	6.400	-4.2048, 43.3875
P-CLGT	2022/05/16	17.200	-4.1096, 43.3987
P-CLMBRS	2022/05/26	4.300	-4.5447, 43.3804
P-CMNTR	2022/05/16	2.600	-4.1470, 43.4001
P-DR	2022/03/21	2.500	-4.1424, 43.3967
P-FNFR	2022/05/16	10.100	-4.2657, 43.3880
P-LLT	2022/05/03	9.600	-4.1515, 43.4001
P-LNDRS1	2022/05/16	3.200	-4.2510, 43.3880
P-LNDRS2	2022/05/16	5.400	-4.2503, 43.3880
P-LNDRS3	2022/05/16	8.500	-4.2497, 43.3872
P-LNDRS4	2022/05/16	9.100	-4.2467, 43.3877
P-MT	2022/05/04	19.900	-4.1536, 43.3980
P-NMS	2022/02/10	5.500	-4.1490, 43.4003
P-QNTLS2	2022/05/16	8.500	-5.5840, 43.5458
P-SNTLLN	2022/03/17	14.200	-4.1170, 43.3935
P-SNVCNT1	2022/05/16	6.700	-4.4048, 43.3939
P-SNVCNT2	2022/05/16	29.200	-4.4001, 43.3945
P-STBN	2022/05/04	11.300	-4.1366, 43.3960
P-TGL2	2022/05/16	12.300	-4.0701, 43.4276
P-TNNS1	2022/05/26	19.500	-4.1871, 43.3996
P-TNNS2	2022/05/26	15.800	-4.1918, 43.3987
P-VG1	2022/04/09	12.200	-5.4866, 43.4699
P-VG2	2022/04/13	4.900	-5.4801, 43.4693
P-VLDMR	2022/02/07	17.500	-4.1561, 43.4056
P-VNS	2022/04/23	16.600	-4.1504, 43.4042

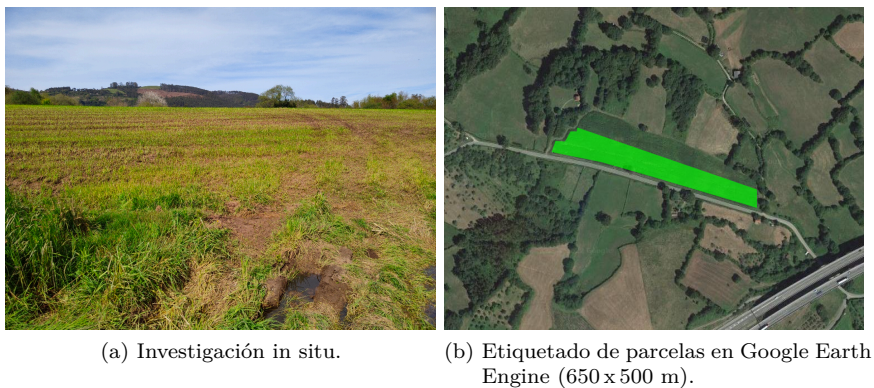


Figura 3.10: Ejemplo de etiquetado (P-VG1).

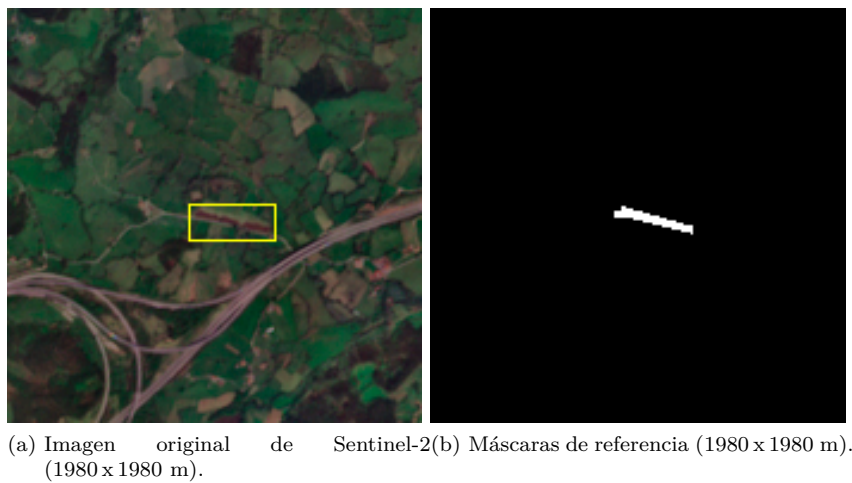


Figura 3.11: Ejemplo de máscaras de referencia (P-VG1).

Sobre este conjunto de datos se evalúan los métodos más comunes del aprendizaje automático: *Decision Tree*, *Discriminant Analysis*, *Logistic Regression*, *Naïve Bayes*, *Support Vector Machines*, *Nearest Neighbor Classifiers*, *Kernels Approximation*, *Ensemble*, y *Neural Network*.

Para visualizar las máscaras de abono detectadas, se utilizan técnicas morfológicas para reducir/eliminar el ruido. Debido a que la detección se realiza píxel por píxel, es común que estas máscaras sean ruidosas. Este paso solo se realiza en la visualización final y no afecta los resultados de las métricas. Para eliminar el ruido de la máscara, se utilizan dos técnicas: una erosión utilizando una estructura cuadrada de  $2 \times 2$  para borrar todos los píxeles flotantes, lo que hace que las regiones sean más pequeñas; y una dilatación utilizando una estructura de cruz de  $3 \times 3$  para restaurar el tamaño de las regiones.

Se analizan los 51 índices de vegetación generados en diferentes momentos, antes y después de la aplicación del abono. El propósito de este análisis es determinar si la elección de los índices de vegetación es correcta basándose en su correlación con la aplicación de abono. La Figura 3.12 muestra un mapa de calor con los 51 índices de vegetación para seis de las parcelas para representar la intensidad de sus valores. Los valores de cada fila se escalan a los valores mínimos y máximos. La primera imagen después de aplicar el abono se etiqueta como imagen 0 en el eje X. Las imágenes se clasifican como negativas o positivas dependiendo de si la aplicación de abono ocurrió antes o después de ellas. Para ayudar en la visualización, se coloca una línea vertical de puntos rojos al comienzo de la primera imagen después de la fecha de aplicación del abono. En el día de aplicación del abono, la intensidad de la mayoría de los índices de vegetación en la Figura 3.12 es casi cero, y a medida que pasa el tiempo, los valores aumentan. Los índices restantes también parecen estar inversamente asociados, con valores cercanos a 0 antes de la aplicación del abono y cerca de 1 después de la aplicación del abono. Solo un pequeño número de índices de vegetación parece no estar relacionado con la aplicación del abono. Por lo tanto, parece que los índices utilizados ofrecen información útil sobre la presencia de abono en las parcelas.

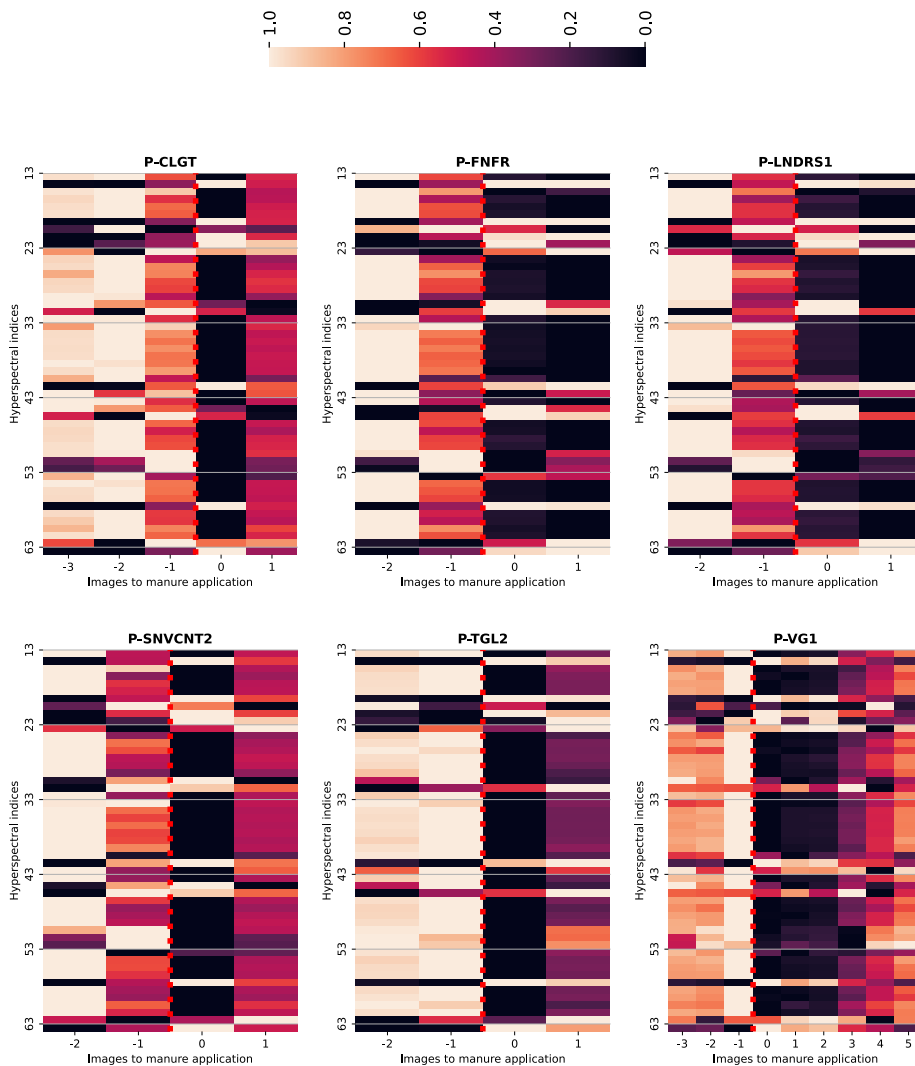


Figura 3.12: Respuesta de los índices de vegetación antes y después de la aplicación de estiércol.

A veces tener demasiadas características puede ser contraproducente, ya que disminuye la precisión del modelo de clasificación al aumentar su dimensionalidad. Por esta razón, y debido a que múltiples características podrían ser redundantes, se evalúan diferentes enfoques para la selección de características: el método Boruta [65], el método de Análisis de Componentes Principales (*Principal Component Analysis*, PCA) [127] utilizando el 95% de la variabilidad; el método de Eliminación de Características Recursivas (*Recursive Feature Elimination*, RFE) [47]; y el algoritmo de agrupamiento Ward utilizando la Ecuación (3.1), donde  $A$  y  $B$  representan los clústeres, mientras que  $cA$  y  $cB$  son sus respectivos centroides.

$$d(A, B) = \frac{\sqrt{2|A||B|}}{|A| + |B|} |cA - cB| \quad (3.1)$$

Además de los métodos de selección de características, este trabajo evalúa las bandas crudas de Sentinel-2, las bandas RGB, las bandas de infrarrojo (IR) y los índices de vegetación por separado. Para cada conjunto de características, se entrenan y evalúan los métodos de clasificación más conocidos del aprendizaje máquina.

La Tabla 3.13 describe todos los conjuntos de características a evaluar y los asocia con un identificador. Es importante tener en cuenta que la selección de características Boruta no descarta ninguna característica, lo que significa que el experimento que utiliza todas las características (BA-128) es el mismo y no necesita ser repetido. La Eliminación de Características Recursiva ha encontrado que el número óptimo de características es 90. Por esta razón, solo se muestra el conjunto de características BA-90-RFE para este método de selección.

Tabla 3.13: Descripción de todos los conjuntos de características.

Conjunto	Descripción
A-13-S2	Todas las bandas de Sentinel-2 usando la primera imagen después de la aplicación de estiércol.
BA-26-S2	Todas las 13 bandas de Sentinel-2 de la primera imagen después de la aplicación de estiércol y la imagen inmediatamente anterior, en total 26 características.
BA-6-RGB	Las bandas RGB de la primera imagen después de la aplicación de estiércol y la imagen inmediatamente anterior, en total 6 características.
BA-16-IR	Todas las bandas infrarrojas de la primera imagen después de la aplicación de estiércol y la imagen inmediatamente anterior, en total 16 características.
A-51-VI	Todos los índices de vegetación utilizando la primera imagen después de la aplicación de estiércol, en total 51 características.
BA-102-VI	Todos los índices de vegetación utilizando la primera imagen después de la aplicación de estiércol y la imagen anterior, en total 102 características.
A-64	Todas las 64 características (bandas e índices de vegetación) de la imagen inmediatamente después de la aplicación de estiércol.
BA-128	Todas las 64 características (bandas e índices de vegetación) de la imagen inmediatamente después de la aplicación de estiércol y la imagen anterior, en total 128 características.
A-7-W	7 de las 64 características (bandas e índices de vegetación) de la imagen inmediatamente después de la aplicación de estiércol. Las 7 características fueron seleccionadas manualmente utilizando el método de <i>clustering Ward</i> .
BA-14-W	7 de las 64 características generadas (índices de vegetación y bandas) de la imagen inmediatamente después de la aplicación de estiércol y la imagen inmediatamente anterior. Cada imagen tiene las mismas 7 características seleccionadas manualmente utilizando el método de <i>clustering Ward</i> , en total 14 características por píxel.
BA-95 %-PCA	Componentes de PCA que extraen el 95 % de la variabilidad de las 128 características (bandas e índices de vegetación) de la imagen inmediatamente después de la aplicación de estiércol y la imagen inmediatamente anterior.
BA-90-RFE	90 de las 128 características (bandas e índices de vegetación) de la imagen inmediatamente después de la aplicación de estiércol y la imagen inmediatamente anterior. Las 90 características fueron seleccionadas mediante el método de eliminación recursiva de características (RFE).



### 3.1.2.1. Resultados y discusiones

La Tabla 3.14 muestra los mejores resultados para cada conjunto de características. Todos los modelos están entrenados utilizando el conjunto de entrenamiento y todas las métricas se calculan utilizando el conjunto de prueba. El mejor conjunto de características es BA-102-VI. BA-90-RFE muestra resultados muy similares, sin embargo, la *Precision* media y la *Recall* media están ligeramente más equilibradas en el conjunto de características BA-102-VI.

Tabla 3.14: Resultados de cada conjunto de características.  $mP=Precision$  media,  $mR=Recall$  media,  $mF_1=F_1$ -Score medio.

Características	Mejor método	$mP$	$mR$	$mF_1$	OA
A-13-S2	<i>Support Vector Machines</i>	.929	.799	.859	.915
BA-26-S2	<i>Discriminant Analysis</i>	.951	.844	.894	.935
BA-6-RGB	<i>Neural Network</i>	.869	.823	.846	.906
BA-16-IR	<i>Support Vector Machines</i>	.912	.863	.887	.931
A-51-VI	<i>Support Vector Machines</i>	.930	.839	.882	.928
BA-102-VI	<i>Discriminant Analysis</i>	.952	.916	.934	.959
A-64	<i>Ensemble</i>	.948	.822	.881	.927
BA-128	<i>Discriminant Analysis</i>	.935	.907	.921	.951
A-7-W	<i>Naïve Bayes</i>	.825	.824	.824	.888
BA-14-W	<i>Support Vector Machines</i>	.882	.814	.847	.908
BA-95 %-PCA	<i>Nearest Neighbor</i>	.901	.502	.645	.802
BA-10-PCA	<i>Neural Network</i>	.863	.851	.857	.910
BA-90-RFE	<i>Discriminant Analysis</i>	.960	.910	.934	.959

Este estudio demuestra que la aplicación de abono en campos puede ser detectada mediante satélites multispectrales con gran éxito. El mejor modelo tiene una puntuación media de  $F_1$  de 93,4%. Sin embargo, estos resultados se obtuvieron utilizando un conjunto de datos de 31,48 ha de campos recién abonados. Con un conjunto de datos más grande y variado, las conclusiones podrían cambiar ligeramente.

Se descubrió que combinar las características de las imágenes antes y después de la aplicación de abono proporcionaba mejores resultados. Por ejemplo, la media de *Recall* de BA-128 es aproximadamente un 8% mejor que la de A-64 mientras mantiene su media de *Precision*. Esta mejora es consistente para cada experimento del tipo “BA” versus “A”.

Los modelos con un número reducido de características tienen resultados inferiores. El método de selección de características de agrupamiento de *Ward* (A-7-W y BA-14-W) muestra resultados aproximadamente un 5-7% menores que cuando se utilizan todas las características.

El método de eliminación recursiva de características (BA-90-RFE) logra mejores resultados que el uso de todas las características (BA-128). El uso de más características de las necesarias puede hacer que los modelos sean más difíciles de entrenar y producir detecciones menos precisas. Este problema se conoce como la “maldición de la dimensionalidad”. La mayoría de las características están altamente correlacionadas, lo que hace innecesario el uso de todas ellas.

Se logran resultados aceptables con BA-RGB, utilizando solo las bandas RGB de ambas imágenes (después y antes de la aplicación de abono). Esto puede deberse a la resolución espacial ofrecida por estas bandas de  $10\text{ m} \times 10\text{ m}$  por píxel. Sin embargo, BA-16-IR todavía lo supera en aproximadamente un 4%. Incluso cuando las bandas IR (bandas con una longitud de onda mayor que la banda Roja B04) tienen una resolución espacial de  $20\text{ m} \times 20\text{ m}$  por píxel, todavía se logran mejores resultados. Esto es apoyado por la respuesta de las bandas IR cuando se aplica abono, como se muestra en la Figura 3.13. La figura ilustra que las bandas B06, B07, B08, B8A y B09 están altamente correlacionadas, mientras que B10 y B11 están inversamente correlacionadas. Esto respalda los hallazgos del estudio y muestra que los datos en bruto en longitudes de onda IR son esenciales para detectar la aplicación de abono. Si las bandas IR tuvieran una resolución espacial similar a las bandas RGB, se espera que los resultados sean aún más precisos. Del mismo modo, la longitud de onda de las bandas IR va desde 700 nm hasta 2200 nm, por lo que es posible que si este rango se ampliara, los resultados mejorasen aun más.

*Discriminant Analysis* es el método de clasificación que ofrece los resultados más precisos, especialmente cuando se utiliza un alto número de características. *Support Vector Machines* suele obtener mejores resultados cuando el número de características es menor.

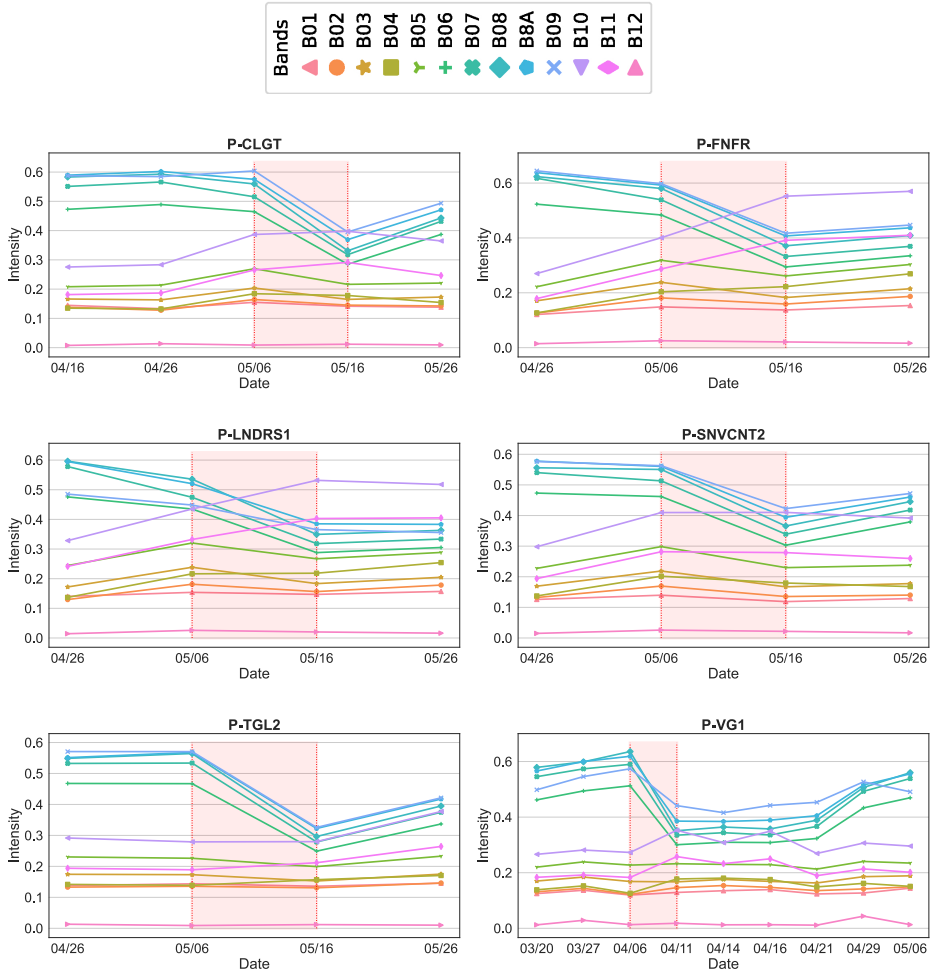


Figura 3.13: Valores de intensidad de las bandas de Sentinel-2 para la serie temporal de las parcelas P-CLGT, P-FNFR, P-LNDRS1, P-SNVCNT2, P-TGL2, y P-VG1.

El mejor experimento es BA-102-VI. La Tabla 3.15 muestra los resultados para cada método de clasificación sobre este conjunto de características. En este caso, el método más preciso es *Discriminant Analysis*, que supera al segundo método más preciso, los clasificadores *Ensemble*, por casi un 2% de  $F_1$ -Score. La Tabla 3.16 muestra los detalles del método más preciso. En esta tabla, se muestran la *Precision*, *Recall* y  $F_1$ -Score para la clase Otros y la clase Aplicación de estiércol. Las clases están ligeramente desequilibradas porque hay más muestras para la clase Otros. Por esta razón, es más razonable centrarse en los resultados de la clase Aplicación de estiércol, específicamente el  $F_1$ -Score, que considera tanto la *Precision* como el *Recall*. El  $F_1$ -Score para la clase Aplicación de estiércol es del 89,1%, lo que indica que este modelo es capaz de detectar con éxito campos fertilizados a nivel de píxel.

Tabla 3.15: Resultados por método de clasificación para el experimento BA-102-VI. mP=*Precision* media, mR=*Recall* media, mF<sub>1</sub>=F<sub>1</sub>-Score medio.

Método de clasificación	mP	mR	mF <sub>1</sub>	OA
Decision Tree	0.833	0.769	0.800	0.881
<b><i>Discriminant Analysis</i></b>	0.952	<b>0.916</b>	<b>0.934</b>	<b>0.959</b>
<i>Logistic Regression Classifiers</i>	0.838	0.911	0.873	0.904
<i>Naïve Bayes Classifiers</i>	0.779	0.879	0.826	0.851
<i>Support Vector Machines</i>	<b>0.955</b>	0.836	0.891	0.933
<i>Nearest Neighbor Classifiers</i>	0.805	0.771	0.787	0.871
<i>Kernels Approximation Classifiers</i>	0.814	0.671	0.736	0.852
<i>Ensemble Classifiers</i>	0.953	0.884	0.917	0.949
<i>Neural Network</i>	0.923	0.906	0.914	0.946

Tabla 3.16: Resultados por clase para el método de clasificación de Análisis Discriminante del experimento BA-102-VI.

Class	<i>Precision</i>	<i>Recall</i>	$F_1$ -Score
Otros	0.962	0.987	0.974
Aplicación de abono	0.942	0.845	0.891

La Figura 3.14 muestra la representación visual de la detección de estiércol en el conjunto de pruebas. Algunas parcelas aparecen en la misma imagen y tienen la misma fecha de aplicación de estiércol. Este es el caso de P-MT y P-LLT, y de P-LNDRS1, P-LNDRS2, P-LNDRS3 y P-LNDRS4.

Las parcelas recién abonadas son detectadas casi perfectamente con pocos o ningún píxel clasificado incorrectamente. Solo una pequeña región se clasifica incorrectamente. Esto ocurre en las tres imágenes. Sin embargo, algunas regiones fuera de la máscara objetivo son desconocidas y no se pueden utilizar para juzgar la calidad de la detección. Esto es solo una visualización de las imágenes completas de las parcelas del conjunto de pruebas. Estas regiones desconocidas no afectan a las métricas numéricas obtenidas anteriormente ya que los píxeles desconocidos no se utilizan en el cálculo de las métricas. No obstante, los resultados visuales parecen ser correctos.

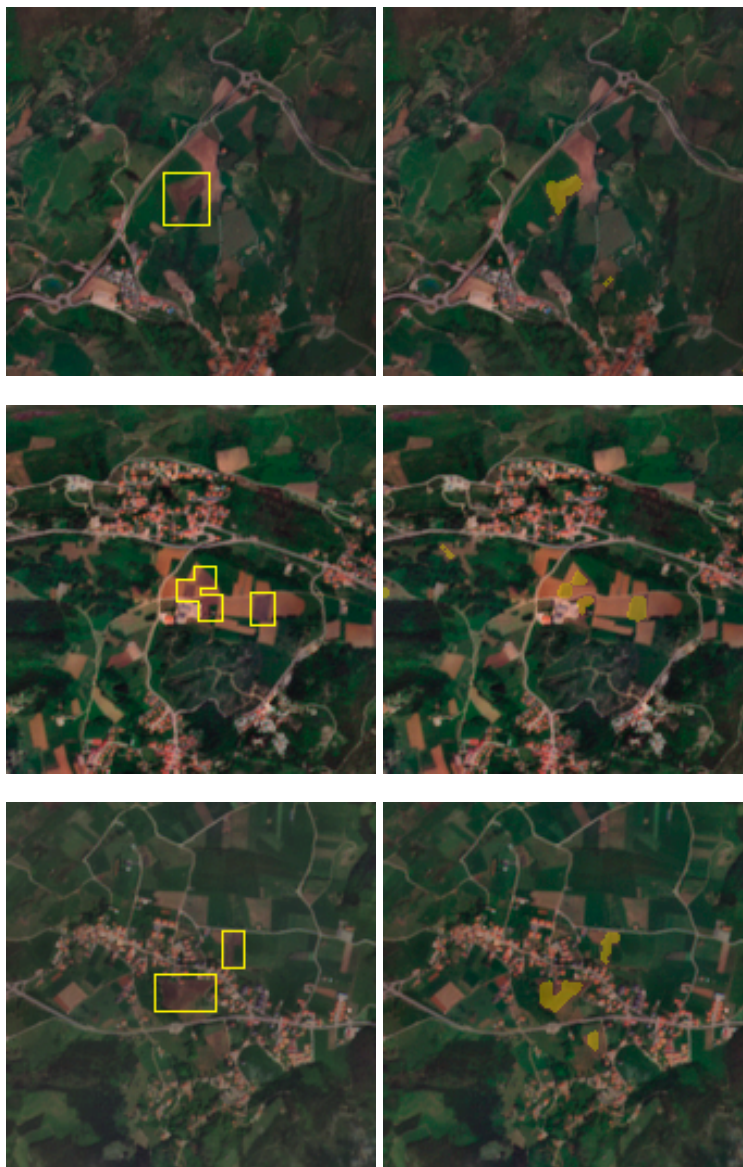


Figura 3.14: Detecciones del conjunto de pruebas. La columna izquierda es la ubicación de la verdad terrestre y la columna derecha es la máscara de detección. P-CLGT en la primera fila, P-LNDRS1/2/3/4 en la segunda fila y P-LLT y P-MT en la tercera fila.

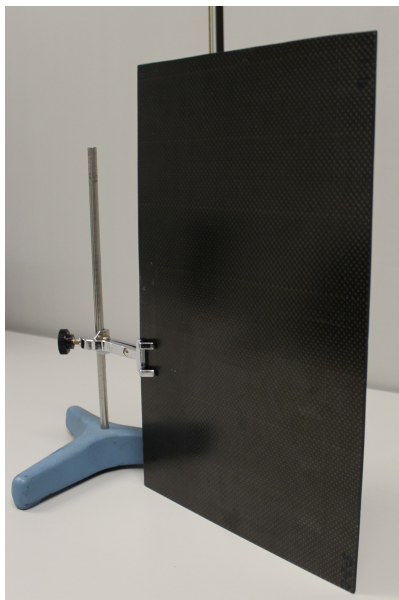
## 3.2. Defectos subsuperficiales

Para evaluar el rendimiento de la detección de defectos subsuperficiales mediante la termografía infrarroja activa se evalúan y comparan las redes de segmentación semántica UNet y DeepLabV3+ contra *Random Forest* y *Support Vector Machines*. Para ello se crean varios conjuntos de datos en los que se recopila la información de grabaciones digitales de calentamiento y enfriamiento de una pieza mediante una imagen multicanal.

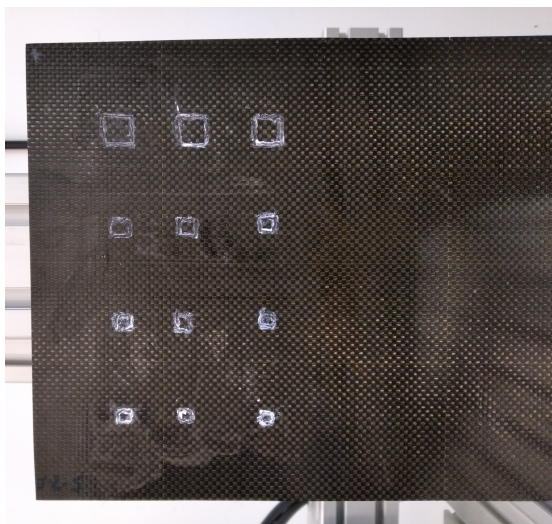
Esta investigación se centra en el polímero reforzado con fibra de carbono (CFRP). El CFRP es un material compuesto por fibra de carbono reforzada y una resina para unir los refuerzos. Se utilizará una pieza para el entrenamiento y prueba, y otras dos para la validación de los modelos. Cada pieza contiene 12 defectos artificialmente inducidos. Hay dos tipos diferentes de defectos: películas delgadas de politetrafluoroetileno (PTFE) y defectos de virutas de acero. Las películas de PTFE simulan delaminaciones, que son defectos comunes en materiales compuestos producidos por la separación de capas adyacentes, mientras que las inserciones de acero simulan la inclusión accidental de pequeñas piezas de herramientas de corte utilizadas durante el proceso de fabricación del material. Los defectos tienen diferentes tamaños y están introducidos en diferentes profundidades. Estas variaciones sirven para estudiar como se comportan los modelos en diferentes condiciones. De esta forma las conclusiones sacadas son más generalizables a casos reales. Una mayor superficie del defecto implica un mayor flujo de calor afectado; y como consecuencia, implica una mayor variación de temperaturas en las áreas cercanas al defecto. Por otro lado, cuanto más superficial, menor será el efecto de disipación de calor lateral. Como consecuencia de la presencia de un defecto, el flujo de calor hacia la superficie será menos degradado, lo que hace que el efecto térmico en la superficie sea más evidente.

La Figura 3.15 muestra: una fotografía general del espécimen que se utilizará para el entrenamiento y evaluación (Figura 3.15a), y una fotografía que muestra la ubicación de los defectos (Figura 3.15b). El espécimen de 360 mm × 240 mm tiene un grosor de 2,5 mm, con una estructura de 12 capas.

De los 12 defectos, 9 son películas delgadas de PTFE y 3 son defectos de virutas de acero. Hay tres tamaños diferentes de defectos PTFE (12 mm × 12 mm, 7 mm × 7 mm y 5 mm × 5 mm), cada uno a 3 profundidades diferentes (0,63 mm, 1,46 mm y 2,08 mm), y solo un tamaño de viruta de acero (5 mm × 5 mm) ubicado a 0,63 mm, 1,46 mm y 2,08 mm. Los tres defectos inferiores son defectos de virutas de acero y el resto son defectos de PTFE. El grosor de los defectos se midió con un calibre obteniendo un valor de 0,06 mm. En la Figura 3.16, se muestra la ubicación de los defectos en el espécimen. Las profundidades y capas de los defectos se presentan de menos profundo a más profundo, de izquierda a derecha: la primera columna de defectos tiene una profundidad de 0,63 mm, la segunda columna 1,46 mm y la tercera columna 2,08 mm.



(a) Imagen



(b) Defectos

Figura 3.15: Fotografías del espécimen 1.



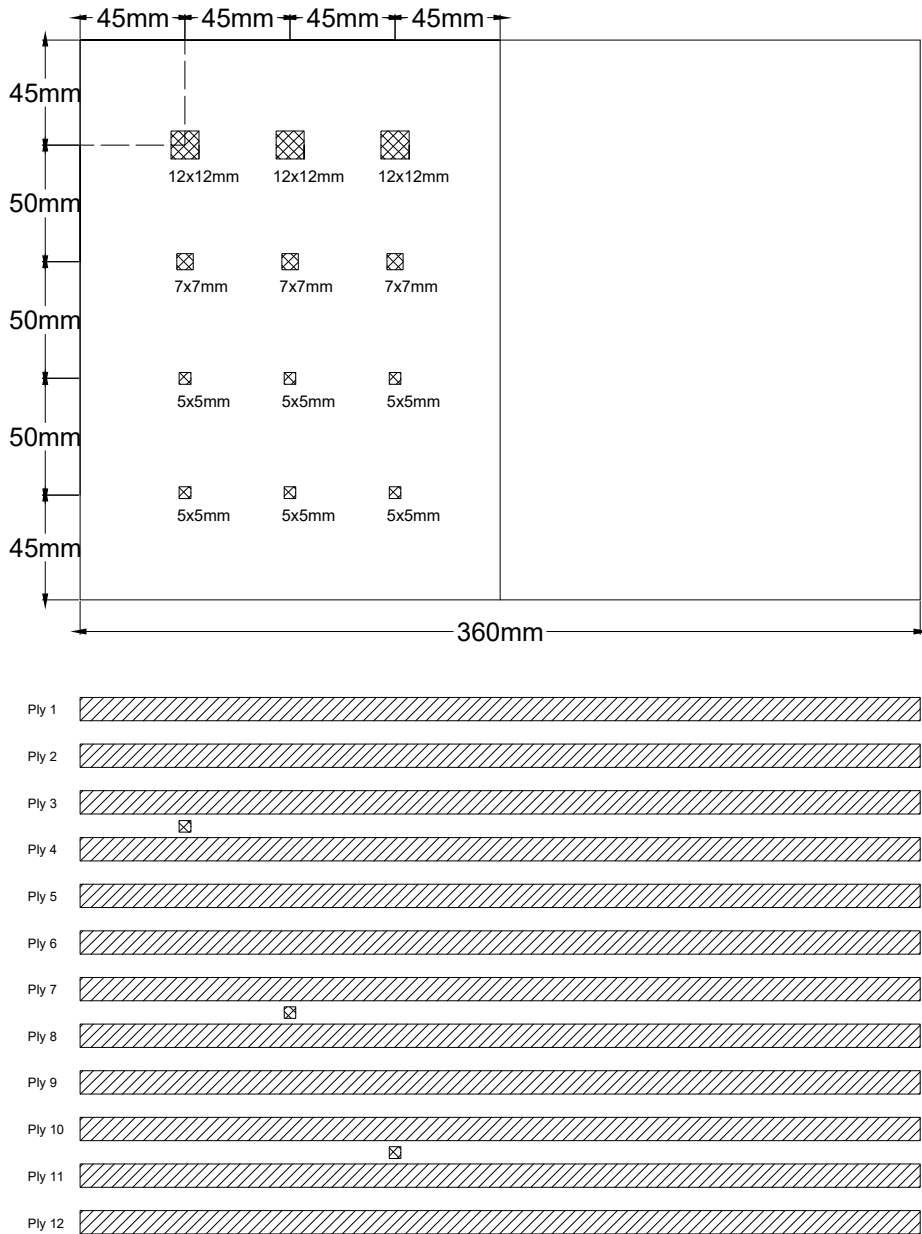


Figura 3.16: Dimensiones y localización de los defectos del espécimen 1.

### 3.2.1. Generación del conjunto de datos

#### 3.2.1.1. Termografía infrarroja mediante calentamiento gradual

El laminado CFRP se calienta durante diez segundos usando dos lámparas halógenas (modelo Eurolite PAR-64 Profi Floorspot de 1.000 W). Después de los diez segundos, se apagan las lámparas para permitir que el espécimen se enfríe durante otros diez segundos. Este proceso se graba usando un detector de IR NETD de menos de 55 mK y ópticas de lentes F/1 de 25 mm durante un total de veinte segundos a 50 fotogramas por segundo, con una resolución de  $640 \times 480$  píxeles. Esto genera un total de 1.000 fotogramas para cada grabación digital. La cámara utilizada para grabar las imágenes digitales es un modelo Xenics Gobi 640 GigE con un rango espectral entre 8 y 14 micras y una resolución de  $480 \times 640$  píxeles. En la Figura 3.17 se muestra un diagrama de la configuración para las grabaciones con la ubicación, distancia y ángulo de la cámara infrarroja, las lámparas halógenas y la muestra.

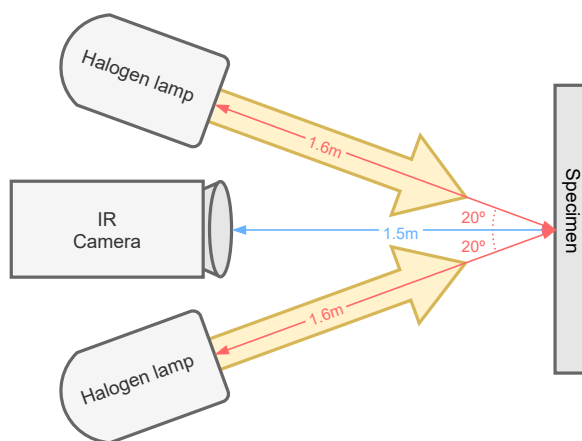


Figura 3.17: Configuración para la grabación.

El tiempo de calentamiento necesario para revelar la presencia de defectos se definió aproximadamente mediante una simulación numérica en una etapa preliminar y posteriormente se verificó el tiempo definitivo mediante una evaluación experimental. Este intervalo de tiempo produjo el máximo número de defectos que se podían detectar, evitando que la muestra se sobrecalentara.

Los defectos subsuperficiales se calientan y se enfrían a diferentes velocidades que el resto del objeto. La estimulación activa se aplica para aprovechar esta característica como una forma de obtener el máximo contraste posible entre los defectos y el resto del objeto. Para cada grabación digital, el laminado CFRP se enfría a temperatura ambiente antes de que comience el proceso, para evitar que el objeto se caliente a diferentes temperaturas.

La Figura 3.18 muestra este proceso de calentamiento y enfriamiento para un píxel con defecto y otro píxel cercano sin defecto (ver Figura 3.19). Además, se añaden estos mismos puntos de referencia pero con la muestra girada  $120^\circ$  (ver Figura 3.20). No se necesitan valores absolutos, solo se requieren las diferencias entre píxeles cercanos para localizar los defectos. Al girar la muestra se puede observar que la respuesta no es la misma. Esto se debe a que la energía térmica no se transmite uniformemente por toda la muestra, lo que es de gran interés para crear un conjunto de datos que permita que la red se generalice.

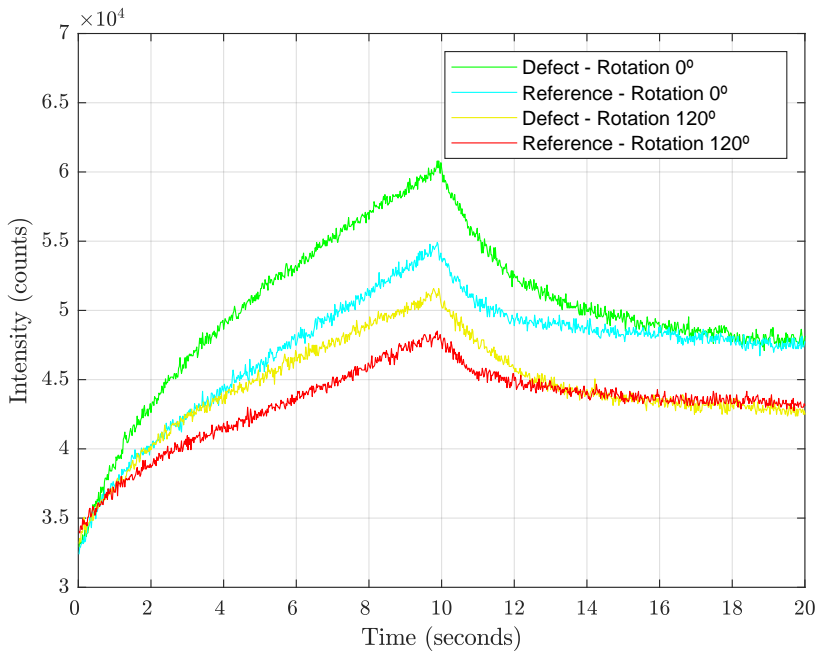


Figura 3.18: Señales de intensidad de calentamiento y enfriamiento para la secuencia temporal.

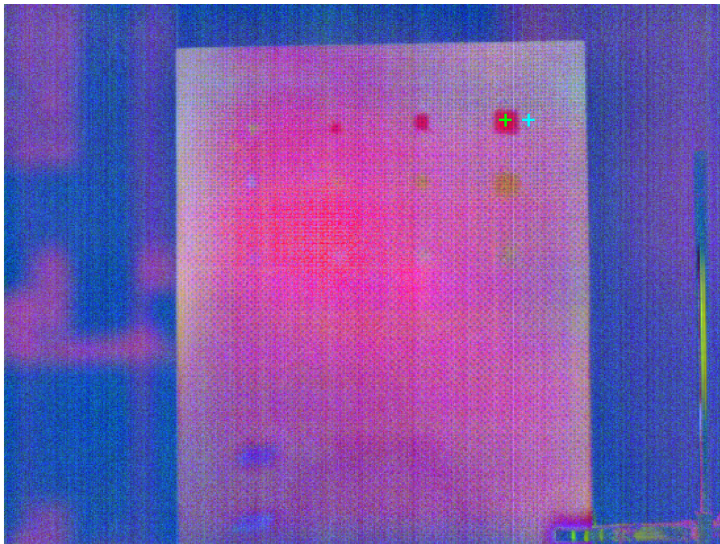


Figura 3.19: Píxeles de referencia con la muestra rotada  $0^\circ$ .

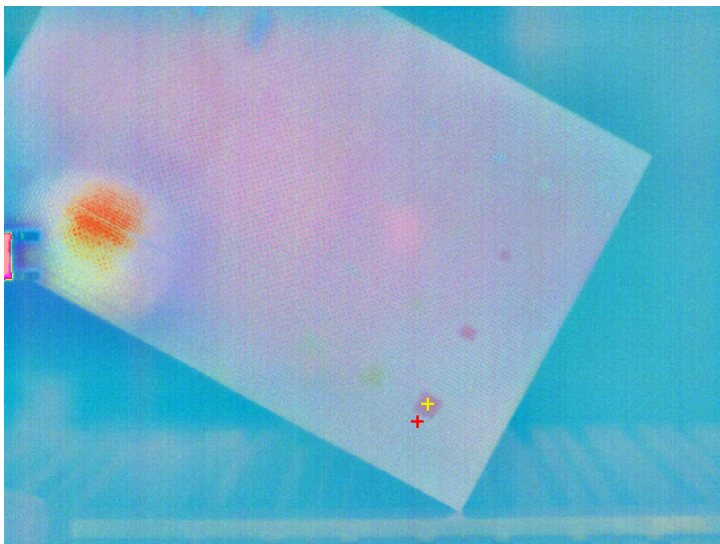


Figura 3.20: Píxeles de referencia con la muestra rotada  $120^\circ$ .

### 3.2.1.2. Preprocesado de los datos

Se utilizan métodos de postprocesamiento de imágenes para resumir la información de un vídeo termográfico a una imagen multicanal particular. Esta compresión de datos es necesaria para poder utilizar UNet y DeepLabV3+ debido a su coste computacional. Para estudiar el efecto de esta compresión, se evalúan dos enfoques diferentes.

El primer enfoque convierte el proceso de calentamiento de la grabación digital en una imagen con 3 canales diferentes utilizando el método *Principal Component Thermography* (PCT) [101]. Este enfoque se prueba con los métodos *Random Forest*, *Support Vector Machines*, UNet y DeepLabV3+.

El segundo enfoque aprovecha tanto las secuencias de calentamiento como de enfriamiento y utiliza 15 canales para cada una, lo que resulta en imágenes de 30 canales. Cada canal almacena imágenes postprocesadas generadas por los siguientes métodos: PCT [79], *Pulsed Phase Thermography* (PPT) [13, 90], *Kurtosis* [75], *Skewness* [74] y *Thermographic Signal Reconstruction* (TSR) [9, 113]. Este enfoque se prueba con UNet.

- ***Principal Component Thermography*** se aplica a cada historial temporal de píxeles, calculando una transformación lineal a los datos iniciales a partir de los autovectores de la matriz de covarianza asociada. Con este método, la distinción entre defecto y no defecto es más fácilmente visible. En este estudio, para las imágenes de 3 canales, cada imagen postprocesada corresponde a los componentes 1, 3 y 4 de la PCT de la secuencia de calentamiento (500 fotogramas). El segundo componente no se utiliza para las imágenes de 3 canales, ya que la relación señal-ruido es mayor en los componentes 3 y 4 [125]. Para las imágenes de 30 canales, se utilizan los cuatro primeros canales de la PCT tanto en las secuencias de calentamiento (500 fotogramas) como en las de enfriamiento (500 fotogramas), lo que da un total de ocho canales.
- ***Pulsed Phase Thermography*** es un método para calcular la fase de los datos termográficos por historial temporal de píxeles basado en el algoritmo de Transformada Discreta de Fourier (*Discrete Fourier Transform*, DFT) [16]. El algoritmo DFT se utiliza generalmente en el postprocesamiento de imágenes para filtrar el ruido periódico. Se puede utilizar para obtener una imagen que solo representa los bordes de los elementos en la imagen. Para las imágenes de 30 canales, se utiliza la fase de la frecuencia mínima, obteniendo una imagen postprocesada para el proceso de calentamiento y otra para el proceso de enfriamiento.

- **Kurtosis** mide el grado de pico de una distribución. Si la distribución es la misma que la distribución normal, tiene un valor de cero, si es más alta tiene un valor positivo y si es más baja un valor negativo. En este caso, esta medida se calcula por historial temporal de píxeles utilizando las secuencias de calentamiento y enfriamiento, obteniendo dos canales para las imágenes de 30 canales.
- **Skewness** mide la falta de simetría. Una asimetría positiva significa que la cola más larga de la distribución está a la derecha del histograma y al revés para la asimetría negativa. Una distribución completamente simétrica tiene un valor de cero. La asimetría se calcula por píxel utilizando cada fotograma en el proceso de calentamiento o enfriamiento. Para las imágenes de 30 canales, esto resulta en dos canales, uno para el proceso de calentamiento y otro para el proceso de enfriamiento.
- **Thermographic Signal Reconstruction** se calcula utilizando expresiones logarítmicas, es un método para estimar la difusividad térmica eliminando el ruido de una señal térmica basada en una secuencia. Este método se calcula por historia temporal de píxel y se utiliza comúnmente para la detección de defectos. Se considera que un grado de 7 generalmente proporciona resultados óptimos para la detección de defectos en laminados [10]. Esto genera una imagen posprocesada para los coeficientes de cada grado más su coeficiente cero. Teniendo esto en cuenta, para las imágenes de 30 canales se crean ocho canales para el proceso de calentamiento y otros ocho canales para el proceso de enfriamiento.

### 3.2.1.3. Conjuntos de datos

Usando el método descrito anteriormente, se generan 36 grabaciones digitales. Todas las grabaciones digitales se enfocan en el mismo espécimen en diferentes rotaciones, lo que altera la iluminación, los reflejos de las lámparas y el fondo, entre otras cosas. Este proceso se realiza para obtener más datos para el entrenamiento y mejorar la variabilidad. Para cada grabación digital, el laminado CFRP se gira 10°. A partir de cada grabación digital, se generan dos imágenes (una con 3 canales y la otra con 30 canales) utilizando los métodos de postprocesamiento de imágenes. En la Figura 3.21 se muestra un ejemplo de cada canal de una imagen de 30 canales. El objetivo de este estudio no es la visualización, sino la detección y localización. La Figura 3.21 simplemente proporciona una visualización de las entradas normalizadas.

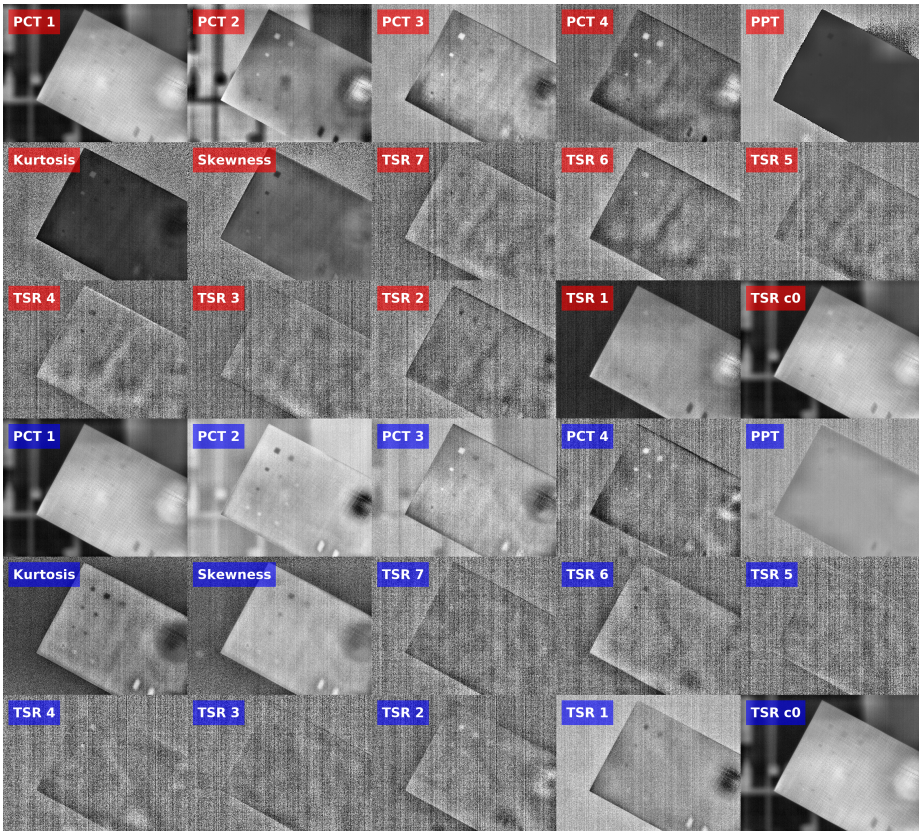


Figura 3.21: Ejemplo de una imagen de 30 canales. Las imágenes con etiqueta roja se obtienen de la secuencia de calentamiento. Las imágenes con etiqueta azul se obtienen de la secuencia de enfriamiento.

Los defectos de estos especímenes son generados artificialmente, sin embargo, pueden estar ligeramente desplazados del esquema original. Por esta razón, se realiza una inspección ultrasónica para encontrar y comprobar las posiciones reales de los defectos. La máscara objetivo se genera mediante un procedimiento manual. Primero, se pasa una sonda sobre la superficie del espécimen, escaneando la señal que recibe de manera similar a un osciloscopio. De esta forma, es posible detectar cambios de señal que son indicativos de un defecto. Esta área defectuosa se marca con un lápiz en el propio espécimen. Finalmente, utilizando la imagen termográfica y la imagen RGB en la que se pueden observar las marcas de lápiz, se genera manualmente una máscara de verdad de campo observando y superponiendo ambas imágenes.

Se crean dos conjuntos de datos, uno que utiliza imágenes con 3 canales y otro que utiliza imágenes con 30 canales. De esta manera, se puede hacer una comparación para determinar si más información resulta en una mayor precisión. Cada conjunto de datos consta de 36 imágenes. Las primeras 30 se utilizan para el entrenamiento y las últimas 6 para pruebas y visualización. Ambos conjuntos de datos se pueden encontrar en el siguiente DOI<sup>2</sup>.

Las clases se dividen en “Defecto” y “Otro”. El objetivo es la clasificación binaria, por lo que la clase Defecto es la clase objetivo, y la clase Otro es la clase no objetivo que se refiere a todo lo demás, incluyendo el resto del espécimen y el fondo de la grabación digital.

## 3.2.2. Resultados y discusiones

### 3.2.2.1. *Random Forest* y *Support Vector Machines*

Se realizan experimentos con *Random Forest* y *Support Vector Machines* como base para hacer comparaciones. Ambos usan el mismo vector de características, que se calcula usando trece características. Las primeras tres características consisten en los valores rojo, verde y azul (RGB) de cada píxel de la imagen de entrada, que corresponden a el primer, tercer y cuarto componente de PCT.

La cuarta característica es el patrón binario local (*local binary pattern*, LBP), un descriptor de textura utilizado en la visión por computador, que se calcula aplicando un umbral a la vecindad de cada píxel en una ventana de  $3 \times 3$  y convirtiendo el resultado en un número decimal [59]. Para calcular los componentes de la vecindad, se utiliza un radio de 24, lo que suma un total de 192 vecinos. LBP se aplica a una versión en escala de grises de las imágenes de 3 canales para obtener más contexto espacial. LBP se calcula con la Ecuación 3.2, donde P es el número total de vecinos, R es el radio, c es el píxel central y g es el valor de un píxel.

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)^{2^p} \text{ with } s(x) = \begin{cases} 1, & \text{if } x \geq 0; \\ 0, & \text{otherwise.} \end{cases} \quad (3.2)$$

Las últimas nueve características consisten en múltiples características de textura Haralick, que son descriptores de textura utilizados en la visión por computador para la clasificación de imágenes. Todas las características de Haralick se basan en la matriz de co-ocurrencia de niveles de gris que muestra la frecuencia con la que ocurre cada nivel de gris en un píxel en una ubicación geométrica fija con respecto a otros píxeles. Las características utilizadas son: momento angular

---

<sup>2</sup>[10.5281/zenodo.5426792](https://doi.org/10.5281/zenodo.5426792)



de segundo orden, contraste, correlación, suma de cuadrados: varianza, momento de diferencia inversa, promedio de suma, entropía de suma y entropía. Las ecuaciones para todas las características de textura de Haralick están en disponibles en [81]. Las características de Haralick se aplican a la versión en escala de grises de las imágenes de 3 canales para obtener más contexto espacial.

Se realiza un submuestreo al vector de características de cada imagen para reducir el tiempo de entrenamiento y el uso de memoria. Esto genera 1.000 observaciones por imagen con 13 características por observación. Con 30 imágenes para entrenar, se utilizan 30.000 observaciones para el entrenamiento.

Para *Random Forest*, se optimiza manualmente el número de estimadores y su profundidad máxima. Además, se prueban diferentes pesos de clase tanto para *Random Forest* como para *Support Vector Machines*. El mejor experimento de *Random Forest* utiliza 1.000 estimadores y una profundidad máxima de 10. En el caso de *Support Vector Machines*, se utiliza un kernel de función de base radial y el valor de gamma se calcula como la inversa de la multiplicación del número de características por la varianza. En ambos casos, el peso de la clase para la clase no objetivo y para la clase objetivo o defecto se equilibra utilizando la inversa proporcional de las frecuencias de las clases. Los resultados de estos experimentos se pueden ver en la Tabla 3.17.

Según la Tabla 3.17, ambos experimentos obtienen métricas bajas: por debajo del 12% en  $F_1$ -Score. Para demostrar que estos valores son demasiado bajos, se muestran visualizaciones de *Random Forest* y *Support Vector Machines* en la Figura 3.22 y 3.23, respectivamente.

En SVM, los bordes del espécimen son clasificados como defectos, esto se debe a la gran variación entre el espécimen y el fondo. En RF, aunque esto se puede observar en algunos casos, es mucho menos obvio. Además, ambos modelos predicen muchos más píxeles de los defectos más superficiales ya que son los que tienen más variación.

Hay un área circular detectada en la parte inferior tanto del modelo de RF como de SVM. Esta área es el reflejo de las lámparas de calentamiento. Al observar la Figura 3.23, es evidente que SVM es muy sensible a estos artefactos, más que RF. Estos reflejos se pueden evitar colocando la cámara adecuadamente, aunque esto no siempre es posible en inspecciones reales debido a la falta de espacio. Es muy común encontrar reflejos en inspecciones reales. Por lo tanto, parece razonable incluirlos en el estudio y analizar la robustez de los modelos en su presencia.

Teóricamente, se podrían lograr mejores resultados mejorando el vector de características. La selección de características tiene el impacto más significativo en el rendimiento de estos métodos. Sin embargo, en este estudio, se usan características comunes para la segmentación de imágenes [59, 81].

Tabla 3.17: Métricas para los experimentos con *Random Forest* y *Support Vector Machines*.

Método	<i>Precision</i>	<i>Recall</i>	<i>IoU</i>	$F_1$
RF	.103	.132	.061	.116
SVM	.037	.604	.036	.069

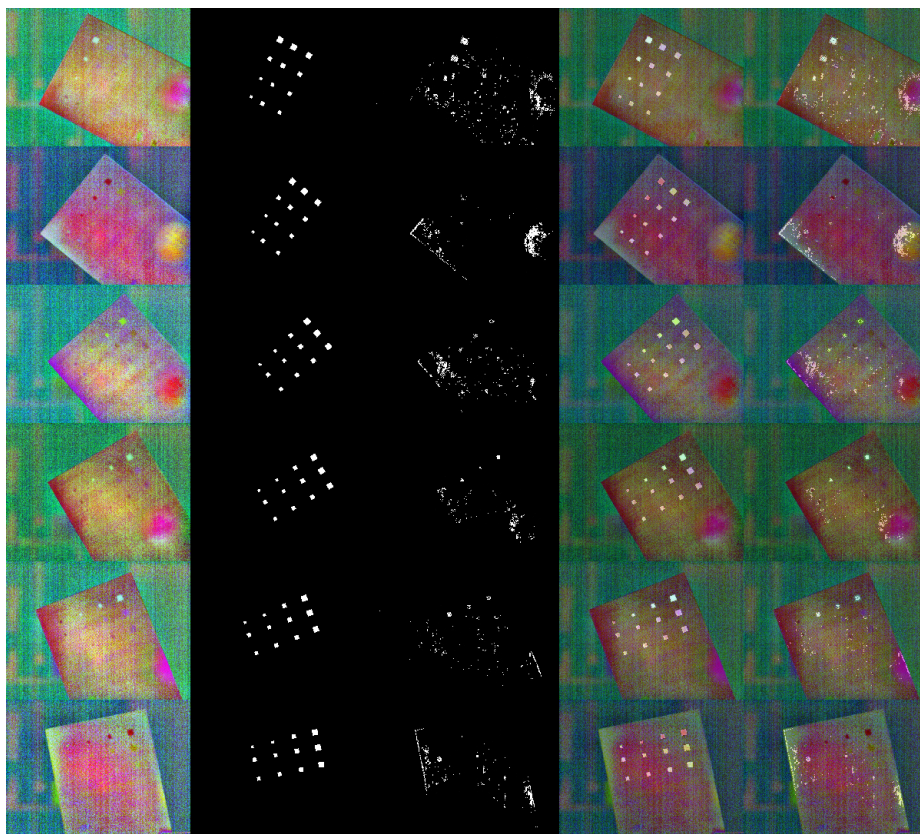


Figura 3.22: La visualización de los resultados predichos por *Random Forest*. (1ra col.) Imágenes originales, (2da col.) máscaras de la verdad terrena, (3ra col.) predicciones, (4ta col.) imágenes originales y máscaras de la verdad terrena, (5ta col.) imágenes originales con las predicciones.

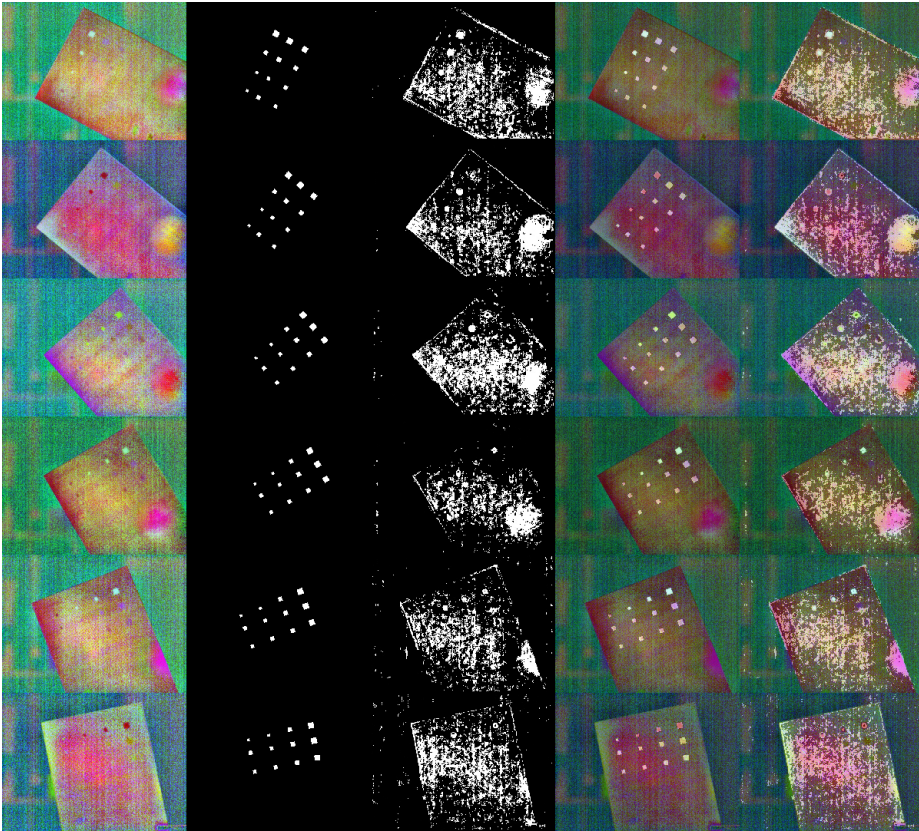


Figura 3.23: Visualización de los resultados predichos por *Support Vector Machines*. (**1ra col.**) Imágenes originales, (**2da col.**) máscaras de la verdad terrena, (**3ra col.**) predicciones, (**4ta col.**) imágenes originales con máscaras de la verdad terrena, (**5ta col.**) imágenes originales con predicciones.

### 3.2.2.2. UNet

Esta sección presenta los resultados de segmentación con UNet para imágenes de 3 canales e imágenes de 30 canales. Ambos experimentos tienen los mismos hiperparámetros óptimos (Tabla 3.18 y 3.19) con la excepción del *batch size* y los pesos de clase. Dado que las imágenes de 3 canales ocupan menos memoria que las imágenes de 30 canales, el *batch size* se puede aumentar de cuatro a ocho imágenes. En cuanto a los pesos de clase, los valores óptimos difieren entre los conjuntos de datos, desde un valor de 0,65 a 0,80.

Tabla 3.18: Parámetros de UNet.

Parámetros de la red		
Parámetros	3-canales	30-canales
Entrada	640×480×3	640×480×30
Clases	2	2
Profundidad	4	4
Filtros en el primer nivel	32	32
Relleno o <i>Padding</i>	Sí	Sí

Tabla 3.19: Parámetros de entrenamiento para UNet.

Parámetros de entrenamiento		
Parámetros	3-canales	30-canales
<i>Solver</i>	Adam	Adam
<i>Epochs</i>	1000	1000
<i>Batch size</i>	8	4
<i>Learning rate</i>	0.001	0.001
Balanceo de clases	0.35 - 0.65	0.20 - 0.80
<i>Gradient clipping</i>	No	No
Regularización L2	0.0001	0.0001
Aumento de datos	Espejo X/Y	Espejo X/Y
Shuffle	Sí	Sí

La profundidad de la arquitectura de UNet coincide con la implementación original, pero se ha reducido el número de filtros en el primer nivel en dos. Esto afecta a toda la arquitectura dividiendo los números de filtros por dos. El uso de menos filtros significa tiempos de entrenamiento más rápidos y tamaños de *batch size* mayores. No hay necesidad de limitar el gradiente ya que no se presenta el

problema del gradiente explosivo. La regularización L2 funciona mejor cuando se utiliza el valor predeterminado de 0,0001. Todos los datos de entrenamiento se reordenan antes de cada *epoch* para evitar el sobreajuste del modelo.

Las métricas de las pruebas de los experimentos “imágenes de 30 canales” e “imágenes de 3 canales” se pueden ver en la Tabla 3.20.

Tabla 3.20: Métricas para los experimentos con UNet.

<b>Experimento</b>	<b><i>Precision</i></b>	<b><i>Recall</i></b>	<b><i>IoU</i></b>	<b><math>F_1</math></b>
3-canales	.689	.717	.542	.703
30-canales	.764	.726	.593	.745

La Tabla 3.20 muestra una gran diferencia entre el uso de 3 y 30 canales. En este caso, el experimento con imágenes de 30 canales tiene un  $F_1$ -Score casi un 5% más alto. Ambos experimentos superan el 70% en  $F_1$ -Score y obtienen un equilibrio entre *Precision* y *Recall*.

El experimento con imágenes de 3 canales requiere 00h:31m:31s en entrenar con una tarjeta de procesamiento NVIDIA GeForce RTX 2080 Ti, mientras que el experimento con imágenes de 30 canales requiere 02h:15m:15s. Los canales adicionales hacen que la arquitectura sea más compleja.

Para acompañar estos resultados, se puede observar una visualización de las imágenes de prueba en las Figuras 12 y 13. En estas figuras se puede ver una gran diferencia entre los modelos. El experimento con imágenes de 3 canales (Figura 3.24) detecta todos los defectos, aunque los que tienen mayor profundidad tienen un área mucho más pequeña que las máscaras objetivo y hay algunos falsos positivos. La Figura 3.25 tiene mucho menos ruido pero tiene problemas para detectar todos los defectos en algunas de las imágenes.

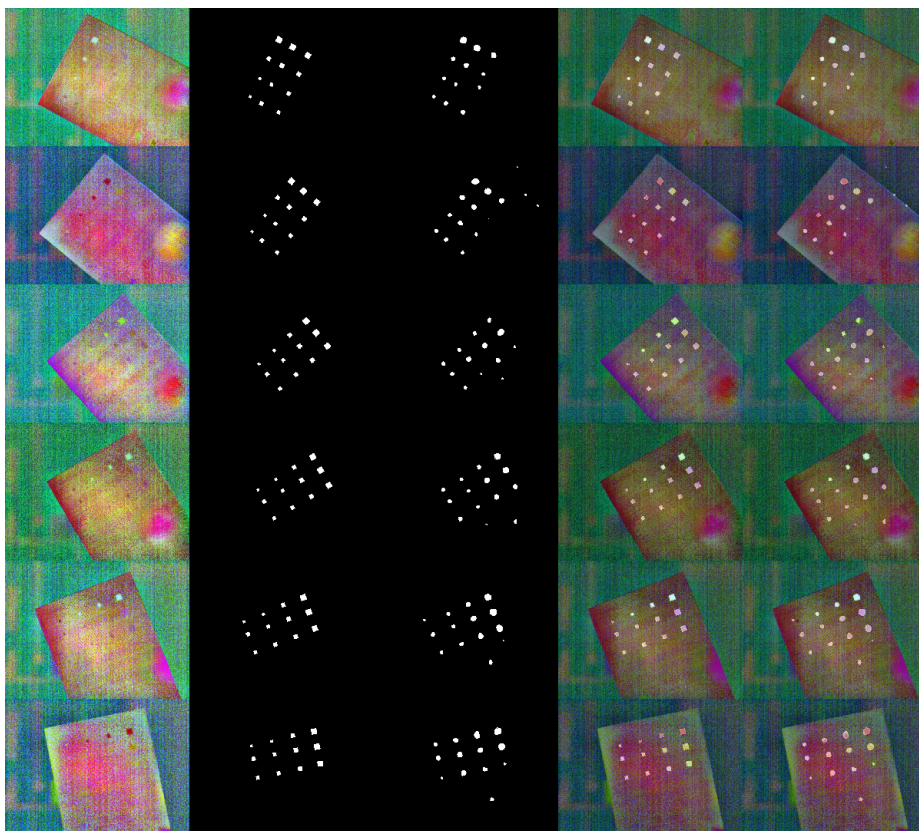


Figura 3.24: Visualización de los resultados predichos para UNet evaluado con 3 canales. **(1ra col.)** Imágenes originales, **(2da col.)** máscaras de verdad terreno, **(3ra col.)** predicciones, **(4ta col.)** imágenes originales con máscaras de verdad terreno, **(5ta col.)** imágenes originales con predicciones.

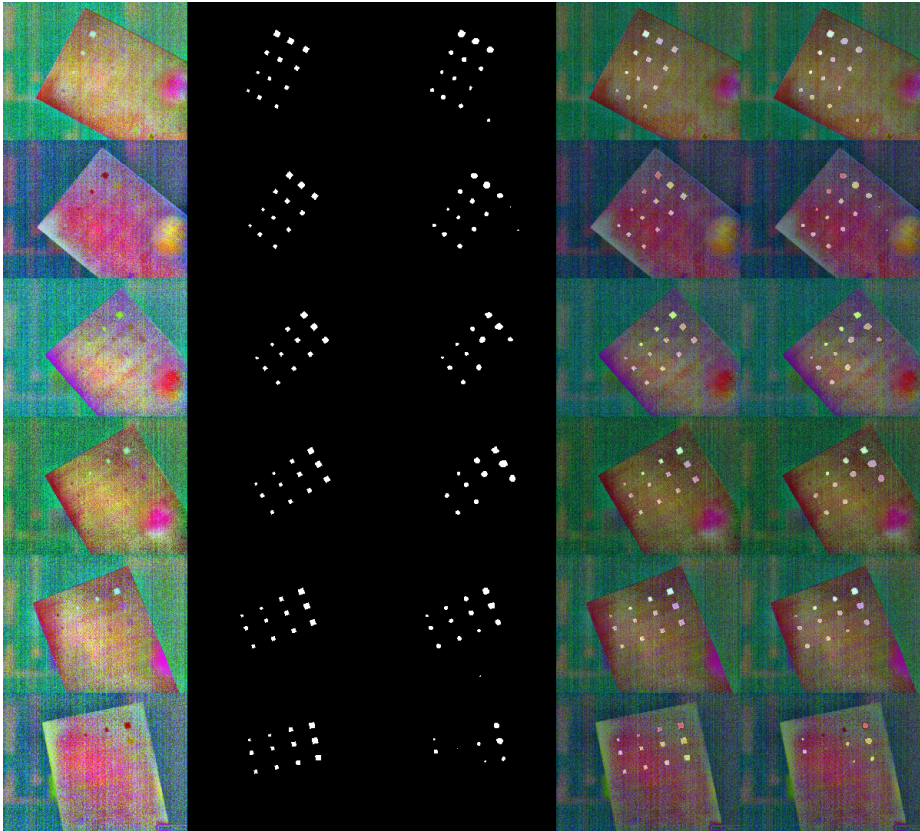


Figura 3.25: Visualización de los resultados predichos para UNet evaluado con 30 canales. Solo se muestran los primeros tres canales en la imagen, que consisten en los componentes primero, tercero y cuarto, utilizando exactamente los mismos canales que las imágenes de 3 canales. (**1<sup>a</sup> col.**) Imágenes originales, (**2<sup>a</sup> col.**) máscaras de verdad terreno, (**3<sup>a</sup> col.**) predicciones, (**4<sup>a</sup> col.**) imágenes originales con máscaras de verdad terreno, (**5<sup>a</sup> col.**) imágenes originales con predicciones.

### 3.2.2.3. DeepLabV3+

Esta sección presenta el mejor experimento con DeepLabV3+ usando imágenes de 3 canales. La Tabla 3.21 muestra los parámetros de la arquitectura de red. En este caso, la arquitectura Xception65 es la que funciona mejor como backbone. Xception71 tiene más capas y, por lo tanto, se necesita más VRAM para el mismo tamaño de *batch size*. Un *batch size* más pequeño, incluso cuando se utiliza una red con más capas, funciona peor. Por esta misma razón, se prefiere un *output stride* de 16.

La Tabla 3.22 muestra los parámetros de entrenamiento. En este caso, DeepLabV3+ tiene una arquitectura más compleja que UNet, por lo que el tamaño máximo de *batch size* posible para once gigabytes de VRAM es de cuatro imágenes. El valor del peso de clase óptimo para las imágenes de 3 canales es el mismo que para UNet. No hay necesidad de recorte de gradiente ya que no hay problema de gradiente explosivo. Además, esta arquitectura funciona mejor con una tasa de aprendizaje más pequeña que UNet. El mejor valor de regularización L2 coincide con el recomendado por los desarrolladores. Todo el conjunto de entrenamiento se mezcla antes de cada *epoch* para evitar el sobreajuste.

Tabla 3.21: Parámetros de DeepLabV3+.

<b>Parámetros de red</b>	
Tamaño de entrada	640×480×3
Clases	2
<i>Backbone</i>	Xception65
<i>Output stride</i>	16
<i>Padding</i>	Sí

Para lograr tiempos de entrenamiento más rápidos y permitir que el modelo se generalice mejor, el entrenamiento comienza desde un modelo pre-entrenado con el conjunto de datos de ImageNet [26].



Tabla 3.22: Parámetros de entrenamiento para DeepLabV3+.

<b>Parámetros de entrenamiento</b>	
<i>Solver</i>	Adam
<i>Epochs</i>	1000
<i>Batch size</i>	4
<i>Learning rate</i>	0.0005
<i>Class weighting</i>	0.35 - 0.65
<i>Gradient clipping</i>	No
Regularización L2	0.00004
<i>Data augmentation</i>	Escala de 0.5 a 2.0 con saltos de 0.25
<i>Shuffle</i>	Sí

Las métricas de la prueba del experimento de “imágenes de 3 canales” se pueden ver en la Tabla 3.23. Los resultados muestran valores altos para las métricas, por encima del 77 % en puntuación  $F_1$ -Score y con *Precision* y *Recall* equilibrados. Este experimento obtiene resultados aún mejores que el experimento de imágenes de 30 canales con UNet, lo cual es impresionante dada la diferencia entre los experimentos de imágenes de 30 canales y 3 canales en UNet.

Tabla 3.23: Métricas para el experimento con DeepLabV3+.

<b>Experimento</b>	<b><i>Precision</i></b>	<b><i>Recall</i></b>	<b><i>IoU</i></b>	<b><math>F_1</math></b>
3-canales	.760	.786	.629	.773

Este experimento con DeepLabV3+ tarda 01h:18m:27s, más de 2,5 veces más que UNet en las mismas condiciones. Sin embargo, todavía es casi dos veces más rápido que el experimento de imágenes de 30 canales con UNet.

Para acompañar estos resultados, se puede ver una visualización de las imágenes de prueba en la Figura 3.26. En esta figura se observa una gran diferencia con respecto a UNet. Este modelo detecta casi todos los defectos y prácticamente no tiene ruido. Tiene más problemas para detectar defectos de  $5\text{ mm} \times 5\text{ mm}$  en la profundidad máxima. Sin embargo, en la mayoría de las imágenes de prueba se encuentran todos los defectos.

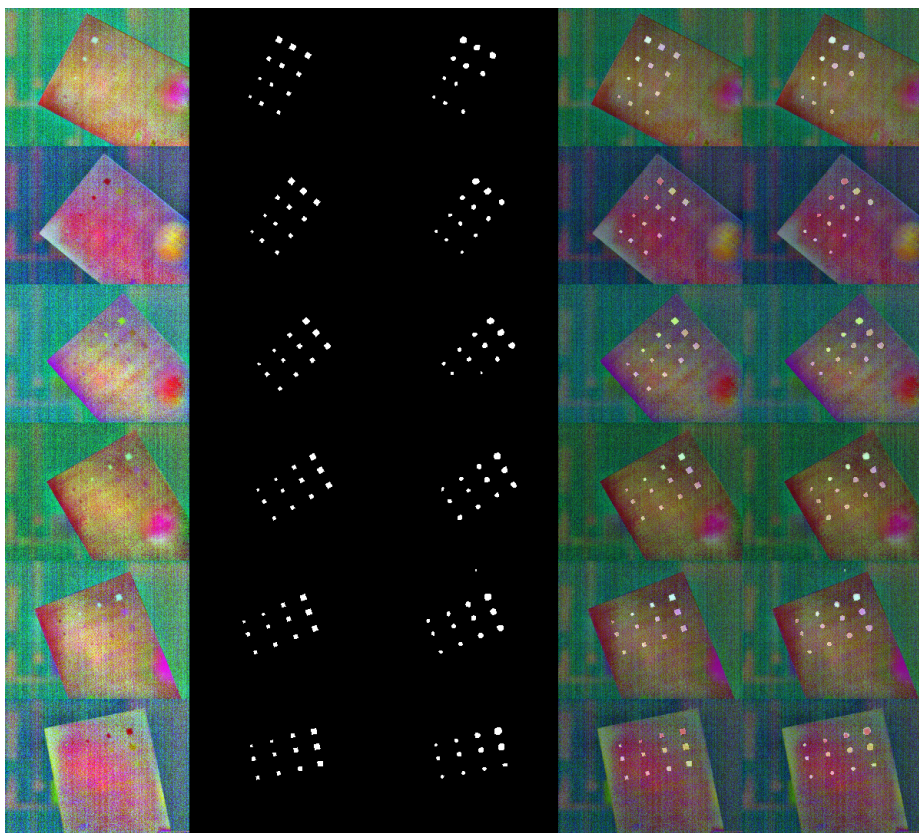


Figura 3.26: Visualización de los resultados predichos para DeepLabv3+. (**1ra columna**) Imágenes originales, (**2da columna**) máscaras de la verdad terrena, (**3ra columna**) predicciones, (**4ta columna**) imágenes originales con máscaras de la verdad terrena, (**5ta columna**) imágenes originales con predicciones.

### 3.2.3. Comparación de resultados

Ni RF ni SVM pueden detectar defectos en los laminados CFRP utilizando los métodos de procesamiento de imagen descritos. Las métricas (Tabla 3.24 y Figura 3.27) y las imágenes de visualización (Figura 3.22 y 3.23) dejan claro que estos métodos no son lo suficientemente fiables, al menos con las características seleccionadas, para detectar defectos con alta confianza. No generalizan lo suficiente. Los datos termográficos generalmente tienen altos niveles de ruido y bajos niveles de contraste. Estas características dan una alta varianza a las características para el mismo defecto, lo que las hace difíciles de detectar para modelos convencionales como RF y SVM.

Tabla 3.24: Métricas para todos los métodos.

Experimento	<i>Precision</i>	<i>Recall</i>	<i>IoU</i>	$F_1$
RF (3-canales)	.103	.132	.061	.116
SVM (3-canales)	.037	.604	.036	.069
UNet (3-canales)	.689	.717	.542	.703
UNet (30-canales)	.764	.726	.593	.745
DeepLabV3+ (3-canales)	.760	.786	.629	.773

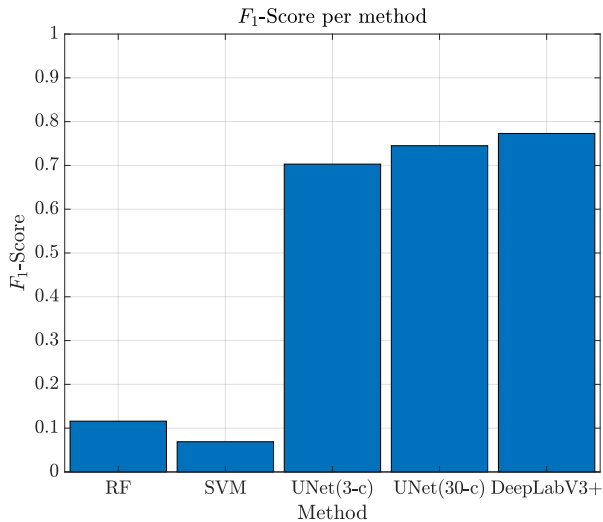


Figura 3.27:  $F_1$ -Score de de cada método.

En el caso de UNet, los resultados son mucho mejores. Con las imágenes de 3 canales, las métricas podrían considerarse bajas. Sin embargo, los defectos son todos distinguibles en las imágenes de visualización aunque hay algo de ruido en las predicciones. En cuanto a las imágenes de 30 canales, las métricas de resultados muestran una clara mejora. El ruido de las predicciones se reduce en gran medida y las imágenes de visualización muestran que se encuentran casi todos los defectos.

DeepLabV3+ tiene un nivel de acierto superior al de UNet incluso cuando solo se pueden usar imágenes de 3 canales. Esta evaluación proporciona los mejores resultados, cercanos al 80% de  $F_1$ -Score. Las imágenes de visualización son más claras que las producidas por UNet y se encuentran casi todos los defectos. DeepLabV3+ es más complejo computacionalmente que UNet en las mismas condiciones, requiriendo más del doble de tiempo de entrenamiento. Sin embargo, DeepLabV3+ sigue siendo casi dos veces más rápido que UNet con 30 canales.

Para modelos de segmentación semántica, a diferencia de los modelos SVM y RF, la reflexión de la lámpara no se clasifica como un defecto. Esto es deseable en inspecciones reales, donde las reflexiones a menudo son inevitables. Esto indica que las características creadas manualmente no son suficientes para aprender que la reflexión no es una característica distintiva de los defectos. Sin embargo, UNet y DeepLabV3+ son capaces de “aprender” que las reflexiones no son una parte distintiva de los defectos. Esto es posible porque al rotar la muestra, la reflexión no siempre está en la misma parte de la muestra.

### 3.2.3.1. Validación de los resultados

Esta sección evalúa nuevos especímenes con diferentes estructuras internas. El objetivo de estas evaluaciones es observar hasta qué punto los modelos de segmentación semántica se generalizan en especímenes nunca vistos. Para este propósito, las detecciones de estos especímenes se ejecutan con los modelos entrenados con el primer espécimen.

El espécimen 2 tiene una estructura similar al espécimen de entrenamiento. Sin embargo, este espécimen tiene la mitad de la profundidad (1,125 mm) y un menor número de capas (6 capas). En este caso, la profundidad de los defectos es de 0,75 mm, 0,56 mm y 0,19 mm de izquierda a derecha. Los tres primeros defectos son defectos de virutas de acero y el resto son defectos de PTFE. (Véase la Figura 3.28).

Como se puede ver en la Figura 3.30, todos los defectos de la pieza son detectados con éxito con DeepLabV3+. Parece que los defectos más pequeños tienen un área predicha mayor que el área de los defectos de la máscara objetivo. UNet también es capaz de detectar casi todos los defectos, pero la imagen predicha tiene más ruido. En la Tabla 3.25 se obtienen métricas para esta evaluación.

Estas métricas presentan menor precisión de la esperada debido a este aumento en el área de los defectos pequeños.

El espécimen 3 tiene una estructura muy diferente a la muestra de entrenamiento. Esta muestra no solo tiene mayor profundidad (20 capas y una profundidad total de 3,825 mm), sino que la capa 7 es de mayor profundidad y reflectividad (ver Figura 3.29). Esta capa tiene un grosor de 20 mm y se llama “núcleo Rohacell”, que es una marca registrada de espumas estructurales que tienen un alto rendimiento mecánico<sup>3</sup>. Estas espumas se han utilizado en el sector aeronáutico durante mucho tiempo para aligerar materiales compuestos y actualmente se utilizan para el mismo propósito en otras industrias, como la automotriz y la eólica.

No hay defectos más profundos que los del núcleo Rohacell porque con la termografía no es posible detectar estos defectos debido a que es un gran aislante térmico. La profundidad de los defectos es de 0,38 mm, 0,75 mm, 1,125 mm de izquierda a derecha. Los tres defectos inferiores son de viruta de acero y el resto son de PTFE.

Como se puede ver en la Figura 3.30, la mayoría de los defectos no se detectan correctamente. Parece que la nueva capa afecta agresivamente la reflectividad y, por lo tanto, al comportamiento del modelo para la detección de defectos. En la Tabla 3.26 se obtienen las métricas para esta evaluación. Estas métricas presentan resultados muy pobres.

Como resultado de estas evaluaciones, se puede observar que siempre y cuando el espécimen evaluado tenga una estructura similar a la del espécimen de entrenamiento, se pueden lograr detecciones de alta calidad incluso si la profundidad del espécimen no es exactamente la misma que en el espécimen de entrenamiento. Sin embargo, si la estructura del espécimen se altera severamente, ya sea mediante la adición de una capa interna con una reflectividad diferente, o cambios de profundidad muy drásticos, los modelos de segmentación semántica no pueden encontrar todos los defectos en el espécimen. En particular, el núcleo de Rohacell cambia las condiciones de frontera del problema de transferencia de calor, lo que afecta los resultados obtenidos en las inspecciones. Finalmente, se demuestra mediante la validación en dos nuevos especímenes, que este método es capaz de generalizar en piezas nunca antes vistas siempre y cuando tengan una estructura interna similar.

---

<sup>3</sup>[www.rohacell.com/en](http://www.rohacell.com/en)

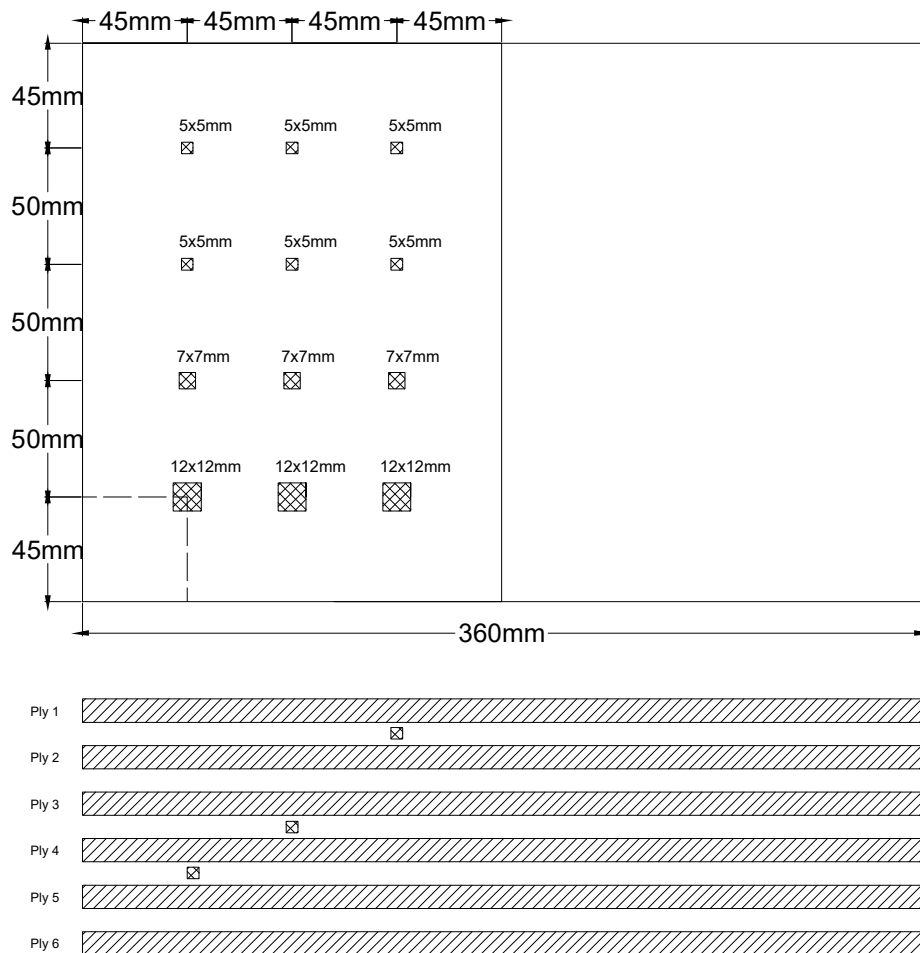


Figura 3.28: Configuración del espécimen 2.

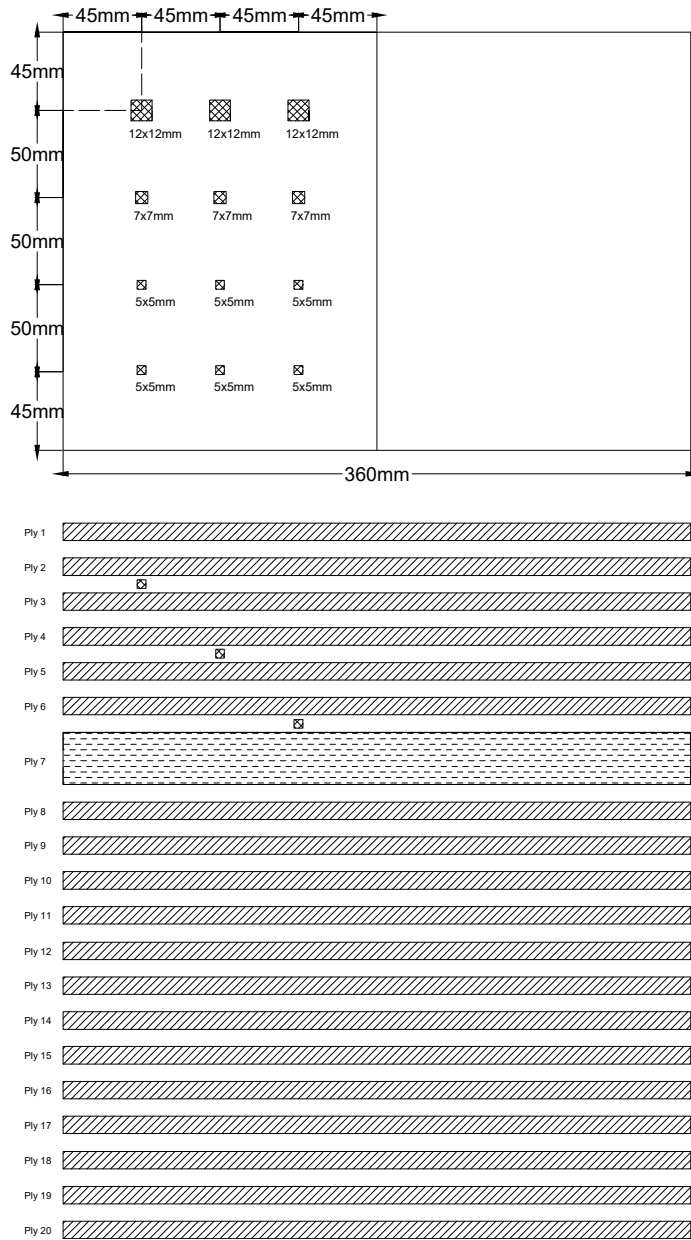


Figura 3.29: Configuración del espécimen 3.

Tabla 3.25: Métricas para el espécimen 2.

<b>Experimento</b>	<i>Precision</i>	<i>Recall</i>
UNet (3-canales)	.56	.47
DeepLabV3+ (3-canales)	.83	.41

Tabla 3.26: Métricas para el espécimen 3.

<b>Experimento</b>	<i>Precision</i>	<i>Recall</i>
UNet (3-canales)	.56	.50
DeepLabV3+ (3-canales)	.84	.32

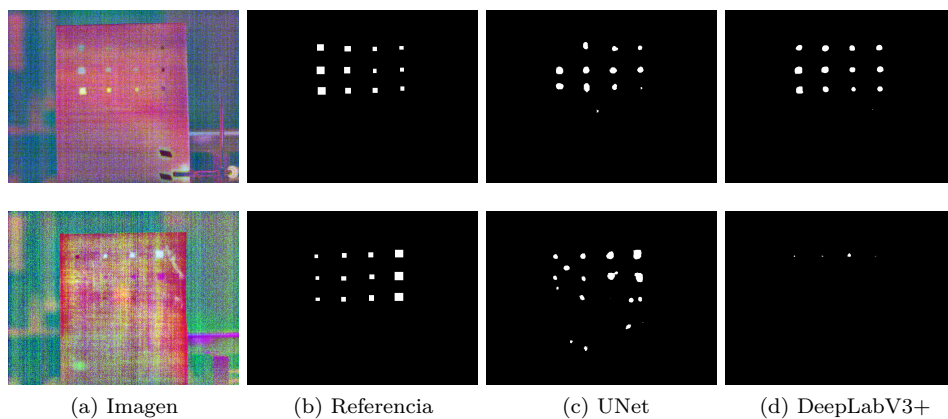


Figura 3.30: Especímenes 2 (arriba) y 3 (abajo).



### 3.3. Emisiones fugitivas en plantas industriales

Las emisiones fugitivas son un problema importante en la industria, ya que estas emisiones pueden contribuir al efecto invernadero y poner en peligro la salud de las personas y los animales que viven cerca. Estas emisiones no salen de una chimenea, por lo que tanto su localización como su forma pueden ser impredecibles. A medida que las leyes de regulación de la contaminación se vuelven más estrictas, las empresas deben encontrar nuevas e innovadoras formas de controlar y prevenir la contaminación de forma rentable para poder competir. Una opción atractiva es el uso de cámaras de vigilancia para la detección basada en visión, ya que estas cámaras no tienen los mismos inconvenientes que los sensores no ópticos y pueden ser utilizadas durante el día y la noche. Este trabajo estudia la posibilidad de utilizar cámaras de vigilancia y adaptar la infraestructura existente en las plantas industriales para detectar emisiones fugitivas. Aunque este método no reconoce químicamente las emisiones, es capaz de detectar emisiones basándose en ejemplos de emisiones previas y puede ser utilizado para localizar y evaluar el tamaño de las emisiones.

En esta investigación se generan tres conjuntos de datos separados, cada uno con imágenes de una planta industrial diferente (Plant1, Plant2 y Plant3). Cada planta industrial proporciona imágenes desde una única cámara. Estas cámaras ya existían como cámaras de vigilancia para monitorizar edificios con riesgo de emisiones fugitivas. Sin embargo, el proceso existente era manual. Las imágenes están censuradas para proteger el anonimato de la empresa que proporciona esta información. Sin embargo, toda la experimentación se ha realizado con las imágenes originales.

Los conjuntos de entrenamiento y prueba se dividen en un 75 % y un 25 % respectivamente. Todas las imágenes se toman en momentos aleatorios en cuatro días diferentes, excluyendo la noche ya que las cámaras no tienen suficiente visibilidad. La separación mínima de tiempo entre dos imágenes es de 5 segundos. Las cámaras tienen un sensor 4K/8MP, una velocidad de 30 fotogramas por segundo y una distancia focal variable. La calidad está ligeramente degradada debido al uso de zoom para ajustar las imágenes a los edificios correspondientes. El objetivo de las cámaras de vigilancia es tener un campo de visión más amplio para cubrir la mayor cantidad posible de terreno, pero el área de interés es solo una zona específica de la imagen. Además, son cámaras con poco mantenimiento ya que se encuentran en lugares inaccesibles, por lo que es común que exista suciedad en las lentes. Todas las imágenes constan de bandas rojas, verdes y azules (RGB) y tienen una resolución de  $2.048 \times 1.536$  píxeles o  $1.024 \times 768$  píxeles. Para aliviar la cantidad de memoria necesaria en VRAM para entrenar los modelos, y mantener todas las imágenes del mismo tamaño, todas las imágenes se escalan a  $512 \times 384$  píxeles utilizando la interpolación bicúbica.







En caso de uso real existe una proporción de 1:34 de imágenes con emisiones a imágenes sin emisiones. Durante un día normal, en promedio, se capturó una sola imagen con emisiones fugitivas por cada 34 imágenes sin emisiones. Para probar cómo los modelos entrenados con diferentes proporciones funcionan en un entorno real, se creó y evaluó un conjunto de prueba de 250 imágenes de emisiones fugitivas y 8.500 imágenes sin emisiones para Plant1 y Plant2. Además, para estudiar la importancia de la proporción utilizada para entrenar un modelo que normalmente se utilizará bajo otras proporciones (proporciones realistas), se diseñan múltiples variaciones para cada conjunto de datos. Cada variación utiliza un número diferente de imágenes sin emisión. En la Tabla 3.27, se muestran las diferentes proporciones para los conjuntos de datos de Plant1 y Plant2 junto con el número de imágenes que corresponden a emisiones y no emisiones para los conjuntos de entrenamiento y prueba. En todas las variaciones se mantiene el mismo conjunto de imágenes con emisiones fugitivas para mejorar la comparabilidad. El conjunto de datos Plant3 solo tiene la proporción 2:1 ya que no tiene suficientes imágenes.

Tabla 3.27: Proporciones para los diferentes conjuntos de datos. (emisión:sin emisión)

Proporción	Entrenamiento	Prueba
2:1	750:375	250:125
1:1	750:750	250:250
1:2	750:1.500	250:500
1:4	750:3.000	250:1.000

Plant1 tiene las siguientes clases: edificio, nube, cielo, fuego, chimeneas de vapor de agua y emisión fugitiva. Plant2 tiene: edificio, nube, cielo, chimeneas de vapor de agua y emisión fugitiva. Plant3 tiene: edificio, nube, cielo y emisión fugitiva. El etiquetado fue realizado por expertos utilizando herramientas de software y revisado por los operadores de las plantas industriales.

Las clases en las máscaras objetivo tienen los siguientes colores asociados:

-  Edificio
-  Chimenea de vapor de agua
-  Nube
-  Fuego
-  Emisión fugitiva
-  Cielo

Las emisiones fugitivas son la clase objetivo de este estudio de evaluación. Ésta es la única clase de interés, por lo tanto, se puede estudiar una comparación entre la clasificación multi-clase y la clasificación binaria y sus efectos en la clase objetivo.

Esta investigación se centra en DeepLabV3+ debido a que se busca analizar la creación de un conjunto de datos y de su entrenamiento en lugar de la comparación con otros métodos. En una experimentación previa se ha obtenido que DeepLabV3+ tiene un mayor grado de acierto que UNet, RF o SVM. La mejor configuración de DeepLabV3+ encontrada mediante un estudio de ablación para los tres conjuntos de datos se puede ver en la Tabla 3.28. Como los tres conjuntos de datos son similares, las mejores configuraciones coinciden.

Tabla 3.28: Hiperparámetros para DeepLabV3+.

<b>Parámetros de entrenamiento</b>	
Entrada	512×384×3
<i>Backbone</i>	Xception65
<i>Output stride</i>	16
<i>Padding</i>	Sí
<i>Solver</i>	Adam
<i>Epochs</i>	80
<i>Batch size</i>	6
<i>Learning rate</i>	0.00005
<i>Gradient clipping</i>	No
Regularización L2	0.0004
<i>Data augmentation</i>	Escalado de 0.5 a 2.0 con saltos de 0.25
<i>Shuffle</i>	Sí

### 3.3.1. Detección de emisiones

#### 3.3.1.1. Emisiones diurnas

En primer lugar, se llevaron a cabo experimentos de múltiples clases para determinar si se puede obtener una segmentación de emisiones fugitivas en plantas industriales a partir de imágenes de cámaras de vigilancia. Esta sección muestra los resultados de los experimentos utilizando DeepLabV3+ para la segmentación de múltiples clases. Estos experimentos utilizan el método MFW para ponderar el desequilibrio de clases.

Las métricas de la Tabla 3.29 muestran valores de  $F_1$ -Score altos, superando la barrera del 80 % en cada clase para los tres conjuntos de datos, lo que demuestra que se puede obtener una máscara de detección robusta.

Las emisiones fugitivas son una de las clases con los valores de  $F_1$ -Score más bajos y sus métricas suelen estar ligeramente sesgadas hacia *Recall* sobre *Precision*. Estas métricas son de esperar ya que la clase de emisiones fugitivas es mucho más difícil de ver y tiene una mayor variabilidad en el área y color que otras clases. Esto hace que tanto la red como las máscaras objetivo sean propensas a errores. De la misma manera, como es difícil distinguir entre ellas, las clases Nube y Cielo podrían aumentar sus métricas al fusionarse en la misma clase.

A partir de estos resultados, se puede establecer que es posible diferenciar entre nubes, chimeneas de vapor de agua y emisiones fugitivas.

Tabla 3.29: Métricas para los experimentos de clasificación multi-clase.

Conjunto	Clase	<i>Precision</i>	<i>Recall</i>	<i>IoU</i>	$F_1$
Plant1	Edificio	0.995	0.997	0.992	0.996
	Vapor	0.915	0.900	0.831	0.907
	Nube	0.955	0.834	0.802	0.890
	Fuego	0.794	0.837	0.688	0.815
	<b>Emisión</b>	0.836	0.832	0.715	0.834
	Cielo	0.926	0.951	0.884	0.938
Plant2	Edificio	0.997	0.988	0.985	0.992
	Vapor	0.744	0.934	0.707	0.847
	Nube	0.800	0.931	0.755	0.861
	<b>Emisión</b>	0.767	0.895	0.704	0.826
	Cielo	0.984	0.935	0.921	0.959
Plant3	Edificio	0.992	0.994	0.986	0.993
	Nube	0.779	0.710	0.591	0.743
	<b>Emisión</b>	0.881	0.939	0.833	0.909
	Cielo	0.954	0.936	0.895	0.945

Aunque la segmentación multi-clase puede localizar con éxito emisiones fugitivas en plantas industriales, es interesante evaluar y comparar la segmentación multi-clase con la segmentación binaria. De esta forma se puede reducir la complejidad a la hora de crear las máscaras objetivo. Es mucho más barato y sencillo construir máscaras objetivo con una sola clase en lugar de múltiples clases que requieren varias regiones diferentes por imagen. Además, el proceso requerido para ajustar la red es mucho más simple, ya que los parámetros como el equilibrio de clases son más fáciles de establecer. Esto se debe a que solo hay dos clases, por lo que modificar la ponderación de una solo afecta a la otra en lugar de afectar a varias clases al mismo tiempo. Esto permite un estudio en profundidad del equilibrio de clases. La Tabla 3.30 solo muestra los resultados para la clase de emisiones. La clase “sin emisión” no se detalla ya que no es necesaria.

Tabla 3.30: Métricas para los experimentos de clasificación binaria.

<b>Conjunto</b>	<b><i>Precision</i></b>	<b><i>Recall</i></b>	<b><i>IoU</i></b>	<b><math>F_1</math></b>
Plant1	0.634	0.958	0.617	0.763
Plant2	0.648	0.965	0.633	0.775
Plant3	0.885	0.938	0.836	0.911

Los resultados de los experimentos de ponderación de clase MFW para la segmentación binaria que se muestran en la Tabla 3.30 son comparables a los del experimento de segmentación multi-clase de la Tabla 3.29. Estos experimentos están más sesgados hacia la *Recall*, obteniendo valores muy altos pero acompañados de una menor *Precision*. A pesar de que existe una disparidad entre *Recall* y *Precision*, los valores de  $F_1$ -Score no están lejos de los obtenidos en los experimentos multi-clase (ver Tabla 3.29). El uso de la clasificación binaria en lugar de la multi-clase hace que el desequilibrio entre *Recall* y *Precision* sea mayor, disminuyendo la calidad de las segmentaciones. Plant3 es la excepción ya que los resultados multi-clase y binarios son prácticamente iguales. Esto indica que estas diferencias dependen en gran medida del conjunto de datos utilizado y de su complejidad. En este caso Plant3 no tiene elementos como chimeneas de vapor de agua o fuego. En vista de estos resultados, se puede concluir que fusionar todas las clases no objetivo aumenta el desequilibrio del número de muestras respecto a la clase objetivo. En otras palabras, la diferencia en el número de píxeles entre las dos clases aumenta, lo que provoca que la *Recall* sea alta, pero la *Precision* disminuya.

Los resultados de los experimentos de segmentación binaria muestran que sus métricas están sesgadas hacia el *Recall*. Esto disminuye la calidad de las detecciones: las métricas de *Recall* y *Precision* deben estar equilibradas. Para aliviar el problema del desequilibrio del número de muestras entre clases se utiliza la ponderación de clases. El peso de las clases tiene un gran impacto en el proceso de entrenamiento del modelo. Los métodos IFW y MF para ajustar el peso de las clases se utilizan a menudo. Sin embargo, los resultados muestran que no son capaces de aliviar el desequilibrio entre clases. Por este motivo, se estudian pesos personalizados, configurados mediante un proceso manual, para lograr un mejor equilibrio. Plant3 no se incluye en este estudio ya que sus métricas ya están equilibradas. La segmentación binaria se beneficia del hecho de que sólo se necesita encontrar el mejor peso para una clase. Esto simplifica en gran medida el estudio de cómo el peso de las clases afecta a los resultados obtenidos.

En la Tabla 3.31 se prueban múltiples valores para el peso de las clases para estudiar el comportamiento de la red. Para hacer más fácil la experimentación, todos los valores se normalizan entre 0 y 1. De esta manera, si se utiliza un valor de 0,60 para la clase objetivo, se asume que valor restante (0,40) se utiliza para la clase no objetivo.

Tabla 3.31: Métricas para los experimentos de clasificación binaria con ponderación de clases personalizada.

Conjunto	Peso	<i>Precision</i>	<i>Recall</i>	<i>IoU</i>	$F_1$
Plant1	(MFW) 0.88	0.634	0.958	0.617	0.763
Plant1	0.70	0.765	0.884	0.695	0.820
Plant1	0.60	0.826	0.841	0.715	<b>0.833</b>
Plant2	(MFW) 0.923	0.648	0.965	0.633	0.775
Plant2	0.80	0.779	0.891	0.711	0.831
Plant2	0.70	0.842	0.842	0.727	<b>0.842</b>
Plant2	0.60	0.866	0.779	0.695	0.820

Las métricas de la Tabla 3.31 muestran que los pesos personalizados pueden equilibrar la *Recall* y la *Precision* y mejorar la calidad de las detecciones, aumentando su  $F_1$ -Score. Los valores de los pesos personalizados para la clasificación binaria logran resultados aún mejores que los de clasificación multi-clase con MFW. Equilibrar tanto la *Recall* como la *Precision* tiene un gran impacto en la calidad del modelo. Después de equilibrar correctamente la *Recall* y la *Precision*, no hay beneficio en agregar clases que no sean las clases objetivo para segmentar. En este caso, sólo la clase de emisiones fugitivas es de valor. Por esta razón, y debido a las ventajas mencionadas anteriormente, el resto del estudio se centrará en la segmentación binaria para la clase de emisiones fugitivas.

La segmentación binaria con pesos de clase personalizados tiene resultados

sólidos. Para validar aún más estos resultados, se evalúa un conjunto de pruebas más grande. Este conjunto de pruebas tiene como objetivo replicar las proporciones en las que las imágenes muestran emisiones fugitivas durante todo un día. En este caso, por cada 34 imágenes sin emisión, se produce una imagen con emisión. Por simplicidad, esto se denomina proporción 1:34.

En el caso de Plant3 no hay suficientes imágenes para crear este tipo de conjunto de pruebas. Para Plant1 y Plant2 se usan 250 imágenes con emisión y 8.500 imágenes sin emisión.

Cuando se prueba una red entrenada con un conjunto de proporción 2:1 con un conjunto de proporción real de 1:34, la *Precision* se reduce considerablemente debido a nuevos falsos positivos (ver Tabla 3.32). Sin embargo, el *Recall* sigue siendo el mismo ya que no se agregaron nuevas imágenes con emisión fugitiva a la prueba. Este efecto se puede ver en la primera fila para Plant1 y en la quinta fila para Plant2 en la Tabla 3.32. Para resolver este problema, se realizaron experimentos con diferentes proporciones de clase. El objetivo era determinar si un cambio en las proporciones durante el entrenamiento afecta la calidad de las detecciones al ajustar el peso de las clases para equilibrar el *Recall* y la *Precision*. La Tabla 3.32 muestra el mejor experimento de pesos de clase personalizados para cada proporción.

Tabla 3.32: Métricas para los experimentos de proporción.

Parámetros			Prop. entrenamiento				Prop. reales (1:34)			
Plant	Prop.	Peso	<i>P</i>	<i>R</i>	<i>IoU</i>	<i>F<sub>1</sub></i>	<i>P</i>	<i>R</i>	<i>IoU</i>	<i>F<sub>1</sub></i>
1	2:1	.60	.826	.841	.715	.833	.585	.841	.527	.690
1	1:1	.51	.831	.787	.678	.808	.678	.787	.573	.728
1	1:2	.60	.833	.815	.700	.824	.819	.815	.691	.817
1	1:4	.50	.783	.850	.688	.815	.540	.850	.493	.660
2	2:1	.70	.842	.842	.727	.842	.365	.842	.342	.509
2	1:1	.55	.852	.826	.723	.839	.786	.826	.675	.806
2	1:2	.635	.849	.816	.713	.832	.835	.816	.702	.825
2	1:4	.57	.843	.823	.714	.833	.815	.823	.694	.819

En la Tabla 3.32 se observa que una proporción de una imagen de emisión por cada dos imágenes sin emisión (1:2) es la proporción óptima en ambos conjuntos de datos. Una proporción más alta de imágenes sin emisión le da al modelo mayor estabilidad y menos falsos positivos. Este descubrimiento es vital ya que las pruebas de entrenamiento estándar obtienen resultados excelentes. Si no se realizara una prueba con proporciones reales, se podría hacer la suposición errónea de que los modelos pueden generalizar de igual forma en un escenario real. El entrenamiento utilizando una proporción realista de 1:34 es impráctico, ya que el tiempo requerido para configurar y entrenar tal modelo sería demasiado largo; el

número de imágenes requeridas está fuera del alcance de esta investigación. Por esta misma razón, no se realizan experimentos con proporciones superiores a 1:4, aunque tales experimentos serían interesantes para confirmar esta afirmación o para encontrar un límite superior.

Dado que se requiere una gran cantidad de configuraciones de parámetros, es razonable intentar aprovechar los modelos entrenados en una planta industrial para ser utilizados en otra. Por este motivo, se llevaron a cabo experimentos utilizando un modelo entrenado con imágenes de una planta industrial para evaluarlo con imágenes de otra. Los resultados de esta experimentación se muestran en la Tabla 3.33.

Tabla 3.33: Métricas para los experimentos de test cruzado.

<b>Conjunto</b>	<b><i>Precision</i></b>	<b><i>Recall</i></b>	<b><i>IoU</i></b>	<b><math>F_1</math></b>
Plant1 → Plant2	0.728	0.380	0.333	0.499
Plant1 → Plant3	0.446	0.129	0.111	0.200

Las métricas de la Tabla 3.33 muestran que la prueba con diferentes conjuntos de datos no produce detecciones de buena calidad. Los modelos no son directamente utilizables con otros conjuntos de datos. Es evidente que los modelos no se pueden utilizar indiscriminadamente en diferentes plantas industriales. Si se desea utilizar este enfoque, se necesita entrenar un modelo para cada planta industrial.

Debido a que es necesario entrenar un modelo para cada planta industrial, se estudia el mínimo número de imágenes necesario para entrenar un modelo y obtener métricas similares a las obtenidas con un conjunto de datos más grande. Esto podría reducir significativamente el tiempo y los costos. Los experimentos con un número reducido de imágenes para Plant2 y Plant3 se evalúan en la Tabla 3.34. Para estos experimentos, se mantiene una proporción de 2:1 para facilitar las comparaciones. Los conjuntos de prueba no se modifican para hacer que los resultados sean directamente comparables.

Tabla 3.34: Métricas para el conjunto de entrenamiento reducido.

<b>Conjunto</b>	<b>Imágenes</b>	<b><i>Precision</i></b>	<b><i>Recall</i></b>	<b><i>IoU</i></b>	<b><math>F_1</math></b>
Plant2	100	0.668	0.834	0.590	0.742
Plant2	250	0.697	0.837	0.614	0.761
Plant3	100	0.847	0.895	0.770	0.870
Plant3	250	0.862	0.890	0.779	0.876



El entrenamiento reducido muestra que el uso de solo 100 imágenes es suficiente para obtener resultados aceptables, como se muestra en la Tabla 3.34. Estas métricas son casi un 5 % más bajas que los experimentos de la Tabla 3.30, que utilizan 1.125 imágenes. Sin embargo, una reducción del 5 % utilizando un conjunto de datos diez veces más pequeño puede ser aceptable cuando existen restricciones de tiempo y de costos.

Dado que se necesitan al menos 100 imágenes para lograr un entrenamiento robusto, es interesante estudiar si se puede reducir aún más este número reutilizando un modelo ya entrenado en otra planta industrial. Incluso si no se puede utilizar directamente, quizás otro modelo pueda servir como punto de control para reducir el tiempo de entrenamiento en otros conjuntos de datos. Por este motivo, se llevan a cabo experimentos para observar si un modelo entrenado para una planta industrial puede ser utilizado como base para el entrenamiento con un nuevo conjunto de datos para otra planta industrial (es decir, aprendizaje por transferencia). Esto significaría entrenar con menos imágenes. Estos modelos son entrenados con las mismas imágenes para permitir comparaciones.

Los resultados del aprendizaje por transferencia de la Tabla 3.35 muestran una mejora del 3 % en  $F_1$ -Score para Plant2 con 100 imágenes en comparación con el entrenamiento reducido de 100 imágenes de la Tabla 3.34. Sin embargo, en el caso de Plant3 con 100 imágenes, esto no mejora sus métricas e incluso reduce su  $F_1$ -Score en un 2 % en el experimento de 100 imágenes. Esto muestra que los modelos podrían depender demasiado del conjunto de datos utilizado y no se pueden generalizar para la aplicación de segmentación de emisiones. Dado que ya es posible entrenar con solo 100 imágenes, la transferencia de aprendizaje puede no ser necesaria. Por lo tanto, la el aprendizaje por transferencia no se puede utilizar para reducir la cantidad de imágenes necesarias.

Tabla 3.35: Métricas para los experimentos de transferencia de aprendizaje.

Conjunto	Imágenes	<i>Precision</i>	<i>Recall</i>	<i>IoU</i>	$F_1$
Plant1 → Plant2	10	0.698	0.620	0.489	0.657
Plant1 → Plant2	50	0.777	0.683	0.571	0.727
Plant1 → Plant2	100	0.788	0.757	0.629	0.772
Plant1 → Plant2	250	0.729	0.815	0.625	0.770
Plant1 → Plant3	10	0.801	0.801	0.668	0.801
Plant1 → Plant3	50	0.808	0.927	0.760	0.863
Plant1 → Plant3	100	0.837	0.871	0.745	0.854
Plant1 → Plant3	250	0.841	0.916	0.781	0.877

Una vez obtenido un modelo robusto para la segmentación de emisiones fugitivas, parece razonable usar estas segmentaciones para detectar y notificar posibles emisiones fugitivas. Para generar una alarma cuando se detecta una emisión fugitiva, es necesario aplicar segmentación semántica a cada fotograma de los vídeos de las cámaras. Esta alarma se usa para indicar si hay o no emisiones fugitivas en la imagen. De esta manera, las detecciones se utilizan para la clasificación de imágenes. Se utiliza el conjunto de pruebas realista para estudiar su comportamiento.

Para evitar generar alarmas para falsos positivos que consisten en regiones extremadamente pequeñas, se necesita un umbral de área para las áreas segmentadas. Este umbral se mide como el porcentaje mínimo de píxeles con emisión sobre el total de píxeles de la imagen. En esta sección, se calculan los umbrales de área óptimos utilizando los modelos del conjunto de pruebas realista de Plant1 y Plant2.

Para cada umbral evaluado, usando pasos del 0.01 %, se determinan las imágenes del conjunto de datos que se considerarán como emisiones verdaderas. Luego, se analizan las imágenes detectadas con y sin emisión en función del umbral utilizado, obteniendo las métricas de *Recall*, *Precision* y *F<sub>1</sub>-Score*. Estas métricas se representan en las Figuras 3.31 y 3.32. Cabe señalar que aunque las métricas utilizadas son las mismas, en este caso se calculan por imagen y no por píxel.

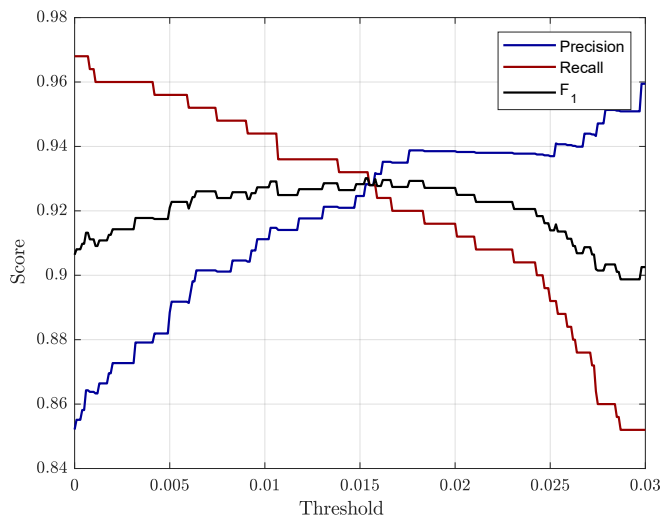


Figura 3.31: Estudio de umbrales (Plant1).

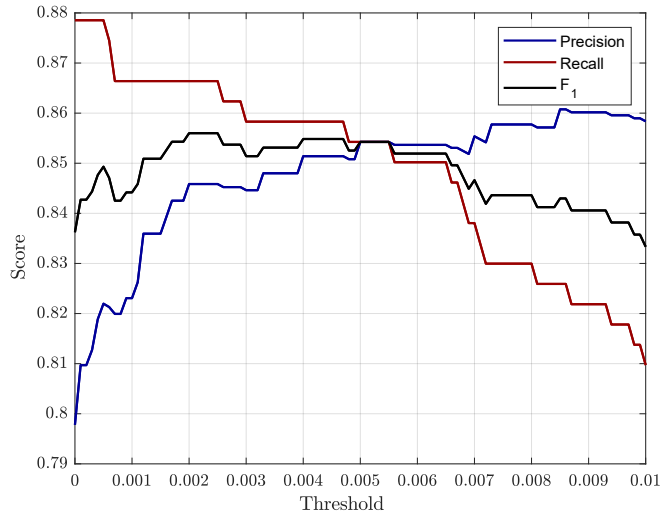


Figura 3.32: Estudio de umbrales (Plant2).

Plant1 tiene un umbral óptimo del 1.56 % del área de la imagen total (Figura 3.31). Plant2 tiene un umbral óptimo del 0.5 % del área de la imagen total (Figura 3.32). Este umbral puede variar dependiendo de factores como la distancia de la cámara al edificio.

Las métricas de la Tabla 3.36 muestran que se logran valores de  $F_1$ -Score de más del 85 % y casi el 93 % para Plant1 y Plant2, respectivamente. Estos experimentos demuestran que este método puede ser utilizado con éxito para la clasificación de imágenes. Además, las regiones de segmentación detectadas se pueden utilizar para determinar diferentes niveles de severidad basados en el área o la forma. Por ejemplo, se podría declarar un nivel de severidad de emisión fugitiva del 1 al 10 utilizando incrementos del 10 % del área de las emisiones. En otras palabras, un nivel de gravedad de 3 podría ser el 30 % del área del cielo ocupado por las emisiones.

Tabla 3.36: Métricas para la alarma.

Conjunto	Umbral	<i>Precision</i>	<i>Recall</i>	<i>F<sub>1</sub></i>
Plant1	0.0050	0.854	0.854	0.854
Plant2	0.0156	0.928	0.928	0.928

### 3.3.1.2. Emisiones nocturnas

La detección de emisiones nocturnas tiene una mayor complejidad debido a que las emisiones no son visibles en cámaras de baja calidad. Por este motivo, se ha instalado una cámara de mayor calidad. Además, esta cámara tiene una nueva banda infrarroja. En esta experimentación se añade el uso de UNet para posteriormente hacer comparaciones con imágenes infrarrojas. En este caso, se utilizan imágenes RGB que tienen emisiones fugitivas visibles. De igual forma que con el resto de conjuntos de datos, las imágenes son escaladas a  $512 \times 384$  para cumplir con los requisitos de VRAM para el entrenamiento de los modelos.

En la Tabla 3.37 se muestran los resultados de día y de noche de forma separada. Se puede observar que en ambos casos las métricas de clasificación de píxeles obtenidas son similares. Se puede concluir que con una buena cámara se pueden detectar emisiones fugitivas nocturnas. En este caso, UNet y DeepLabV3+ obtienen métricas similares, aunque, curiosamente UNet obtiene mejores resultados por la noche y DeepLabV3+ por el día.

Tabla 3.37: Métricas de UNet y DeepLabV3+ para Día o Noche.

<b>Experimento</b>	<b>Tiempo</b>	<b>Precision</b>	<b>Recall</b>	<b>IoU</b>	<b>F<sub>1</sub></b>
UNet	Día	0.708	0.783	0.592	0.744
DeepLab	Día	0.728	0.816	0.625	0.769
UNet	Noche	0.804	0.795	0.665	0.799
DeepLab	Noche	0.716	0.900	0.663	0.798

Debido a los buenos resultados obtenidos, se prueba a ejecutar un entrenamiento con un conjunto de datos que incluye tanto las imágenes de día como de noche. En la Tabla 3.38 se muestran los resultados de esta experimentación para UNet. Las métricas se obtienen para los conjuntos de Día y de Noche de forma separada para poder comparar los resultados obtenidos con la Tabla 3.37. Los resultados de día no solo se igualan sino que se mejoran ligeramente. Sin embargo, Los resultados de noche disminuyen ligeramente. Los resultados indican que es posible entrenar un modelo con imágenes de día y de noche de forma simultanea.

Tabla 3.38: Métricas de Día y de Noche para UNet entrenado con el conjunto de datos Día-Noche.

<b>Experimento</b>	<b>Precision</b>	<b>Recall</b>	<b>IoU</b>	<b>F<sub>1</sub></b>
Día	0.713	0.799	0.605	0.754
Noche	0.697	0.874	0.633	0.776

### 3.3.1.3. Cámaras infrarrojas

En ultima instancia se prueba a experimentar haciendo uso del canal infrarrojo de la nueva cámara. De esta forma se espera mejorar las detecciones. Para estos experimentos se utiliza UNet debido a su capacidad para adaptar su arquitectura para permitir imágenes de cuatro canales (RGB + I). En la Tabla 3.39 se muestran los resultados de los conjuntos de datos de día y de noche de forma separada para realizar comparaciones y observar si se producen resultados diferentes.

Tabla 3.39: Métricas para la comparación entre imágenes RGB y RGBI en Día y Noche.

<b>Experimento</b>	<b>Precision</b>	<b>Recall</b>	<b>IoU</b>	<b>F<sub>1</sub></b>
Día	0.708	0.783	0.592	0.744
Día RGBI	0.763	0.728	0.594	0.745
Noche	0.804	0.795	0.665	0.799
Noche RGBI	0.783	0.816	0.666	0.799

Los resultados de la Tabla 3.39 muestran que no se produce ninguna mejora sobre usar solamente imágenes RGB. Esto puede indicar que el canal infrarrojo no aporta información nueva al modelo. Aunque los resultados no empeoran, es mejor no utilizar este canal infrarrojo para reducir el tiempo de entrenamiento del modelo.

### 3.3.1.4. Resultados y discusiones

La clasificación binaria mediante la segmentación semántica obtiene buenos resultados en los tres conjuntos de datos (Plant1, Plant2 y Plant3) validando su usabilidad en diferentes entornos. En la Figura 3.33 se muestra el mejor experimento de segmentación binaria para Plant1. Se puede observar que a excepción de la imagen de la segunda fila, el resto de detecciones son indistinguibles de sus máscaras objetivo. Esto mismo se repite con el mejor experimento de segmentación binaria de Plant2 en la Figura 3.34. Finalmente, el mejor experimento de segmentación binaria para Plant3 en la Figura 3.35 obtiene unos resultados visualmente perfectos. Para el caso de las imágenes nocturnas se muestra la Figura 3.36. Se puede observar que se producen detecciones idénticas a las máscaras objetivo a pesar de la dificultad aumentada de la tarea.

Cabe destacar que estas figuras contienen solamente imágenes del conjunto de prueba, es decir, imágenes nunca vistas por los modelos. A pesar de que a nivel visual es difícil obtener la silueta de la emisión, estos modelos son capaces de detectar una silueta idéntica a la esperada por los operadores incluso en situaciones desfavorables demostrando la robustez de los modelos.

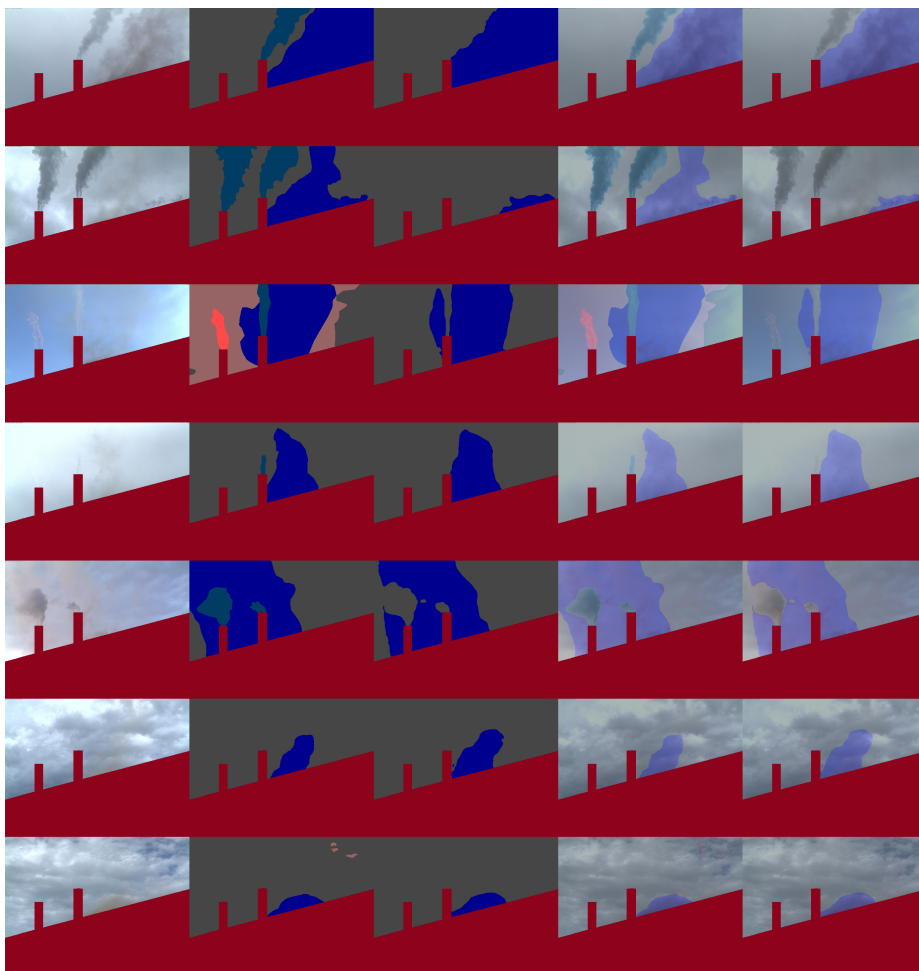


Figura 3.33: Mejor clasificación binaria de Plant1.

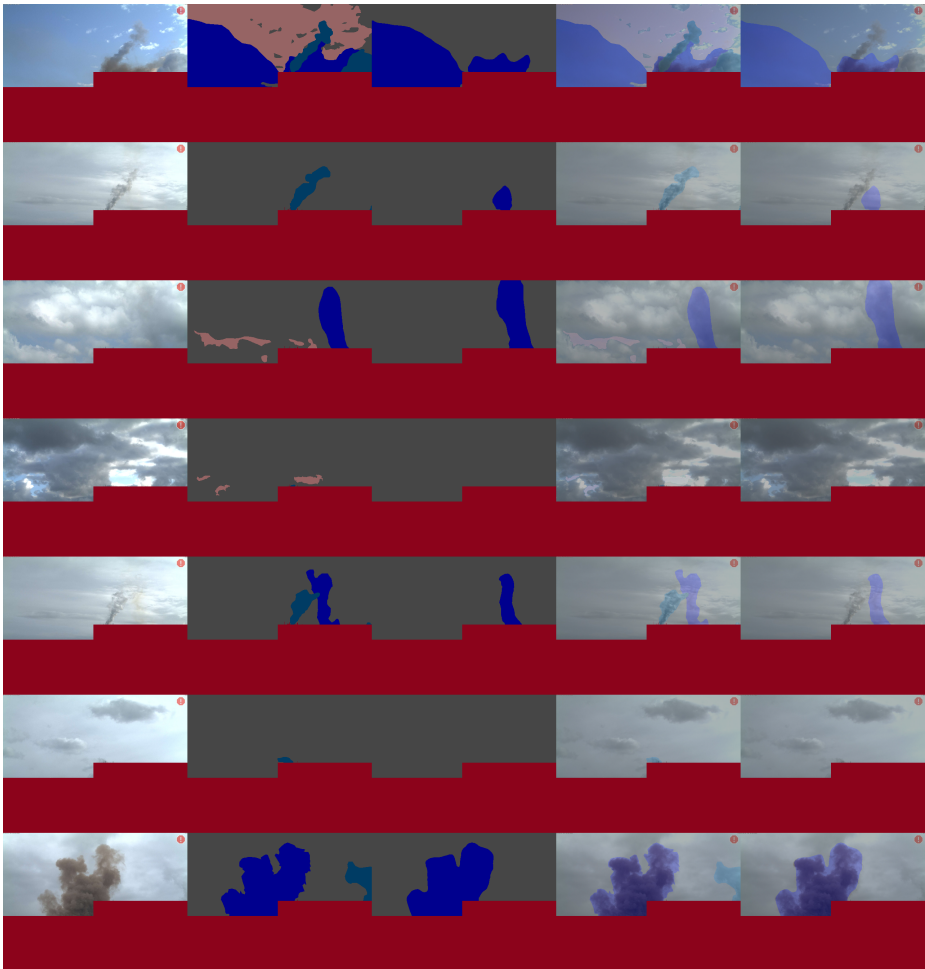


Figura 3.34: Mejor clasificación binaria de Plant2.

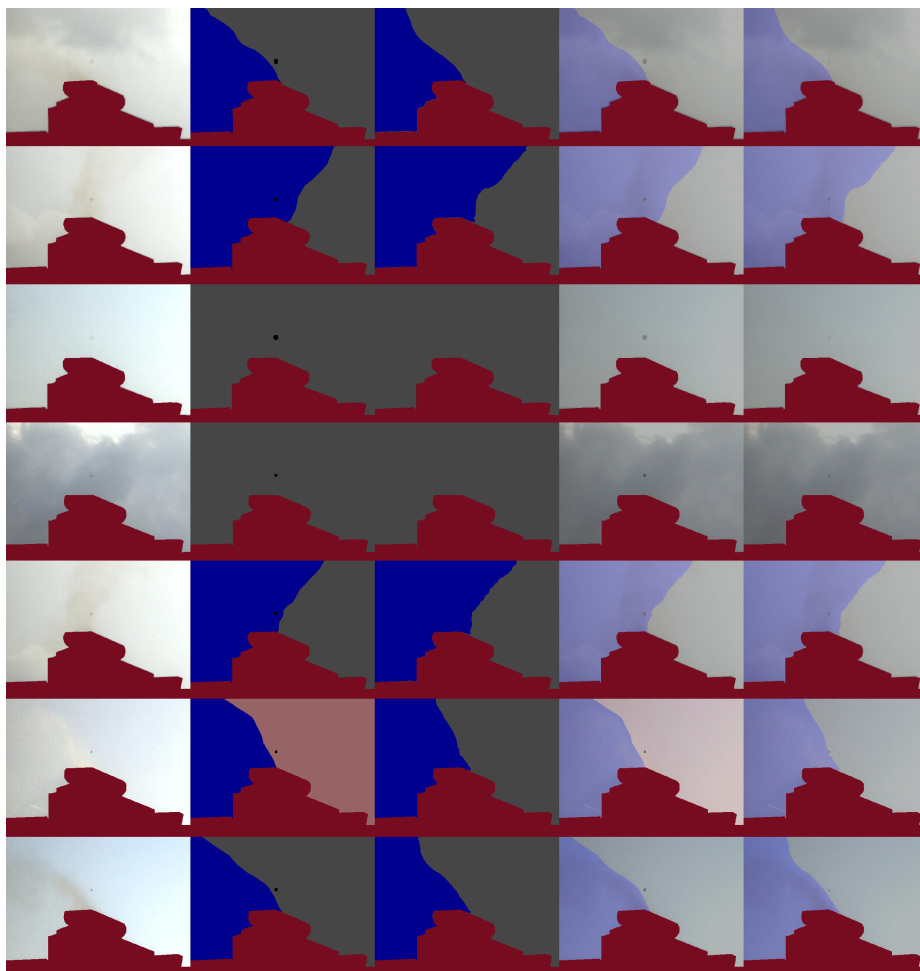


Figura 3.35: Mejor clasificación binaria de Plant3.



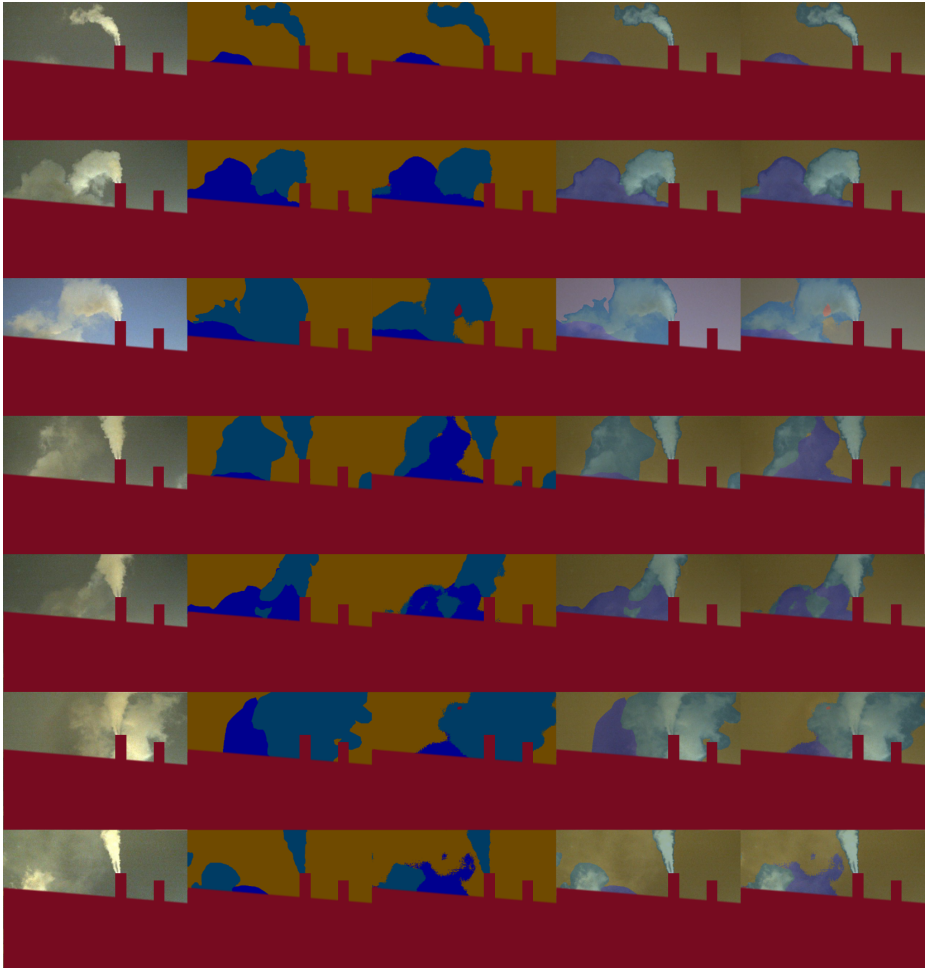


Figura 3.36: Mejor clasificación binaria de emisiones nocturnas con UNet para imágenes RGB.

### 3.3.2. Estimación de la opacidad

A partir de las detecciones generadas por los modelos anteriores. Se pueden obtener las regiones exactas de las emisiones fugitivas así como las regiones asociadas al cielo y a los edificios. Con esta información se puede generar un método capaz de estimar la opacidad de las emisiones fugitivas de forma automatizada. Hoy en día no existe ningún método en la literatura capaz de realizar una estimación completamente automatizada. Todos los métodos requieren de, al menos, la intervención de un operario en alguno de los pasos. De esta forma, en este estudio se propone un método completamente automatizado.

#### 3.3.2.1. Solución propuesta

Los píxeles del cielo más cercanos a la emisión se utilizan para aquellos métodos que requieren obtener la referencia del cielo. Para este propósito, se expande la región de la emisión y se seleccionan los píxeles pertenecientes a la clase cielo. El número de píxeles del cielo seleccionados es igual al número de píxeles de la emisión. Por lo tanto, se obtienen regiones de emisión y cielo del mismo tamaño. Para el proceso de selección de los píxeles del cielo, se descartan los píxeles intermedios entre la emisión y el cielo para evitar errores en el etiquetado de las regiones. Para ello, se dilata la máscara detectada para la emisión usando una estructura en forma de cruz de  $3 \times 3$  píxeles. Este valor se utiliza para ambos conjuntos, sin embargo, este margen puede aumentarse dependiendo de la ubicación de la cámara, tipo de emisión o precisión del modelo. Seleccionar los píxeles del cielo cercanos en lugar de todos los píxeles del cielo permite caracterizar el cielo justo detrás de la emisión, mejorando sobre los operadores que seleccionan cuadros delimitadores de forma manual.

La Figura 3.37 presenta una explicación visual. La emisión se muestra de color gris, el cielo de azul oscuro y el edificio de rojo. El contorno de color blanco es el conjunto de píxeles de cielo ignorados, y conjunto de píxeles de color azul claro son los píxeles de cielo utilizados para el cálculo de la estimación de opacidad.



Figura 3.37: Obtención de las regiones de interés (Cielo, emisión, y edificio)

### 3.3.2.2. Algoritmos de estimación

Para ejecutar la solución propuesta se necesita de un método de estimación de la opacidad de las emisiones que haga uso de las diferentes regiones. Los métodos existentes de estimación de la opacidad de las emisiones son los siguientes.

- *Ringelmann*: el método de la carta de Ringelmann es una adaptación aproximada de la primera metodología basada en una comparación visual con una carta. En este caso, en lugar de una evaluación visual, se compara la intensidad de la emisión convertida a escala de grises utilizando la recomendación BT.709 [112] con la intensidad de cada referencia de la carta (ver Figura 3.38a).

El valor del píxel 255 corresponde al 100% de la intensidad luminosa, mientras que 0 corresponde al 0%. La Figura 3.38b muestra la equivalencia de rangos de los valores de intensidad con la escala de Ringelmann. Una gran desventaja de este método es que depende de la imagen, por lo que dos emisiones con el mismo nivel de opacidad pueden diferir dependiendo de la iluminación de la escena y la calibración de la cámara. Esto se debe a que no se utiliza ninguna referencia para ajustar el cálculo de opacidad.

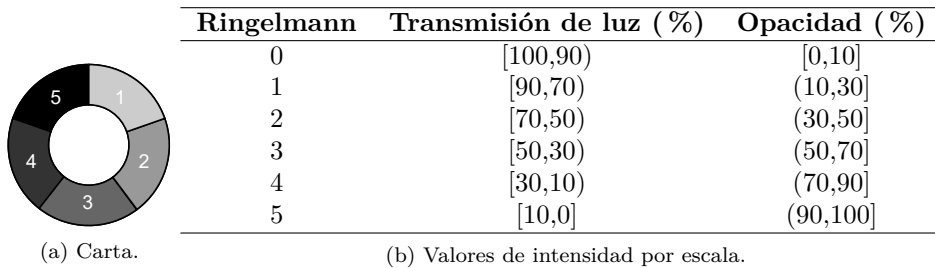


Figura 3.38: Escala de Ringelmann.

- *DOM (modelo de transmisión): Digital optical method* (DOM) determina la opacidad de la emisión mediante la Ecuación 3.3.  $N_p$  se refiere al valor de luminancia de los píxeles de emisión. Para calcular este valor se toma el promedio de todos los píxeles de emisión.  $N$  es el valor de los píxeles de fondo. Para calcular este valor se toma el promedio de todos los píxeles de cielo.  $K$  es un coeficiente que depende de la transmitancia de las partículas y del entorno. Según [32] y la patente DOM [62], se recomienda un valor de  $K$  de 0,16 para emisiones negras y 1,4 para emisiones blancas. Este método está diseñado para fondos uniformes. El conjunto de datos utilizado

contiene emisiones oscuras, por lo que se establece un valor de 0,16 para todas las imágenes.

$$O = \frac{1 - \frac{N_p}{N}}{1 - K} \quad (3.3)$$

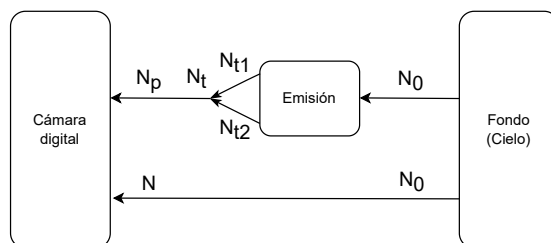


Figura 3.39: Modelo físico de *DOM* (*modelo de transmisión*).

- *Prakasa et al.*: el método descrito por *Prakasa et al.* [100] está diseñado para caracterizar las emisiones de chimeneas. Primero divide la imagen en varias filas que contienen la emisión, de modo que cada fila utilizará valores de referencia contenidos en la misma fila. Esto se hace porque se asume que cuanto más alta sea la elevación desde la chimenea, menor será la densidad de emisión. Por esta razón, se promedian los valores de intensidad del cielo para una fila dada. La Figura 3.40 muestra un ejemplo de separación en filas.

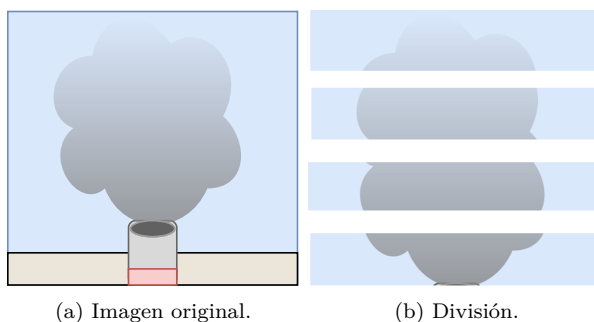


Figura 3.40: División de la emisión en regiones horizontales.

La opacidad se determina comparando la diferencia de color de la emisión

con el fondo del cielo y la máxima diferencia de color. El valor máximo se obtiene asumiendo que el color de la emisión es negro puro. Por lo tanto, todos los valores en el canal RGB serán cero para representar una intensidad completamente negra. Este valor máximo se puede considerar como una referencia para cuantificar el nivel de opacidad.

La opacidad se calcula para cada píxel de la región individualmente. Las intensidades de los píxeles vecinos no influyen en el cálculo de la opacidad de un punto observado. Las Ecuaciones 3.4, 3.5, y 3.6 se utilizan para obtener el valor de opacidad.

$I_p$  es el valor o intensidad de la banda RGB para los píxeles que pertenecen a la emisión o emisión.  $I_s$  son los píxeles que pertenecen al ajuste lineal que representa el cielo para las bandas RGB.

$$d_{RGB} = \sqrt{(I_p - I_s)R^2 + (I_p - I_s)G^2 + (I_p - I_s)B^2} \quad (3.4)$$

$$d_{Ref} = \sqrt{I_{p,R}^2 + I_{p,G}^2 + I_{p,B}^2} \quad (3.5)$$

$$O = \frac{d_{RGB}}{d_{Ref}} * 100 \quad (3.6)$$

La Ecuación 3.6 divide el valor de la diferencia entre la intensidad del píxel de emisión ( $I_p$ ) y la intensidad del cielo en el eje vertical de la región ( $I_s$ ) por la intensidad de la emisión misma ( $I_p$ ). Esto parece ser incorrecto basándose en la ecuación física para calcular la opacidad ( $Opacity = 1 - I/I_0$ , donde  $I$  es el flujo de luz que traspasa la emisión y  $I_0$  es el flujo de luz incidente sin pasar a través de la emisión), ya que debería dividirse por la intensidad del cielo ( $I_s$ ). La diferencia de intensidades se normaliza para cada píxel por separado. Esta normalización causa que la opacidad resultante de los valores de  $I_p$  cercanos a  $I_s$  sea minimizada, mientras que al usar valores con una diferencia mayor entre  $I_p$  e  $I_s$ , el valor de opacidad se maximiza. Esto ocurre porque después de dividir por  $I_p$ , cuando éste es un valor de intensidad muy bajo, el valor por el cual se divide también es muy bajo, y puede ocurrir una división por cero si el valor de  $I_p$  es completamente negro. De manera similar, cuando los dos valores están cerca, su diferencia, es decir, el numerador, tendrá un valor más bajo. Además, en este caso, el denominador tendrá un valor más alto, lo que resultará en un valor de opacidad mucho menor.

La principal desventaja de este método es que al representar visualmente una máscara con valores de opacidad, se pueden ver bordes al dividir la

imagen en regiones. Este método no está diseñado para crear máscaras, sino para obtener representaciones de la opacidad a lo largo del eje vertical. Sin embargo, se generan máscaras para proporcionar una comparación con el resto de los métodos.

- *Yuen et al.*: el método descrito por *Yuen et al.* [136,137] divide las referencias del método DOM (modelo de transmisión) según la intensidad de su fondo. El DOM (modelo de transmisión) utiliza solo una referencia para el cielo y otra para la emisión. El método de *Yuen et al.* utiliza dos referencias para el cielo y dos para la emisión. Una de cada par de referencias está en una zona con mayor intensidad y la otra en una zona con menor intensidad. Este método requiere la recomendación BT.709 para convertir los valores RGB en valores de intensidad en escala de grises.

Este método se ejecuta utilizando la Ecuación 3.7 donde  $O$  es la opacidad de la emisión.  $E_{wp}$  es la cantidad de exposición causada por el fondo brillante con emisión.  $E_w$  es la cantidad de exposición causada por el fondo brillante sin emisión.  $E_{bp}$  es la cantidad de exposición causada por el fondo oscuro con emisión y  $E_b$  es la cantidad de exposición causada por el fondo oscuro sin emisión. Se puede entender visualmente, como se muestra en la Figura 3.41a.

$$O = 1 - \frac{\frac{E_{wp}}{E_w} - \frac{E_{bp}}{E_b}}{1 - \frac{E_b}{E_w}} = 1 - \frac{E_{wp} - E_{bp}}{E_w - E_b} \quad (3.7)$$

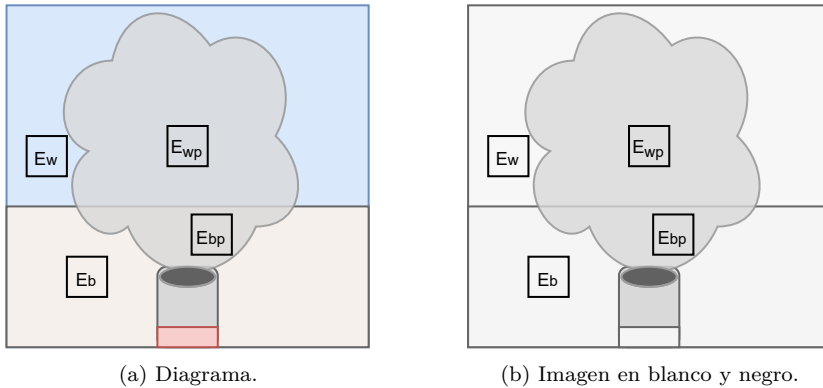


Figura 3.41: Puntos de referencia para el método de *Yuen et al.*

- *DOCS*: con fines de evaluación, en este trabajo se ha implementado el método *Digital Opacity Compliance System* (DOCS) siguiendo las indicaciones de su patente [96]. Sin embargo, esta versión puede no coincidir exactamente con la implementación oficial porque se han dejado sin explicar ciertos aspectos del desarrollo. En [76, 77, 96] se explica que se utiliza RGB para todo el proceso, aunque se puede utilizar HSV.

El primer paso del método DOCS es suavizar la imagen para eliminar o reducir los artefactos visuales [96]. El segundo paso consiste en aplicar el algoritmo PCA a la región de interés de la imagen RGB suavizada. Esto reduce la dimensionalidad de los datos de tres canales a un canal. Los valores obtenidos después de utilizar el método PCA representan la variabilidad de la intensidad del color de las bandas R, G y B. Para este proceso solo se utiliza la primera componente de la PCA. Por simplicidad se llamará PCA1.

El tercer paso consiste en calcular la opacidad de la emisión. Los valores negativos de la representación PCA1 están directamente relacionados con la opacidad a través de una relación lineal. Para evitar que las imágenes generen resultados completamente diferentes, esta relación lineal se establece en todo el conjunto de datos utilizando un valor mínimo y máximo. El valor mínimo se obtiene tomando el percentil 5 de todos los mínimos de cada imagen en el conjunto de datos. Los mínimos de cada imagen se obtienen utilizando el percentil 1 para eliminar los valores atípicos. El máximo se obtiene de la misma manera pero utilizando el percentil 95 para los máximos y el percentil 99 para eliminar los valores atípicos.

- *Transmitancia*: el método de la transmitancia se basa en la fórmula de transmisión [105] mostrada en la Ecuación 3.8. Como se muestra en la Figura 3.42,  $I$  es la intensidad de la luz que traspassa la emisión y  $I_0$  es la intensidad de la luz emitida por el cielo. Es común que los algoritmos sigan una variación de esta ecuación. Por ejemplo, el método de transmisión es similar al método DOM (modelo de transmisión), pero en este caso sin el valor  $K$ . El propósito del método de Transmisión es seguir la ecuación del modelo físico de la manera más simple posible.

Los valores de intensidad se obtienen utilizando la recomendación BT.709 para transformar la imagen RGB en valores de intensidad en formato de escala de grises.  $I_0$  se calcula como la mediana de la región del cielo. La mediana es más robusta que el promedio contra valores atípicos causados por superficies reflectantes o posibles artefactos en la imagen.  $I$  es el valor de un píxel particular de la emisión.

$$O = 1 - \frac{I}{I_0} \quad (3.8)$$

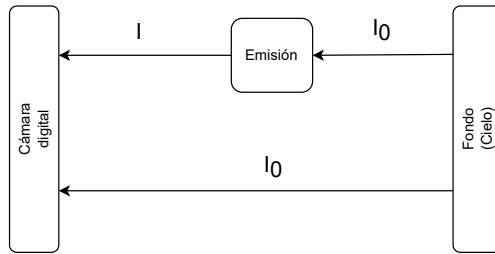


Figura 3.42: Modelo físico del método de Transmisión.

- SBPB - Propuesto: el método *Sky and Building Percentiles in the Blue channel* (SBPB) no existe en la literatura y ha sido desarrollado específicamente para la realización de este estudio. En los métodos descritos anteriormente, la emisión debe ser negra. Con este método, la emisión puede ser de otros colores como marrón o amarillento, excepto azul (ver Figura 3.43a). Por lo tanto, debido a que el cielo/nubes generalmente tienen un tono azul, se asume que cuanto más azul es el valor de un píxel, menos opaco es. Por el contrario, cuanto menor sea su valor de azul, más opaco será. Después de algunas pruebas con las bandas RGB por separado y en escala de grises usando el algoritmo BT.709, todas presentaron problemas cuando la luz solar brillaba directamente sobre el edificio. Sin embargo, la banda B es resistente a esto. Por este motivo se decidió usar solamente la banda B de las imágenes RGB para el cálculo de opacidad.

Este método establece que la región segmentada como edificio tiene una opacidad del 100 % y que la región segmentada como cielo tiene una opacidad del 0 %. Para utilizar este método, el edificio utilizado como referencia no puede ser azul (ver Figura 3.43b).

En base a esta suposición, se obtiene el valor del percentil 75 de la región de los edificios y el valor del percentil 25 de la región del cielo. Estos percentiles proporcionan valores de referencia robustos para el edificio y el cielo, evitando valores atípicos que pueden ser causados por artefactos en la imagen o superficies reflectantes [126]. Estos valores de intensidad luego se pueden utilizar para ajustar el cálculo de opacidad a las condiciones de luz de la imagen. Esto permite el uso de imágenes obtenidas de cámaras dinámicamente autoajustables o mal calibradas.

Dado que el edificio tiene una opacidad del 100 %, los valores de emisión iguales o inferiores representan una opacidad del 100 %. Aquellos iguales o superiores a la región del cielo tienen una opacidad del 0 %. Los píxeles restantes, es decir, aquellos entre los dos valores límite, se normalizan entre



0 % y 100 % utilizando estos valores límite. De esta manera los píxeles de emisión tienen valores entre 0 % y 100 % (ver Figura 3.43c).

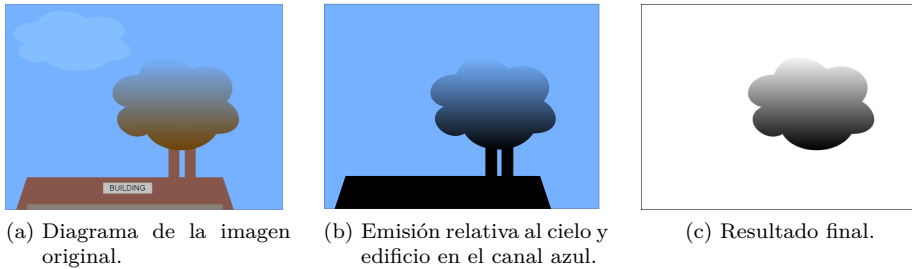


Figura 3.43: Extracción de la emisión utilizando la banda azul.

### 3.3.2.3. Resultados y discusiones

Esta sección presenta los ejemplos más relevantes para la comparación de los diferentes algoritmos de estimación de opacidad para la solución propuesta. Las Figuras 3.44, 3.45, 3.46, y 3.47 muestran la imagen original con la emisión fugitiva y los resultados de la estimación de opacidad de los diferentes métodos de forma visual, donde el negro es la máxima opacidad y el blanco la mínima. Las imágenes muestran los valores en el rango  $[0, 1]$  para cada píxel.

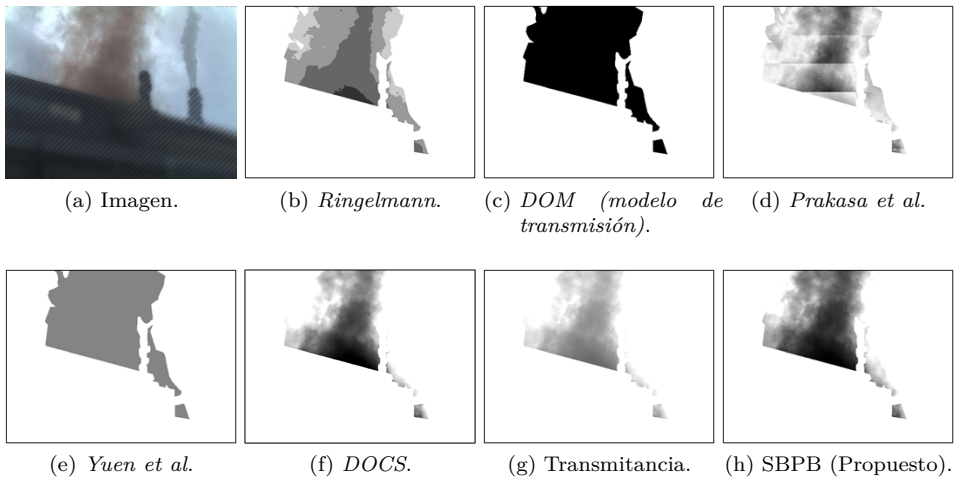


Figura 3.44: Emisiones de alta opacidad (Plant1).

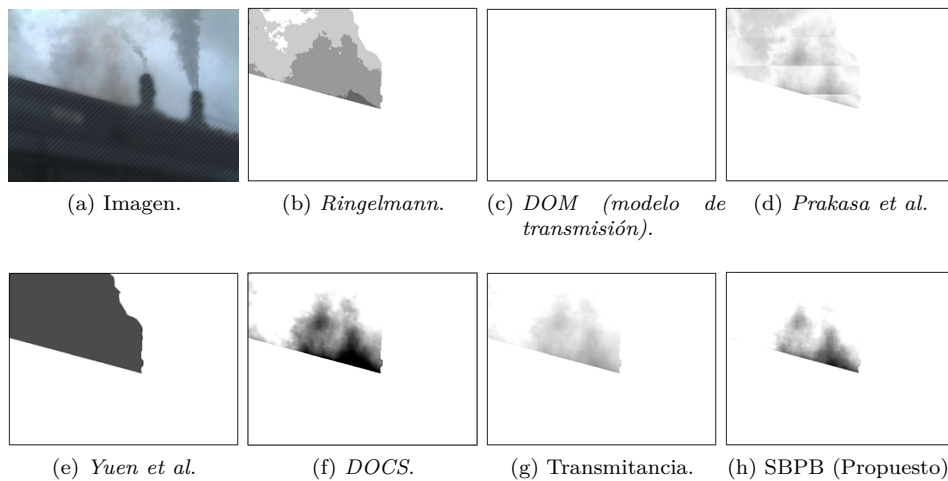


Figura 3.45: Emisiones de baja opacidad (Plant1)

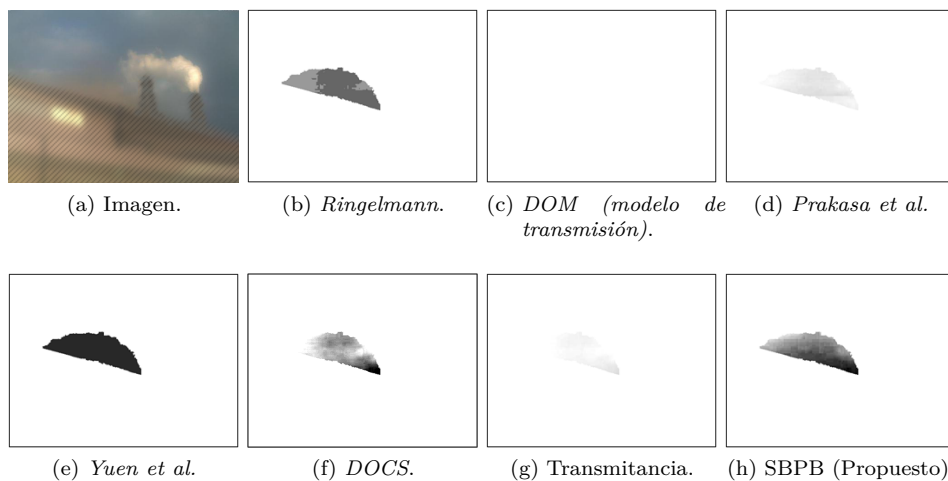


Figura 3.46: Emisiones con alta luminosidad (Plant1)

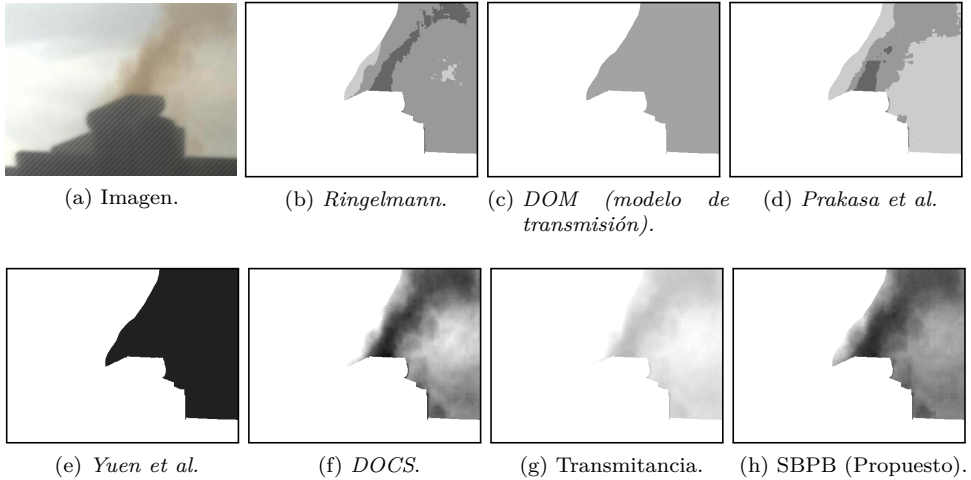


Figura 3.47: Emisiones de alta opacidad (Plant2).

La Figura 3.44 ilustra el comportamiento de los métodos en una situación de opacidad elevada. El método propuesto SBPB es el único que clasifica correctamente la opacidad de la parte central izquierda de la emisión. Los métodos *Prakasa et al.*, *DOCS* y *Transmitancia* dan a esa parte de la emisión un nivel de opacidad excesivamente bajo.

En situaciones de baja opacidad (ver Figura 3.45), se observa una mayor diferencia entre los métodos. Esto se debe a la mayor complejidad de la imagen debido a un menor contraste y mayor confusión con las nubes de fondo. El método *Ringelmann* valora los píxeles de la región por su intensidad sin tener en cuenta el contexto de la imagen, lo que hace que en estos casos se estime una opacidad más alta de lo esperado. El método *DOM (modelo de transmisión)* no es capaz de caracterizar la opacidad de la emisión. El método de *Yuen et al.* sobrestima la opacidad en áreas de transición entre la emisión y el cielo. Los métodos *DOCS* y *Transmitancia* mejoran significativamente las áreas de transición de la región de emisión, sin embargo, *DOCS* sobrestima la opacidad de la emisión. Finalmente, el método propuesto SBPB ajusta aún más la región de emisión y proporciona valores de opacidad más cercanos a los esperados.

En la Figura 3.46 se presenta una emisión en la que el sol está detrás de la cámara. Esto supone una situación difícil, especialmente para los modelos para los métodos basados en modelos físicos que presuponen que el sol se encuentra detrás de la emisión. Los métodos *DOM (modelo de transmisión)*, *Prakasa et al.* y *Transmitancia* no son capaces de determinar correctamente la opacidad de la emisión. En todos estos métodos, la emisión se caracteriza en un nivel de

opacidad muy bajo. Esto se debe a que esta imagen tiene una alta intensidad luminosa, lo que hace que el edificio tenga una alta intensidad. Por esta razón, el autoajuste de la cámara hace que el cielo se oscurezca para que el cielo sea aún más oscuro que el edificio. Sin embargo, el método propuesto SBPB es capaz de caracterizar correctamente la opacidad de la emisión. Esto se debe a que, aunque la intensidad luminosa es incorrecta debido a la autoajuste de la cámara, el cielo sigue teniendo un tono azul más alto que el edificio. En este caso, el método *Ringelmann* no se ve afectado por el cielo y el edificio, por lo que obtiene sus resultados habituales. Como el método *DOCS* se basa en la varianza de la propia imagen, tiene menos problemas para caracterizar la emisión. Finalmente, el método de *Yuen et al.* valora la emisión completa al mismo tiempo, pero es capaz de caracterizar esta emisión de una manera más razonable, lo que puede deberse al uso de múltiples referencias de edificios y cielo.

La Figura 3.47 ilustra el comportamiento de los métodos en una situación de opacidad elevada para Plant2. Aquí se puede ver que se comporta de manera similar al resto de las imágenes del primer conjunto de datos Plant1. La principal diferencia es que en este caso particular, el método *DOCS* parece obtener resultados similares al método propuesto SBPB. Estos dos métodos son los que mejor estiman la opacidad de la emisión.

Los métodos que obtienen un valor para toda la emisión, como es el caso del *DOM (modelo de transmisión)* y *Yuen et al.*, se ven notablemente afectados cuando la emisión tiene partes muy diferenciadas. Es decir, si la mitad de la emisión tiene una baja opacidad y la otra mitad tiene una alta opacidad, el resultado se ve afectado al obtener un valor promedio. Si un operador selecciona un cuadro delimitador, el resultado dependerá completamente de dónde se coloque el cuadro. Por esta razón, la selección automática propuesta en este trabajo proporciona resultados más confiables a pesar del peso de área de los diferentes niveles de opacidad. Esto es mucho más importante para aquellos métodos en los que la opacidad se calcula para cada píxel.

Si el edificio tiene una intensidad de luz más alta que el cielo, los métodos que utilizan el cielo y el edificio como referencias pueden comportarse de manera errónea, como es el caso del *DOM (modelo de transmisión)* y los métodos de *Yuen et al.* Esto suele deberse a situaciones de luminancia que hacen que el autoajuste de la cámara se vuelva muy agresivo. En estos casos, el método SBPB propuesto logra resultados satisfactorios porque el cielo todavía tiene una intensidad luminosa más alta que el edificio en la banda B. Por lo tanto, el método propuesto es mucho más robusto en situaciones en las que la iluminación no es perfecta.

Además de los ejemplos visuales, las Tablas 3.40 y 3.41 presentan los resultados de cada método para Plant1 y Plant2. Las imágenes de opacidad de los diferentes algoritmos de estimación de opacidad se procesan mediante un algoritmo de

clasificación para obtener su nivel de opacidad categórico que se comparará con el objetivo generado por los operadores humanos. Este algoritmo de clasificación descarta los valores de opacidad inferiores al 5 % porque son los más ruidosos [96], y calcula el percentil 80 de los píxeles de emisión restantes para calcular un valor de opacidad único para toda la emisión. Esto es necesario porque la percepción visual humana del brillo no es lineal [78]. Este valor único de opacidad se utiliza para determinar su clase (Baja, Media o Alta) mediante el uso de umbrales. Estos umbrales se calculan por separado para cada método con el fin de maximizar la separación entre clases. Para calcularlos, se utiliza el punto medio entre los valores medianos de las clases adyacentes. En otras palabras, el umbral entre la clase Baja y la clase Media se calcula como la suma de la mediana de la clase Baja con la mitad de la diferencia con respecto a la clase Media. El umbral entre la clase Media y la clase Alta se calcula de la misma manera.

Tabla 3.40:  $F_1$ -Score de los métodos para Plant1.

Método	Clase (Opacidad)			Promedio
	Baja	Media	Alta	
<i>Ringelmann</i>	0.254	0.402	0.322	0.326
<i>DOM (modelo de transmisión)</i>	0.663	0.336	0.415	0.471
<i>Prakasa et al.</i>	<b>0.804</b>	0.071	0.208	0.361
<i>Yuen et al.</i>	0.391	0.000	0.068	0.236
<i>DOCS</i>	0.586	0.019	0.125	0.243
Transmitancia	0.747	0.413	0.478	0.546
<b>SBPB (Proposed)</b>	0.775	<b>0.456</b>	<b>0.540</b>	<b>0.590</b>

Tabla 3.41:  $F_1$ -Score de los métodos para Plant2.

Método	Clase (Opacidad)			Promedio
	Baja	Media	Alta	
<i>Ringelmann</i>	0.609	0.000	0.185	0.264
<i>DOM (modelo de transmisión)</i>	0.769	0.333	0.361	0.488
<i>Prakasa et al.</i>	<b>0.792</b>	0.224	0.000	0.339
<i>Yuen et al.</i>	0.554	0.000	0.107	0.230
<i>DOCS</i>	0.592	0.053	0.128	0.258
Transmitancia	0.720	0.274	0.309	0.434
<b>SBPB (Proposed)</b>	0.753	<b>0.342</b>	<b>0.407</b>	<b>0.500</b>

Teniendo en cuenta los resultados presentados en la Tabla 3.40, se observa que el método propuesto SBPB supera al resto de métodos. Este método tiene un promedio de  $F_1$ -Score aproximadamente 5% superior al segundo mejor método. Las clases Media y Alta del método SBPB tienen un  $F_1$ -Score mucho más alto que el resto, sin embargo, el método *Prakasa et al.* supera al método SBPB en la clase de baja opacidad. Los métodos *DOM (modelo de transmisión)* y *Transmitancia* también producen un alto  $F_1$ -Score en baja opacidad. Los métodos *Prakasa et al.*, *Yuen et al.* y *DOCS* tienen dificultades para distinguir entre opacidad media y alta. La Tabla 3.41 muestra los resultados completos del conjunto de datos Plant2. Aquí se puede ver que los resultados obtenidos son similares a los de Plant1, manteniendo las mismas conclusiones. En este conjunto de datos en particular, el método *Ringelmann* también tiene dificultades con los niveles de opacidad media y alta.

# Capítulo 4

## Conclusiones

Tras una exhaustiva revisión del estado del arte de las tecnologías de análisis de imagen multiespectral y clasificación de regiones mediante aprendizaje profundo, se han identificado un conjunto de tareas de monitorización medioambiental con un gran potencial de mejora. La presente Tesis Doctoral ha desarrollado y validado una metodología basada en técnicas de análisis de imagen multiespectral para mejorar los procesos de monitorización ambiental, lo que ha permitido automatizar sistemas existentes y realizar tareas que antes eran inviables debido a su coste económico y necesidad de personal. Los resultados obtenidos han demostrado que esta metodología puede contribuir a una monitorización constante sin la intervención de operadores, lo que mejora la eficiencia y precisión de los sistemas. Además, la aplicación de la segmentación semántica a estas técnicas de análisis de imagen multiespectral ha permitido obtener mejoras sustanciales en comparación con las técnicas existentes, lo que resalta el potencial de esta tecnología para mejorar la monitorización ambiental.

En el estudio realizado, se ha analizado cómo diferentes ámbitos medioambientales pueden beneficiarse de la automatización de procesos basados en imágenes multiespectrales. Por ejemplo, en el caso de la agricultura de precisión, se puede mejorar la precisión y automatización del reconocimiento de cultivos y la detección del abono, lo que contribuye a la prevención de la contaminación del suelo y, por consiguiente, del medio ambiente por parte del sector de la ganadería. Este enfoque puede ser replicado en otras muchas áreas, como la detección de plantaciones, incendios o vertederos ilegales, así como el control de especies invasoras. Este campo está comenzando a ganar interés por parte de las empresas.

En el caso de la detección de defectos subsuperficiales, se acelera el proceso de inspección de piezas en la industria automotriz, aeroespacial y de manufactura en general. De esta manera, se puede reducir el tiempo de inspección, aumentar la eficiencia y minimizar el riesgo de errores humanos, evitando posibles desastres medioambientales y humanitarios. Además, el mantenimiento constante puede ayudar a reducir la cantidad de desechos generados.

En el caso de la detección de emisiones, se pueden aplicar técnicas para estimar la cantidad de gases contaminantes emitidos por fábricas y vehículos, lo

que puede ayudar a las empresas a cumplir con las regulaciones ambientales y reducir su huella de carbono, así como a mejorar la seguridad de los trabajadores evitando intoxicaciones. La detección de emisiones también puede ser útil para la vigilancia de incendios forestales y la detección de fugas de gas en tuberías y redes de distribución.

Uno de los mayores desafíos para la aplicación de técnicas de análisis de imágenes multiespectrales en la mejora de procesos de monitorización ambiental es la creación de conjuntos de datos de gran tamaño y con una buena cobertura de los casos posibles. El tamaño y la variabilidad del conjunto de datos pueden influir en la selección del método de clasificación. Por ejemplo, para conjuntos de datos más pequeños y con poca variabilidad, es mejor utilizar métodos tradicionales de aprendizaje automático, mientras que para conjuntos de datos más grandes y variados, se recomienda el uso de métodos de aprendizaje profundo, como la segmentación semántica. A mayor cantidad y variedad de datos, los resultados serán mejores y más robustos.

El control de procesos mediante imágenes multiespectrales y la combinación de la segmentación semántica y las técnicas de análisis de imagen multiespectral se presentan como una solución efectiva para mejorar la sostenibilidad de los procesos tecnológicos y garantizar un mejor cuidado del medio ambiente. Los resultados obtenidos representan un importante avance en el conocimiento de estas áreas y permiten establecer estrategias efectivas para la protección y conservación del medio ambiente, a través de la optimización de los procesos y la reducción del impacto ambiental en el sector industrial. En definitiva, la aplicación de técnicas de aprendizaje profundo y análisis de imágenes multiespectrales pueden ser una herramienta clave para impulsar la sostenibilidad y el cuidado del medio ambiente en distintos ámbitos, desde la agricultura de precisión hasta la detección de emisiones contaminantes. Es importante seguir investigando y mejorando estas técnicas para lograr una gestión más eficiente y sostenible de nuestros recursos naturales y minimizar el impacto de la actividad humana en el medio ambiente.

## 4.1. Trabajo futuro

El enfoque desarrollado en esta tesis puede ser reforzado mediante el estudio de pruebas en vivo a largo plazo, así como un entrenamiento continuo aprovechando las imágenes en tiempo real y la supervisión de los operadores. Además, existe la posibilidad de utilizar múltiples fuentes al mismo tiempo para aumentar la confianza de los modelos, ya sea utilizando varios satélites o varias cámaras donde las imágenes alimentan uno o varios modelos. Esto aumentaría la cantidad de información recibida al obtener varios ángulos o resoluciones diferentes, así como la disminución del ruido causado por baja visibilidad.



En los resultados de esta investigación se puede observar que aún existe margen de mejora en las tareas presentadas así como para la creación de tareas más complejas. Por este motivo, el incesante desarrollo del aprendizaje profundo, mediante nuevas y mejores arquitecturas, presenta una línea de investigación que permite la mejora y evolución de los métodos aquí presentados. Por ejemplo, la investigación de novedosos modelos diseñados para secuencias temporales como los *Transformers* podría presentar mejoras en los resultados de cualquier proceso de monitorización de zonas constantes. Por otro lado, la mejora de la disponibilidad de nuevas herramientas de medición o sensores, ya sean satélites o la ubicuidad de dispositivos como los drones, con una mayor resolución espacial y espectral, así como una mayor frecuencia, convertirían este enfoque en una tarea indispensable para un desarrollo sostenible en las tareas desarrolladas en esta investigación.



# Capítulo 5

## Contribuciones

En este capítulo se muestran las contribuciones realizadas durante esta tesis doctoral.

### 5.1. Publicaciones: JCR (Journal Citation Reports)

#### 5.1.1. Evaluation of Semantic Segmentation Methods for Land Use with Spectral Imaging Using Sentinel-2 and PNOA Imagery

- Pedrayes, O. D., Lema, D. G., García, D. F., Usamentiaga, R., & Alonso, Á. (2021). *Evaluation of semantic segmentation methods for land use with spectral imaging using Sentinel-2 and PNOA imagery*. *Remote Sensing*, 13(12), 2292.
- DOI: [10.3390/rs13122292](https://doi.org/10.3390/rs13122292)
- El índice de impacto de la revista *Remote Sensing* en 2021 fue 5.349 (Q1, 85.40 %) y el índice de impacto a 5 años, 5.786.
- Citas: 10



## Article

# Evaluation of Semantic Segmentation Methods for Land Use with Spectral Imaging Using Sentinel-2 and PNOA Imagery

Oscar D. Pedrayes <sup>1,\*</sup> , Darío G. Lema <sup>1</sup> , Daniel F. García <sup>1</sup> , Rubén Usamentiaga <sup>1</sup> and Ángela Alonso <sup>2</sup>

<sup>1</sup> Department of Computer Science and Engineering, Campus de Viesques, University of Oviedo, 33204 Gijón, Spain; UO243567@uniovi.es (D.G.L.); dfgarcia@uniovi.es (D.F.G.); rusamentiaga@uniovi.es (R.U.)

<sup>2</sup> Department of Spatial Data, Seresco S.A., Matemático Pedrayes 23, 33005 Oviedo, Spain; angela.alonso@seresco.es

\* Correspondence: UO251056@uniovi.es

**Abstract:** Land use classification using aerial imagery can be complex. Characteristics such as ground sampling distance, resolution, number of bands and the information these bands convey are the keys to its accuracy. Random Forest is the most widely used approach but better and more modern alternatives do exist. In this paper, state-of-the-art methods are evaluated, consisting of semantic segmentation networks such as UNet and DeepLabV3+. In addition, two datasets based on aircraft and satellite imagery are generated as a new state of the art to test land use classification. These datasets, called UOPNOA and UOS2, are publicly available. In this work, the performance of these networks and the two datasets generated are evaluated. This paper demonstrates that ground sampling distance is the most important factor in obtaining good semantic segmentation results, but a suitable number of bands can be as important. This proves that both aircraft and satellite imagery can produce good results, although for different reasons. Finally, cost performance for an inference prototype is evaluated, comparing various Microsoft Azure architectures. The evaluation concludes that using a GPU is unnecessarily costly for deployment. A GPU need only be used for training.

**Keywords:** sentinel; pnoa; sigpac; unet; deeplab; multi-spectral; aerial; agriculture; convolutional neural network; semantic segmentation



**Citation:** Pedrayes, O.D.; Lema, D.G.; García, D.F.; Usamentiaga, R.; Alonso, Á. Evaluation of Semantic Segmentation Methods for Land Use with Spectral Imaging Using Sentinel-2 and PNOA Imagery.

*Remote Sens.* **2021**, *13*, 2292. <https://doi.org/10.3390/rs13122292>

Academic Editor: Saeid Homayouni

Received: 30 April 2021

Accepted: 8 June 2021

Published: 11 June 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Currently, when there is a need for land use classification from aerial imagery, the standard procedure consists in a manual process carried out by professionals in that specific field. This process is time consuming and costly. Automation both reduces costs and makes the process much faster. New applications that require a response time of milliseconds can be developed, for example, for the classification of each frame of a video, and thus they are able to process constantly changing regions.

Being able to automate location tasks in aerial imagery, such as land use or crop classification, opens up the possibility of exploring new services, such as crop monitoring, which may be of interest to companies. Many satellites not only update their data every few days but also allow free access. This could lead to significant advances in agricultural applications and other fields.

In other works [1–5], whenever a study of a land use classification dataset is performed, the datasets usually remain private, so the results proposed are not reproducible. There does not appear to be a common dataset specifically for testing accuracy and performance of land use classification, in the way COCO [6] and PASCAL VOC [7] are used for image classification. For this reason, two datasets are offered in this work for public use. Both use data from SIGPAC [8], a free Spanish government database for agricultural land identification, to generate the ground truths required for training and testing the models. The first dataset, called UOS2, obtains its source imagery from the Sentinel-2 satellite [9].

The second dataset, called UOPNOA, obtains its source imagery from the aircraft imagery from the National Plan for Aerial Orthophotography [10], also known as PNOA, a free governmental database of aerial orthophotography of Spain. These datasets are released with this work for public use and can be downloaded with the following DOI (last accessed on 10 June 2021): [doi.org/10.5281/zenodo.4648002](https://doi.org/10.5281/zenodo.4648002).

Both UOPNOA and UOS2 identify eleven land use types, which means eleven different target classes. They are constructed from SIGPAC data of the same region, the northern part of the Iberian Peninsula plateau in Spain. The two datasets are very different as one of them has been obtained from aircraft imagery and the other from satellite imagery. However, as both of them are generated from the same region, a realistic comparison can be made. Obviously, the aircraft images are taken much closer to the ground, which means that objects and textures have a much better resolution. In contrast, the dataset taken from satellite imagery has more bands, offering up to thirteen different bands.

Automation of land use classification using satellite imagery is not a new concept [3,11–13]. The previous methods used are mainly based on Random Forest [14] and Support Vector Machine [4,15]. There are over 18,000 articles in 2020 mentioning Random Forest and land use classification and 11,000 articles mentioning SVM and land use classification in 2020. However, there are better alternatives with simple convolutional neural networks [1,16]. These networks are viable as long as the dataset used to train the models has sufficient data and variability, which may be an issue in areas where the data is more restricted. Complex convolutional neural networks using specifically semantic segmentation networks perform even better if a correct dataset is provided, as has been demonstrated with other satellite imagery tasks such as land cover classification [5].

This has led to the study of convolutional neural networks and the development of semantic segmentation for aerial imagery [17]. There are scientific publications comparing these methods with more recent semantic segmentation architectures in satellite imagery [5,18]. However, there are few publications that study the latest advances in semantic segmentation in the specific application of land use classification tasks and their particular behavior using this type of dataset. Those that do exist identify only a few main target classes as abstract as “vegetation” [19] or highly differentiated classes such as “Bare Rock”, “Beaches”, “Water bodies”, etc. For this reason, UNet [20] and DeepLabV3+ [21], two semantic segmentation architectures, are tested with the UOPNOA and UOS2 datasets. In this way, the influence of the different characteristics of the images in semantic segmentation networks is studied.

The aim of this paper is to evaluate the difference between satellite imagery and aircraft imagery for the specific task of land use classification using semantic segmentation methods, the generation of a dataset of each type from the same region, and the evaluation of a service for a realistic use case. Parameters such as ground sampling distance, multi-spectral bands, and complexity of target classes are taken into account.

The accuracy of land use classification depends on the ground sampling distance of the images and the number of bands. The UOS2 uses images with up to thirteen bands, including different infrared bands. These images have bands with a ground sampling distance or GSD of 10, 20, or 60 m/pixel, meaning that there are 10, 20, or 60 m of ground between the center of one pixel and the center of the next. UOPNOA only includes RGB bands, but its GSD is of 0.25 m/pixel. To isolate the effects of the number of bands and to determine the minimum number needed to ensure good results, several versions of the UOS2 dataset are used: two versions of three bands (RGB and non-RGB), six bands, ten bands and thirteen bands. Since UOPNOA has a much smaller GSD than UOS2, results from the two datasets are compared to determine its influence. The greater the GSD, the more difficult it is to differentiate one class from another, even to the human eye. Finally, as having too many classes can be detrimental to the accuracy of the models, tests are conducted with different numbers of classes. Similar classes are merged to see how far two similar classes can be separated without confusion in classification.

This study is performed with semantic segmentation networks, testing well-known [22–26] convolutional neural networks such as UNet [20] or DeepLabV3+ [21]. This means that not only are different imagery sources tested but also different methods. In addition, the complexity of the target classes is studied. Different numbers of classes, generic and specific classes, and the use of an all-purpose class are also evaluated in this study.

After training and testing, the feasibility of deployment is evaluated for the most relevant implementations. A microservices architecture following modern standards is then deployed to create an inference prototype on different infrastructures. Thus, the performance and cost of different deployments can be studied. The deployments compared are local, infrastructure as a service (IaaS), container as a service (CaaS), and function as a service (FaaS). The hardware and its components are also compared, including the use of a GPU versus a CPU, to determine whether the added cost of a GPU is justified.

In summary, the contributions offered by this work are the following:












- Different sources of imagery such as aircraft and satellite imagery are compared.
- Two new public datasets to evaluate land use classification with different characteristics and complexities are released for public use.
- Differences in GSD, number of bands of the images, resolution, and other characteristics are evaluated for semantic segmentation for land use classification.
- Semantic segmentation methods are evaluated in aerial and satellite imagery.
- The complexity of classes and target classes is studied.
- Cost performance of different deployments for a possible implementation of a semantic segmentation model for land use classification in aerial imagery is analyzed.

## 2. Materials and Methods

### 2.1. Datasets



#### 2.1.1. Target Classes

“Sistema de Información Geográfica de Parcelas Agrícolas” or SIGPAC, is a free database provided by the Spanish government which allows one to geographically identify the plots declared by the farmers. It has up to thirty different classes defined, but according to the requisites of the potential users of this kind of product, not all of these classes are relevant for agriculture. From SIGPAC the following classes are extracted, each of them associated to a color. Both UOPNOA and UOS2 datasets will use the same classes and color palette.

-  UN—Unproductive
-  PA—Pastureland
-  SH—Shrub grass
-  FO—Forest
-  BU—Buildings and urban area
-  AR—Arable Land
-  GR—Grass with trees
-  RO—Roads
-  WA—Water
-  FR—Fruits and nuts
-  VI—Vineyard

Pixels that do not correspond with any of these classes will either be assigned to an all-purpose class called “OT—Other” or be unused depending on the experiment. This class is of no relevance to the study and its only purpose is to provide a realistic prediction that permits pixels that do not belong to any other target class.

To test a lower number of more generic classes, new classes made from combinations of the previous one are created. This allows for simpler experiments to study class complexity.

-  PASHGR—All pastures
-  BURO—All infrastructures

- ■ ARVI—Arable lands and vineyards

### 2.1.2. Uopnoa Dataset

PNOA is a database of digital aerial orthophotographs that are accessible for free in Enhanced Compressed Wavelet, a format capable of compressing enormous images and storing their georeference. Georeference is important to establish the ground truth for a dataset as it makes it possible to merge data from different sources such as SIGPAC. This format is necessary because of the resolution of the images, with hundreds of thousands of pixels in width and height, but it can be converted to GEOTIFF if required. Each orthophotograph has a GSD of 0.25 m/pixel and covers a region equivalent to one MTN50 page, the National Topographic Map of Spain. One of the main advantages of PNOA, apart from its great GSD, is that the images have no clouds or other defects. However, it has a great disadvantage for periodic crop monitoring as this data is only updated once a year. Information about the bands is described in Table 1.

**Table 1.** Bands from PNOA.

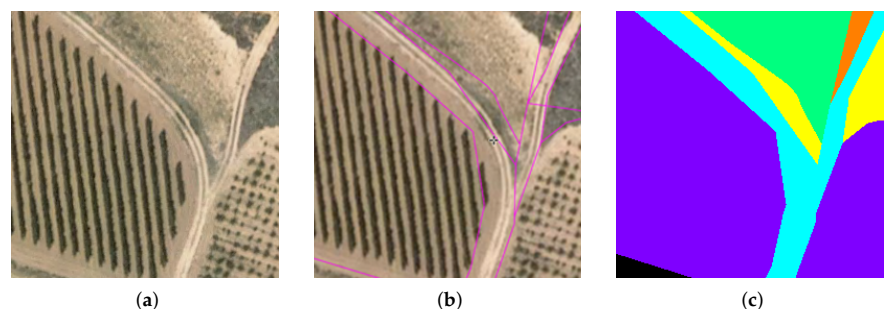
Bands	GSD (m/pixel)	Bits
B1 Red	0.25	8
B2 Green	0.25	8
B3 Blue	0.25	8
B4 NIR	0.25	8

UOPNOA consists of 33,699 images of  $256 \times 256$  pixels. These images are cropped out of PNOA images that cover a region equivalent to an MTN50 page. In order to keep the georeferencing data in the images, functions from the GDAL library are used to crop the images.

Images from PNOA are downloaded using CNIG (Centro Nacional de Información Geográfica) [27], as this is the official procedure of the government of Spain to download PNOA imagery. However, the band “B4 NIR” cannot be downloaded.

The annotation process was carried out using the coordinates of the SIGPAC plots to colour the regions of the images to generate masks. Since the images are georeferenced, this process can be done automatically.

Figure 1 shows one of these cropped images. To check that the ground truth has been correctly generated, it is compared to official data from the SIGPAC visor. A screenshot of this visor is presented next to the ground truth mask. The SIGPAC visor separates each plot, or individually fenced piece of land, but as only the type of land use is needed, there is no need to separate plots with the same type.

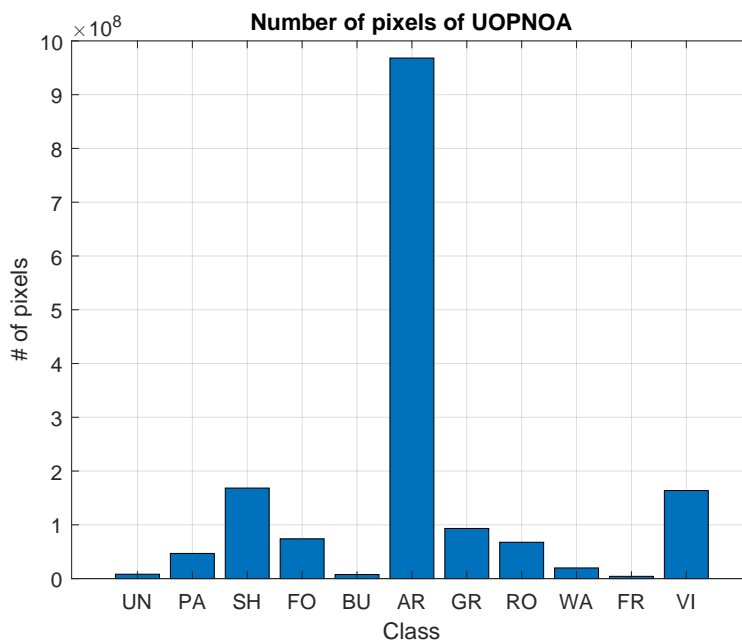


**Figure 1.** Cropped image and ground truth checking for UOPNOA. (a) cropped image of  $256 \times 256$  pixels from a PNOA image, (b) SIGPAC visor of the same region to check the ground truth mask, (c) ground truth mask.

The number of plots from SIGPAC used to make the ground truth for each class is presented in Table 2 along with the number of pixels of the UOPNOA dataset. A visual representation of the pixels is shown in Figure 2. A large difference between the AR class and the rest of the classes can be observed. This kind of land use is usually bigger than the rest of the target classes, and in the region selected this class is very common.

**Table 2.** Number of SIGPAC plots and pixels for each class used in UOPNOA.

Class	Plots	# of Pixels
UN	410	8,095,351
PA	1883	46,738,269
SH	3467	168,342,415
FO	472	73,980,816
BU	125	7,562,696
AR	4935	968,106,602
GR	220	93,208,409
RO	943	67,552,872
WA	340	19,776,364
FR	93	4,150,529
VI	1759	163,774,218



**Figure 2.** Number of pixels for each class of the dataset UOPNOA.

### 2.1.3. Uos2 Dataset

Sentinel-2 is a mission based on a constellation of two identical satellites (Sentinel-2A and Sentinel-2B) on the same orbit, which offers its data for free in GEOTIFF. This format is capable of storing not only the image with as many bands as necessary but also all the data needed to georeference every pixel in the image. These georeferences are used to establish the ground truth from SIGPAC data. Sentinel-2 offers thirteen different bands, with different wavelengths, bandwidths, and GSDs. Each satellite has a 10-day period around the equator and 4–6 days at midlatitudes. However, since they are half a revolution apart, the time required to update the images is reduced by half.



Sentinel-2 has been widely used in works that study semantic segmentation of satellite imagery, as there are more than 11,000 papers mentioning this satellite in 2020 and more than 38,000 in total. One of its main advantages is its period of five days around the equator and 2–3 days at midlatitudes.

In Table 3, the data for each band is shown. The columns “Wavelength” and “Bandwidth” have information from Sentinel-2A/Sentinel-2B in this same order.

**Table 3.** Bands from Sentinel-2A and Sentinel-2B.

Band	Wavelength (nm)	Bandwidth (nm)	GSD (m/pixel)
B1 Coastal aerosol	442.7/442.2	21/21	60
B2 Blue	492.4/492.1	66/66	10
B3 Green	559.8/559.0	36/36	10
B4 Red	664.6/664.9	31/31	10
B5 VRE	704.1/703.8	15/16	20
B6 VRE	740.5/739.1	15/15	20
B7 VRE	782.8/779.7	20/20	20
B8 NIR	832.8/832.9	106/106	10
B8A Narrow NIR	864.7/864.0	21/22	20
B9 Water vapour	945.1/943.2	20/21	60
B10 SWIR Cirrus	1374.0/1376.9	31/30	60
B11 WIR	1614.0/1610.4	91/94	20
B12 SWIR	2202.0/2185.7	175/185	20

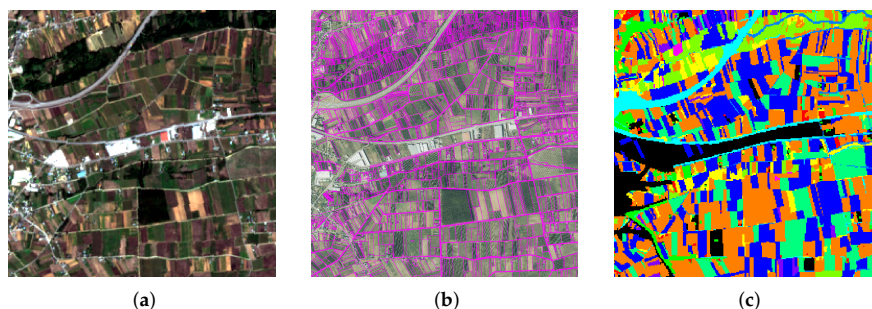
UOS2 consists of 1958 images of  $256 \times 256$  pixels, all of them taken in July 2020. These images are cropped from Sentinel-2 images. To facilitate the generation of ground truth masks, images from Sentinel-2 that cover entire regions from MTN50 pages and its respective data from SIGPAC are obtained. Combining these to make the ground truth is straightforward. Then, these images are cropped to images of  $256 \times 256$  pixels.

Images from Sentinel-2 are downloaded using SentinelHub [28], as this is the easiest way to download a specified region from a concrete date using a simple script.

To obtain valid images from Sentinel-2, images that do not contain clouds or any other defect are searched for manually. When a region of interest includes anything that could compromise its quality, another date on which the image has no defects is found. In this manner, the number of images to check goes from 1958 small images to 39 larger Sentinel-2 images, making it much easier to check manually.

The annotation process was carried out similarly to the UOPNOA dataset, using the coordinates of the SIGPAC plots to paint the mask accordingly.

Figure 3 shows an example of one these cropped images and its mask. In the same way as before, to check that the ground truth is correctly generated, it is compared with the official data from the SIGPAC visor.

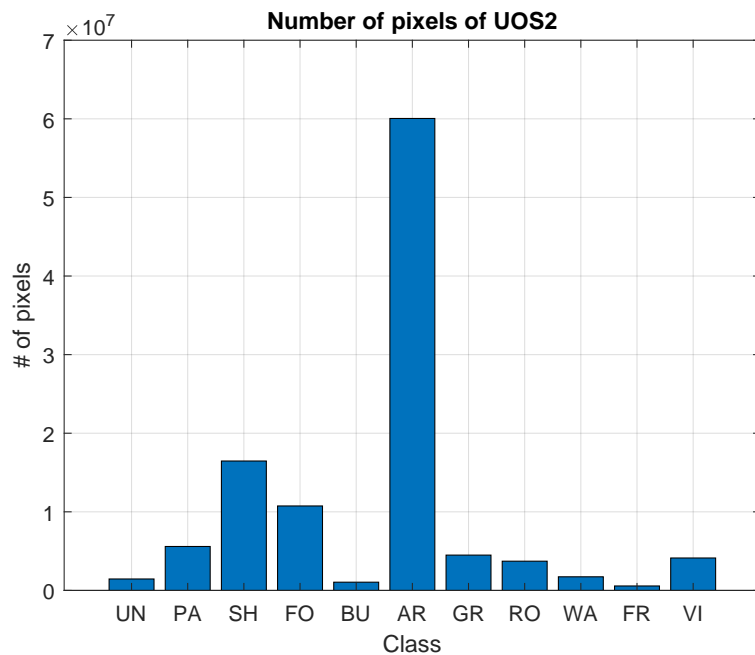


**Figure 3.** Cropped image and ground truth checking for UOS2. (a) cropped image of  $256 \times 256$  pixels from a Sentinel-2 image, (b) SIGPAC visor of the same region, (c) ground truth mask.

The number of plots from SIGPAC used to make the ground truth for each class is presented in Table 4 along with the number of pixels of the UOS2 dataset. A visual representation of the pixels is in Figure 4. Given that the UOS2 dataset covers the same region as the UOPNOA dataset, a large difference between the AR class and the rest of the classes is also observed.

**Table 4.** Number of SIGPAC plots used in UOPNOA.

Class	Plots	# of Pixels
UN	26,912	1,455,995
PA	152,359	5,591,140
SH	265,046	16,465,578
FO	48,977	10,746,146
BU	207,839	1,047,900
AR	772,427	60,046,203
GR	28,649	4,494,801
RO	73,250	3,719,742
WA	16,802	1,733,641
FR	6306	566,355
VI	87,071	4,126,803



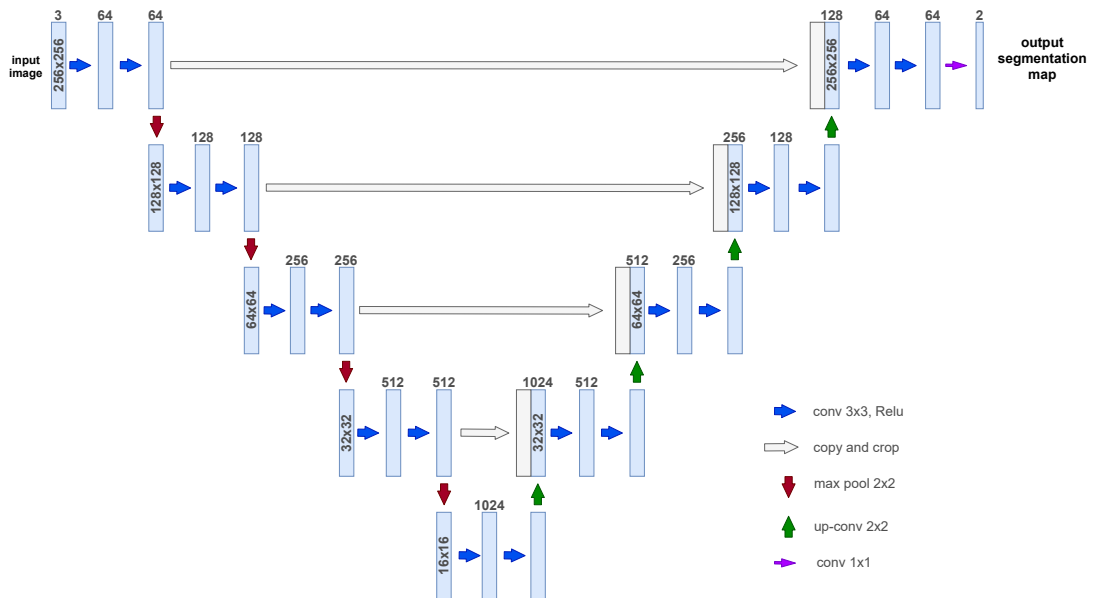
**Figure 4.** Number of pixels for each class of the dataset UOS2.

## 2.2. Analysis of the Evaluated Architectures

### 2.2.1. Unet

Originally developed for binary classification to segment cells in biomedical images, UNet is one of the most widely referenced networks in semantic segmentation, cited in over 25,000 papers. Its original motivation was to train and produce precise predictions with as few training images as possible. The name UNet comes from its symmetric encoder-decoder architecture, giving it a u-shaped architecture. As it has a relatively simple architecture, it offers a high degree of flexibility. Thanks to this flexibility, many networks based on its architecture have been developed. UNet was quickly modified to work with all kinds of images and numbers of classes and is currently the most widely used with satellite imagery.

Therefore, UNet will be used in this evaluation study. In Figure 5, an overview of UNet architecture can be seen.



**Figure 5.** UNet architecture. This diagram is based on the original UNet publication [20].

### 2.2.2. Deeplab

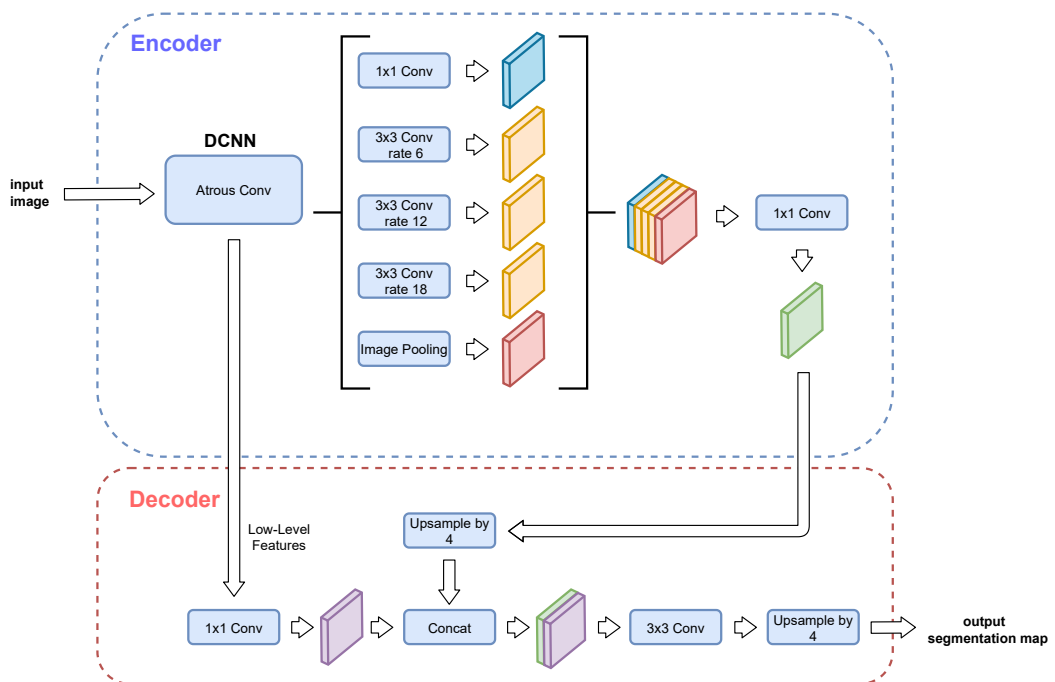
Developed by Google, this network is still under development, having had a number of versions until now. The first version [29] uses atrous convolutions to control the resolution at which feature responses are computed. The second version, DeepLabV2 [30], adds an Atrous Spatial Pyramid Pooling (also known as ASPP) module to make better predictions at different scales. The third version, DeepLabV3 [31], upgrades the ASPP module and also adds Batch Normalization to the architecture to make setting up the training easier, since a manual normalization is no longer needed. The fourth version, DeepLabV3+ [21], adds a decoder module to convert its architecture to encoder-decoder. Finally, there is also an auto machine learning version called AutoDeepLab [32], whose architecture is based on DeepLabV3+. AutoDeepLab is not evaluated in this work because its computational cost would make the time required to train it far too long. Therefore, because DeepLabV3+ is the most recent version and usually obtains better results than UNet, it has been selected for evaluation in this paper. This architecture is widely used since it has over 3600 citations on its paper. In Figure 6, an overview of DeepLabV3+ architecture can be seen.

### 2.3. Network and Training Parameters

In this section the following parameters used in the experiments to train the models and to modify the architecture of the networks are described:

- Input size: The resolution and number of channels of the input images.
- Classes: Number of classes to train.
- Backbone network: Classification network used as a part of the initial architecture of a more complex network, such as DeepLabV3+.
- Depth: The number of max pooling layers in the UNet architecture.
- Filters on first level: The number of filters on the first convolution of the UNet architecture. This value is multiplied by 2 on every depth level.

- Output stride: The division between the input image resolution and the final feature map. For example, if the input image has a resolution of  $256 \times 256$  and the final feature map  $32 \times 32$ , then the output stride is 8. It controls the separation between each step of the convolution. This is a configurable parameter in the Google DeepLabV3+ implementation.
- Padding: A filler that is added to each convolution so as not to reduce the resolution of the final feature map.
- Class balancing: The method used to prevent a biased training when the dataset is unbalanced.
- Solver: Algorithm that calculates the gradient when training the network.
- Epochs: Number of times the complete dataset is used in training.
- Fine-Tune Batch Normalization: A parameter that allows the DeepLabV3+ implementation to train the batch normalization layers instead of using the pretrained ones.
- BatchSize: Number of images used in each batch. Since the entire dataset cannot be stored in memory, the dataset is divided into batches.
- LearningRate: Parameter that controls how the network weights are adjusted with respect to the gradient.
- Gradient clipping: Limits the maximum value of the gradient to prevent the exploding gradient problem when training.
- L2 regularization: A technique to reduce the complexity of a model by penalizing the loss function. As a result, overfitting is reduced.
- Data augmentation: Generation of new data from the original data.
- Shuffle: The dataset is shuffled on every epoch.



**Figure 6.** DeepLabV3+ architecture. A backbone network such as ResNet-101 or Xception65 can be used as DCNN. This diagram is based on the original DeepLabV3+ publication [21].

#### 2.4. Performance Metrics

To evaluate accuracy of the trained models, a brief description of the metrics used [33] is presented below. These metrics are usually calculated per class and then averaged to obtain a global metric for the model. This is done to prevent unbalanced classes from affecting the global results.

- True positive (*TP*): number of pixels that are correctly classified.
- True negative (*TN*): number of pixels that are from other classes and are correctly classified as such.
- False positive (*FP*): number of pixels classified as the target class but belonging to other classes.
- False negative (*FN*): number of pixels that are classified as other classes but are from the target class.
- Producer accuracy (*PA*): Percentage of correctly predicted pixels of a given class. Producer accuracy is often called Recall (*R*).

$$PA = \frac{TP}{TP + FN} \quad (1)$$

- User accuracy (*UA*): A percentage that represents how many predictions are correct from the total number of predictions for that class. User accuracy is often called Precision (*P*).

$$UA = \frac{TP}{TP + FP} \quad (2)$$

- F-score ( $F1_1$ ): equivalent to the Dice Coefficient, is a metric that combines both Producer accuracy (Recall) and User accuracy (Precision) as a way to represent them as a single value, making comparisons between models easier.

$$F_1 = \frac{2 \times UA \times PA}{UA + PA} \quad (3)$$

- Overall accuracy (*OA*): A percentage that represents how many pixels are correctly classified from the total. Overall accuracy is often called Global accuracy (*GA*). This metric can be misleading if the classes are not balanced. For example, given two classes, if one of them represents 99% of the pixels in the dataset and the other one represents the remaining 1%, even if all of the pixels from the second class are classified wrongly as pixels from the first class, this metric will still obtain an *OA* of 99%.

$$OA = \frac{TP}{Total\ of\ pixels} \quad (4)$$

- Intersection-Over-Union (*IoU*): equivalent to the Jaccard Index, this metric measures the degree of similarity between the ground truth and prediction sets.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} = \frac{TP}{TP + FN + FP} \quad (5)$$

#### 2.5. Experimental Procedure

In order to make a comparison between methods, a basic experimentation with RF and SVM has been performed. This experimentation consists in using the values Red, Green, and Blue as features. For the RF and SVM experiments, a total of 12,000 pixels were used, 1000 pixels for each class. The entire dataset has not been used due to temporal constraints and because these models do not scale to the same level as a neural networks, so no improvement is observed above a certain amount of data.

The hyperparameters of all networks must be tuned to match the dataset used. In this work these hyperparameters are tuned manually in each network for the two datasets

separately. In the case of UNet, these hyperparameters are: learning rate, epochs, L2 regularization, depth levels, number of filters in the convolutional layers, type of solver, gradient clipping, and batch size. On the other hand, for DeepLabV3+, the hyperparameters used are: learning rate, epochs, L2 regularization, type of solver, usage of fine-tune batch norm, gradient clipping, batch size, and the backbone network used.

Every parameter is tested one by one until its best configuration is found. The effect of changing multiple parameters at the same time has not been tested. In this regard there is still leeway for further improvement. As this process is done manually, this methodology is the best option to obtain an optimal configuration.

There must be sufficient separation between the training and testing data to avoid repeating samples. To obtain more meaningful and realistic results, the testing must be done with data that the model has not seen before. To do this separation, the dataset is divided into four parts. In this way, not only can a separation between training and testing data be made but also a four-fold crosstesting.

For the selection of the optimal hyperparameters, instead of evaluating each class separately only the global User and Producer accuracy will be compared. For the best experiments, both Producer accuracy and User accuracy for each class will be considered and studied.

The visualization examples from some of the testing data give a better idea of how well the network performs. These images allow for a more in depth analysis and might give explanations for some of the results obtained. To distinguish when the “Other” class is being used, pixels identified with this class are coloured black. Similarly, when no class is associated with a pixel, these pixels are coloured white. Pixels not classified as belonging to any class are not taken into account when training or testing the network. This means that for the purposes of the evaluation, these pixels and their predictions are irrelevant.

The machines used for the experiments consist of a GPU NVIDIA RTX 2080 Ti and a I7-9700K CPU.

### 3. Results And Discussion

#### 3.1. Previous Methods

To make a fair comparison and see the performance of the proposed methods, they should be compared to the previous methods: Random Forest and Support Vector Machine. Both methods need much less data than a common neural network and take much longer to run. For this research work a reduced UOPNOA dataset with 1000 samples (pixels) per class was used, with a total of 12,000 samples. More detailed results can be found in Table 5. Random Forest achieved results of 0.074% overall accuracy with only four seconds of training. The Support Vector Machine achieved 0.073% overall accuracy in three and a half minutes of training. The time required to train these experiments does not scale linearly: when tested with only 6000 samples, 500 per class, SVM took only about thirty seconds. The overall accuracy did not improve, obtaining almost the same value of 0.07%. The testing of the two methods coincides with the same set of images that the rest of the UOPNOA experiments use.

**Table 5.** Global metrics for the experiments with RF and SVM on UOPNOA.

Experiment	OA	PA	UA	IoU	F <sub>1</sub>
RF	0.074	0.114	0.130	0.034	0.121
SVM	0.073	0.138	0.130	0.035	0.134

The features used for these experiments consist of the red, green, and blue values of each pixel. Proper experimentation with these methods must include feature engineering. This is not necessary with neural networks, as they generate their own features. This is a great advantage, although a large dataset is needed. When comparing Random Forest and

SVM with proper feature engineering with neural networks, results can still be worse as long as the dataset used is sufficiently large and variable [22,34,35].

### 3.2. Experimentation with Uopnoa

Experiments with DeepLabV3+ and UNet for the UOPNOA dataset and a discussion of their results are presented in this subsection.

#### 3.2.1. Deeplabv3+

Optimal network configuration for DeepLabV3+ with the dataset UOPNOA is presented in Table 6, and its best training parameters are shown in Table 7. The architecture of the network used for this experiment is the official Google architecture. It can be downloaded from its repository on GitHub along with pretrained models to reduce training time. After manually tuning the network to work with this dataset, the optimal hyperparameters found are listed in Table 6.

**Table 6.** Network parameters for DeepLabV3+ on UOPNOA.

Network Parameters	
Input size	256 × 256 × 3
Classes	11
Backbone network	Xception41
Output stride	16
Padding	Yes
Class balancing	Median frequency weighting

The input size from the evaluation for UNet was maintained for all the experiments in this work to compare the exact same dataset between implementations. The following backbone networks were tested: Xception65, Xception41, Xception71, MobileNetV2, MobileNetV3 Small, MobileNetV3 Large, and Resnet50. For each of them, the best learning rate, batch size, and number of epochs was established. The values tested for output stride are 8 and 16, the most commonly used. Padding is always added to equal the sizes of the outputs to the inputs, making tasks such as translating information to geojson easier. For class balancing, “No class weighting”, “inverse frequency weighting”, and “median frequency weighting” are compared.

**Table 7.** Training parameters for DeepLabV3+ on UOPNOA.

Training Parameters	
Solver	Adam
Epochs	60
Fine Tune Batch Normalization	No
Batch size	12
Learning rate	0.00005
Gradient clipping	No
L2 regularization	0.0004
Shuffle	Yes
Data augmenting	No

For the training parameters in Table 7, the network solvers Adam [36] and SGDM [37] were both evaluated. The number of epochs for training was calculated by overshooting their values and then observing where the overfitting starts or where the loss seems to plateau. The use of Fine-Tune Batch Normalization was tested with multiple configurations to ensure that it does not depend on parameters such as the backbone network, learning rate, batch size, or epochs. Batch size is the parameter with the greatest effect. With small batch sizes, the use of Fine-Tune Batch Normalization makes the results noticeably better, but with bigger batch sizes, in the order of twelve images, results are slightly worse.

DeepLabV3+ is a very complex architecture but a batch of twelve images can be used for all the backbone networks evaluated for eleven gigabytes of VRAM. The more images used in a single batch, the better the results. Batch sizes in the range of four to sixty-four were tested for those backbone networks that allow it. Constant learning rates from 0.001 to 0.00005 were studied using decrements that consist of halving or dividing by ten its value for each experiment, depending on its loss during training. The training process did not encounter the exploding gradient problem so gradient clipping was not needed.

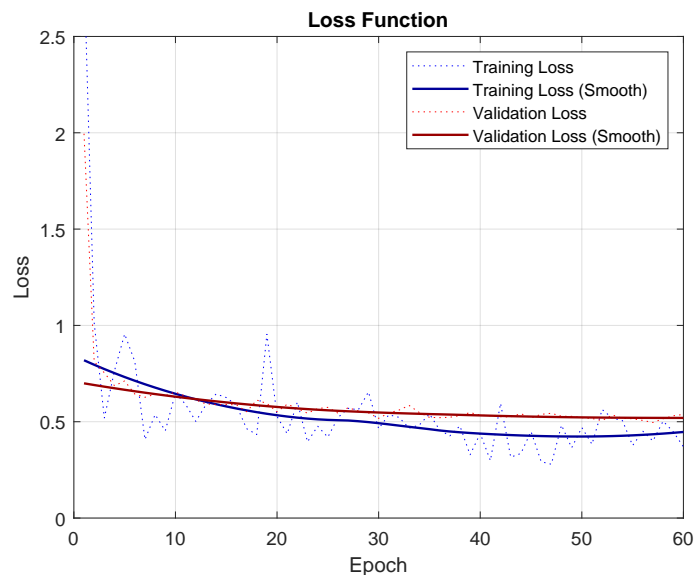
Two experiments were carried out to compare the use of the class “Other”. The first experiment, called “Base”, had no “Other” class, whereas the second experiment, called “All-Purpose”, did. Both experiments used the same data and configurations.

Global metrics for both experiments can be seen in Table 8. In the “Base” experiment, a considerable increase in PA and UA was observed, with approximately 10% and 8% better results for PA and UA respectively. Moreover, the OA improves by up to 15% when the “Other” class is not used. This indicates that an all-purpose class does not help in this kind of dataset and network.

**Table 8.** Global metrics for the experiment with DeepLabV3+ on UOPNOA.

Experiment	OA	PA	UA	IoU	F <sub>1</sub>
Base	0.898	0.781	0.758	0.637	0.769
All-Purpose	0.750	0.678	0.678	0.524	0.678

To prove that the tuning was done correctly, progress from the loss function for training and validation can be seen in Figure 7. Training loss starts to plateau and validation loss stabilizes.



**Figure 7.** Loss function for the “Base” experiment with DeepLabV3+ for PNOA.

Looking at the results of both experiments by class in Table 9, it is obvious that in most cases an all-purpose class only introduces confusion. This may be because an all-purpose class like “Other” is too generic. This class obtains the best results of all of the classes, but since it is of no interest, this is not the desired behavior.



**Table 9.** Class metrics for the experiment with DeepLabV3+ on UOPNOA.

Class	Experiment Base			Experiment All-Purpose		
	PA	UA	IoU	PA	UA	IoU
UN	0.56	0.66	0.43	0.65	0.63	0.47
PA	0.79	0.78	0.65	0.55	0.48	0.27
SH	0.58	0.53	0.38	0.38	0.39	0.30
FO	0.84	0.84	0.73	0.63	0.68	0.49
BU	0.85	0.86	0.76	0.78	0.75	0.62
AR	0.94	0.97	0.92	0.76	0.82	0.65
GR	0.75	0.89	0.69	0.81	0.87	0.72
RO	0.84	0.60	0.54	0.82	0.75	0.65
WA	0.77	0.47	0.41	0.75	0.52	0.44
FR	0.66	0.73	0.53	0.46	0.56	0.34
VI	0.96	0.95	0.92	0.62	0.77	0.53
OT	-	-	-	0.86	0.86	0.76

From Figures 8 and 9 the two experiments can be compared graphically. At first glance, both seem to make good predictions, especially when all the pixels of an image are of the same class. However, upon closer inspection the “Base” experiment is slightly more stable and closer to a human technician’s classifications. This confirms the figures presented in Table 9.

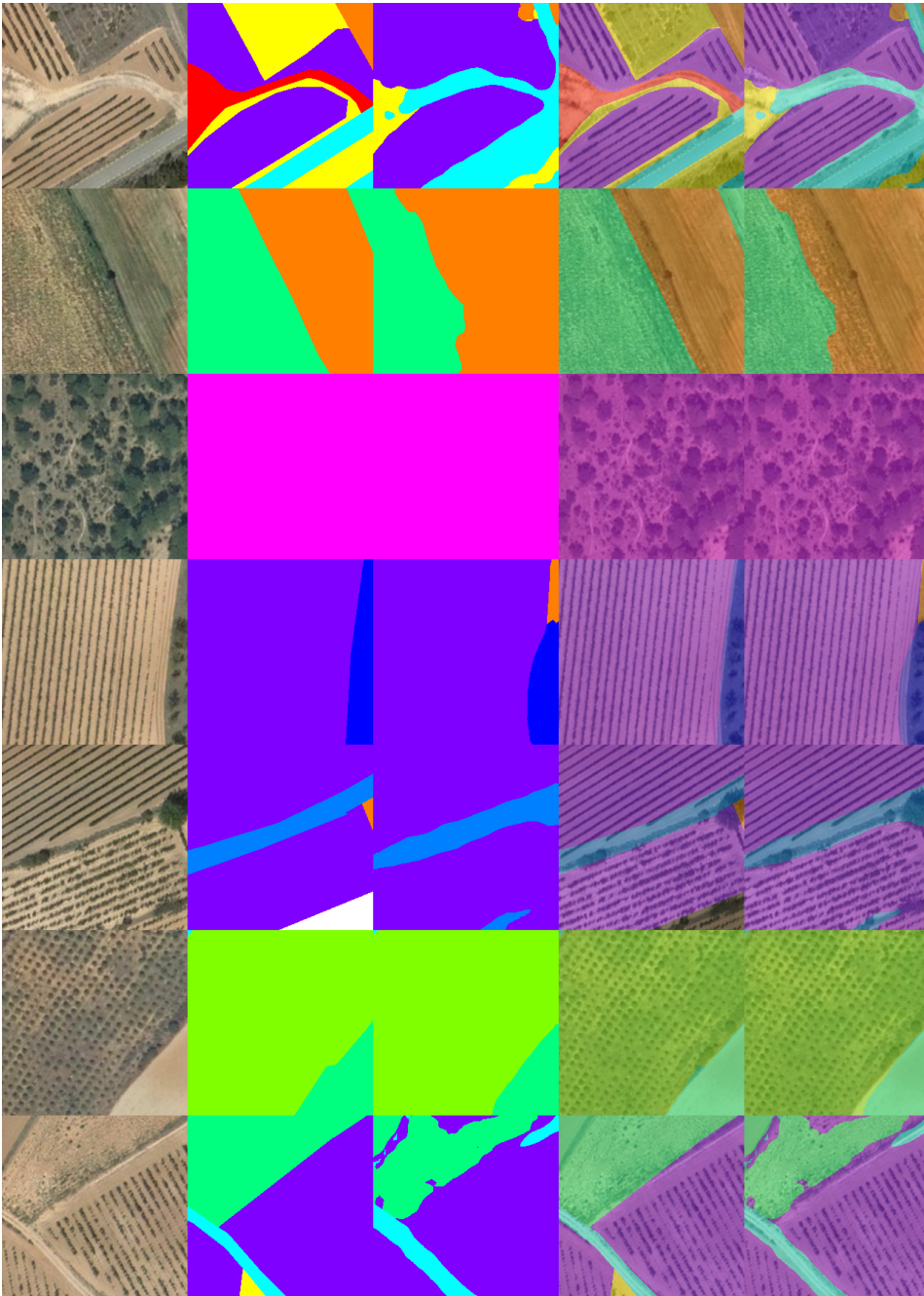
### 3.2.2. Unet

The optimal network configuration for UNet with the dataset UOPNOA is presented in Table 10, along with its best training parameters in Table 11. This architecture is exactly the same as the official publication [20], except that it has been adapted to take RGB images instead of grayscale images. As this problem is a multi-class classification rather than a binary classification, categorical cross entropy (CCE) is used to calculate loss values. Different numbers of depth levels, from one to five, and filters on the first level, from 16 to 128, were tested. The optimal configurations coincide with the original architecture.

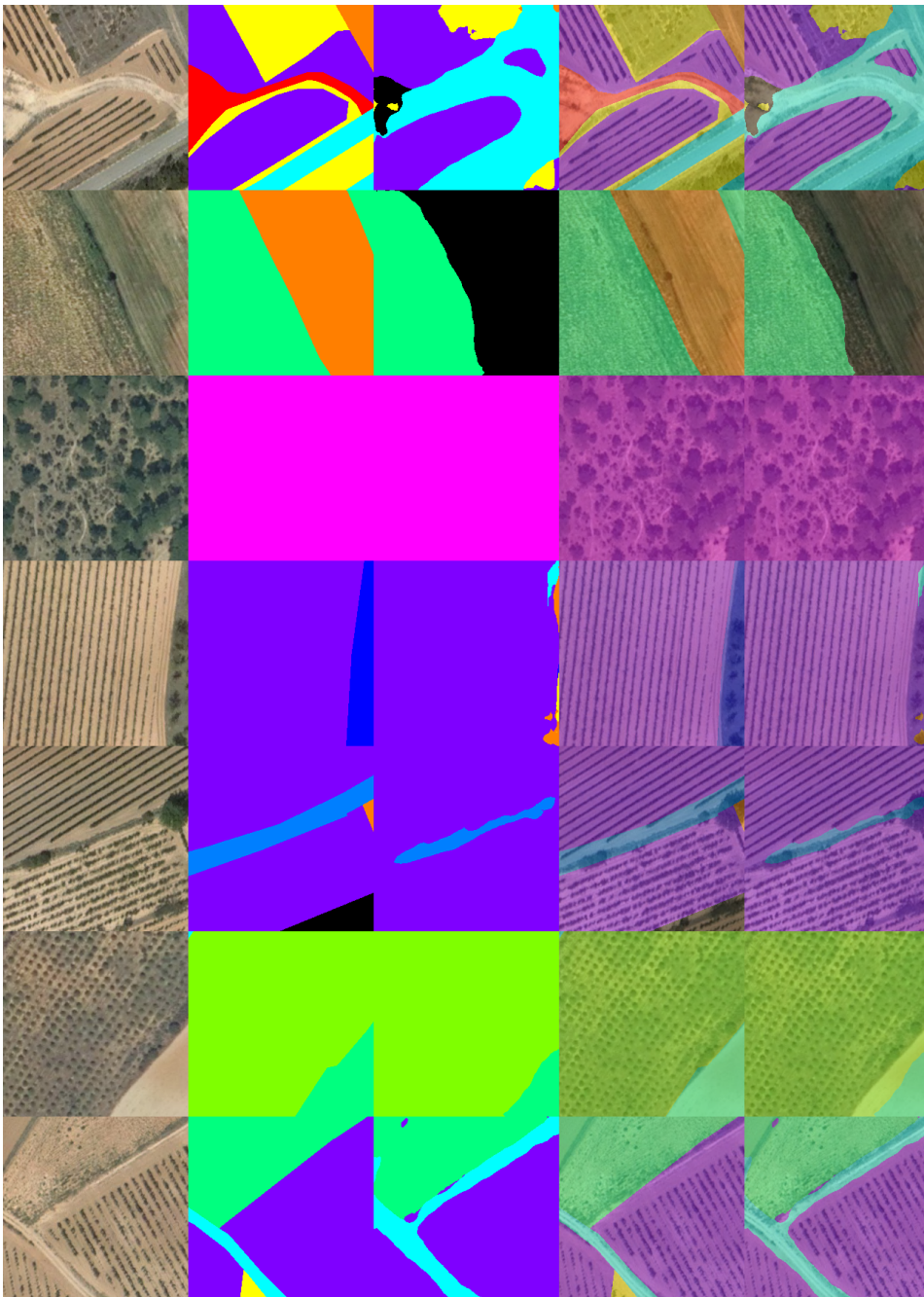
**Table 10.** Network parameters for UNet on UOPNOA.

Network Parameters	
Input size	$256 \times 256 \times 3$
Classes	11
Depth	4
Filters on first level	64
Padding	Yes
Class balancing	Median frequency weighting

After testing multiple resolutions from  $1024 \times 1024$  pixels to  $256 \times 256$  pixels for the input size of the images, it was found that the network performs far better when using a resolution of  $256 \times 256$  pixels. To maintain cohesion and facilitate comparisons, this resolution is used throughout this work. As for class balancing, “median frequency weighting” always obtains the best results.



**Figure 8.** Visualization of the predicted results for DeepLabV3+ evaluated in UOPNOA (Experiment Base). (1st col.) Original images, (2nd col.) ground truth masks, (3rd col.) predictions, (4th col.) overlapping of original images with ground truth masks, (5th col.) overlapping of original images with predictions.



**Figure 9.** Visualization of the predicted results for DeepLabV3+ evaluated in UOPNOA with “Other” class (Experiment All-Purpose). (1st col.) Original images, (2nd col.) ground truth masks, (3rd col.) predictions, (4th col.) overlapping of original images with ground truth masks, (5th col.) overlapping of original images with predictions.

**Table 11.** Training parameters for UNet on UOPNOA.

Training Parameters	
Solver	Adam
Epochs	20
Batch size	16
Learning rate	0.0001
Gradient clipping	1.0
L2 regularization	0.0001
Shuffle	Yes
Data augmenting	No

Tuning the training parameters from Table 7 is a process equivalent to that described for the DeepLabV3+ network, except there is no backbone network or Fine-Tune Batch Normalization. Gradient clipping is not relevant as its value in this case is too high, so it does not affect training. Furthermore, it is not needed since there is no exploding gradient problem. Finally, L2 regularization was tested and a value of 0.0001 works best, reducing overfitting and improving results. Higher values were tested but caused accuracy to drop. Similarly, when using lower values, overfitting starts earlier, reducing accuracy in testing.

To compare UNet with DeepLabV3+, both experiments done with UOPNOA are recreated with the optimal configuration for UNet. This also allows for a study on the behavior of the all-purpose class “Other” in different networks.

Global metrics for both experiments can be seen in Table 12. In the experiment “Base”, an increase in PA and UA is observed, around 1% and 8% better results respectively. The OA improves up to 19% when the “Other” class is not used. Like the experiments with DeepLabV3+, this indicates that an all-purpose class does not help in this kind of dataset and network. When comparing these results to DeepLabV3+, a difference of 16% and 30% in PA and UA is observed for the best experiments.

**Table 12.** Global metrics for the experiment with UNet on UOPNOA.

Experiment	OA	PA	UA	IoU	F <sub>1</sub>
Base	0.830	0.618	0.473	0.473	0.536
All-Purpose	0.641	0.607	0.391	0.391	0.476

To prove that the tuning has been done correctly, progress from the loss function for training and validation can be seen in Figure 10. Validation loss stabilizes but training loss keeps lowering; if more epochs are executed, overfitting of the model would start to occur.

In Table 13, the results from each class are presented. Like the experiments conducted on DeepLabV3+, there is a considerable increase in accuracy when the all-purpose class is not used. On the other hand, in this case the class “Other” has the worst results. Like the DeepLabV3+ experiments, the rest of the classes drop noticeably when PA and UA are considered.

Figures 11 and 12 show the predicted results from the experiments. These predictions, when compared to those of DeepLabV3+, are quite noisy, and there is far more confusion between classes. In this case the experiment “Base” is similar to the experiment “All-Purpose” for these particular examples, even though there is a difference of 8% in UA.

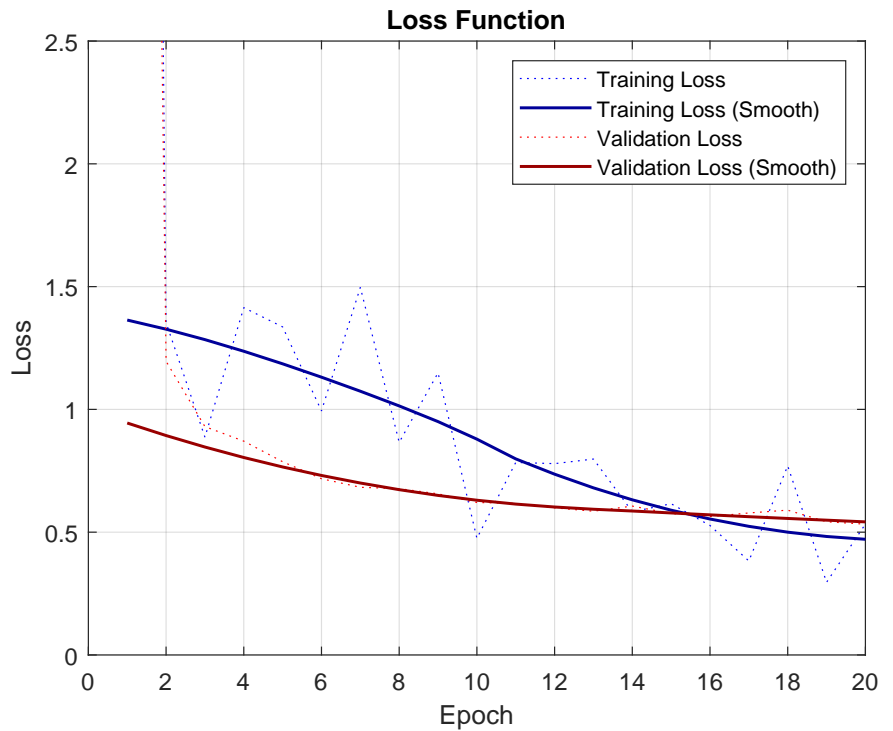
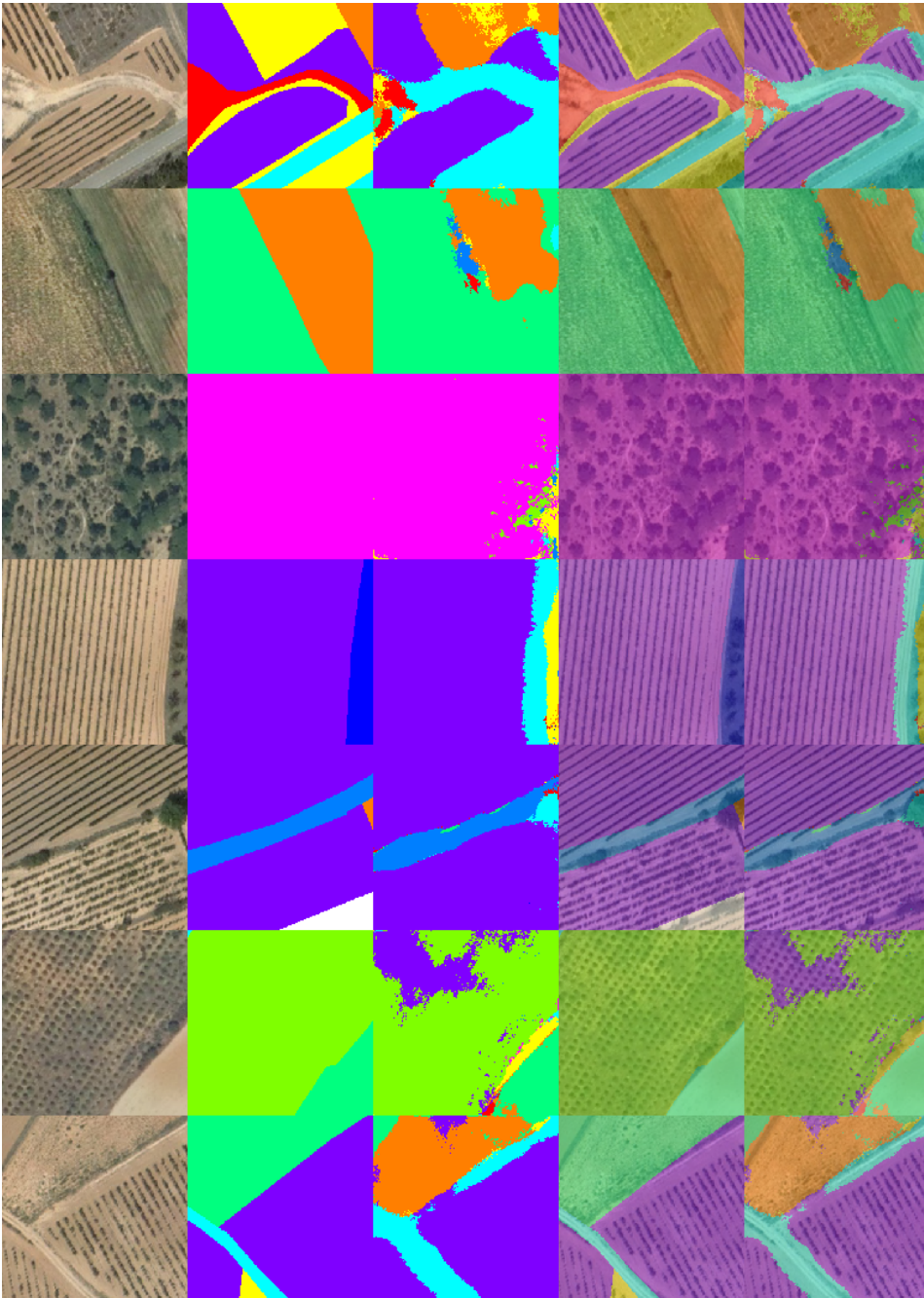


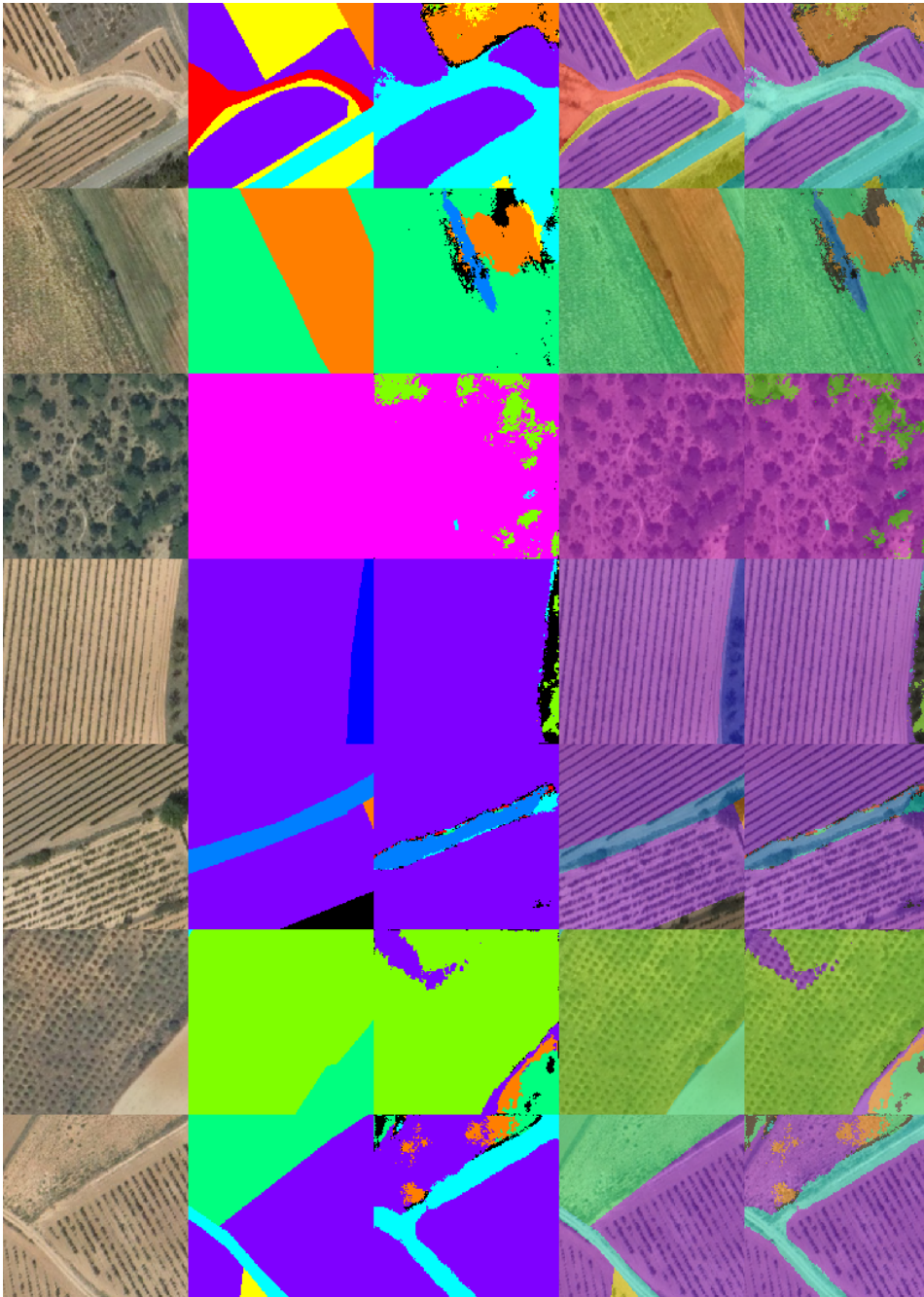
Figure 10. Loss function for the “Base” experiment with UNet for PNOA.

Table 13. Class metrics for the experiment with UNet on UOPNOA.

Class	Experiment Base			Experiment All-Purpose		
	PA	UA	IoU	PA	UA	IoU
UN	0.45	0.29	0.21	0.40	0.23	0.17
PA	0.54	0.25	0.20	0.36	0.21	0.15
SH	0.69	0.62	0.49	0.64	0.51	0.39
FO	0.21	0.78	0.20	0.52	0.63	0.40
BU	0.59	0.92	0.56	0.71	0.62	0.49
AR	0.93	0.96	0.90	0.88	0.75	0.68
GR	0.62	0.70	0.49	0.67	0.59	0.46
RO	0.77	0.51	0.45	0.79	0.37	0.34
WA	0.60	0.43	0.34	0.63	0.30	0.26
FR	0.45	0.92	0.43	0.54	0.70	0.44
VI	0.90	0.97	0.88	0.95	0.76	0.74
OT	-	-	-	0.15	0.47	0.13



**Figure 11.** Visualization of the predicted results for UNet evaluated in UOPNOA (Experiment Base). (1st col.) Original images, (2nd col.) ground truth masks, (3rd col.) predictions, (4th col.) overlapping of original images with ground truth masks, (5th col.) overlapping of original images with predictions.



**Figure 12.** Visualization of the predicted results for UNet evaluated in UOPNOA with “Other” class (Experiment All-Purpose). (1st col.) Original images, (2nd col.) ground truth masks, (3rd col.) predictions, (4th col.) overlapping of original images with ground truth masks, (5th col.) overlapping of original images with predictions.

### 3.3. Experimentation with Uos2

Experiments with DeepLabV3+ and UNet for the UOS2 dataset and discussion of their results are presented in this subsection.

#### 3.3.1. Unet with All Classes

Optimal network configuration for UNet with the dataset UOS2 is presented in Table 14 and its best training parameters in Table 15. This architecture is similar to its first official publication [20], but as the images used have ten different bands, the architecture has been changed to allow these input images. In addition, the problem to solve is not a binary classification but rather a multi-class classification. The final difference from the original implementation is that rather than using sixty-four filters on the first depth level, thirty-two are used, as this is the optimal configuration found. This means that on every depth level there are half as many filters as the original implementation. As a result, far less computational resources are needed to train the network, with no tradeoff in accuracy for this particular dataset. The number of depth levels necessary were also tested, and four were found to be optimal, coinciding with the official implementation.

**Table 14.** Network parameters for UNet on UOS2.

Network Parameters	
Input size	$256 \times 256 \times 10$
Classes	11
Depth	4
Filters on first level	32
Padding	Yes
Class weighting	Median frequency

**Table 15.** Training parameters for UNet on UOS2.

Training Parameters	
Solver	Adam
Epochs	125
Batch size	32
Learning rate	0.0005
Gradient clipping	1.0
L2 regularization	0.0001
Shuffle	Yes
Data augmenting	Yes (Mirroring on both axes)

Training parameters for Table 15 were found following the same procedure described on UNet for UOPNOA. In this case data was augmented for the training data since there were far fewer images than on UOPNOA.

After training the model, the test data was evaluated. The results obtained can be seen in Table 16. In addition to the experiments “Base” and “All-Purpose”, four more experiments were carried out. These included: experiment “RGB” to test how the dataset performs with only RGB bands, experiment “ME”, which utilizes only three multi-spectral bands (B8 NIR, B12 SWIR, B6 VRE) to compare with the RGB experiment, and two additional experiments using six and thirteen bands. Given that the “Base” experiment and the “All-Purpose” experiments use ten bands, a comparison between three, six, ten, and thirteen bands can be made. All the experiments except the “Base” experiment use the class “Other”.

Using only three bands, both the “RGB” and “ME” experiments perform badly, obtaining a PA and UA close to 10%. Moreover, there is little difference between using RGB bands or the three multi-spectral bands selected.



The experiments with six, ten, and thirteen bands have similar results. This proves that only six bands are really needed and that three bands does not provide enough data to differentiate between classes at this GSD.

A comparison between the “Base” experiment from UNet on UOPNOA (0.61 *PA* and 0.47 *UA*) and the best experiment from UNet on UOS2 (0.56 *PA* and 0.42 *UA*) reveals that UOPNOA has better results. Thus, GSD is the most important feature of an aerial imagery dataset. The next most important factor is to have more bands than simply RGB.

Finally, the “Base” experiment outperforms the “All-Purpose” experiment in the same way that occurs in the UOPNOA dataset. This can be seen in detail in Table 17

**Table 16.** Global metrics for the experiment with UNet on UOS2.

Experiment	<i>OA</i>	<i>PA</i>	<i>UA</i>	<i>IoU</i>	<i>F<sub>1</sub></i>
Base	0.647	0.569	0.428	0.323	0.489
All-Purpose	0.527	0.521	0.364	0.259	0.429
RGB	0.585	0.090	0.079	0.053	0.084
ME	0.570	0.091	0.109	0.052	0.099
6 bands	0.569	0.480	0.373	0.252	0.420
13 bands	0.589	0.463	0.371	0.261	0.412

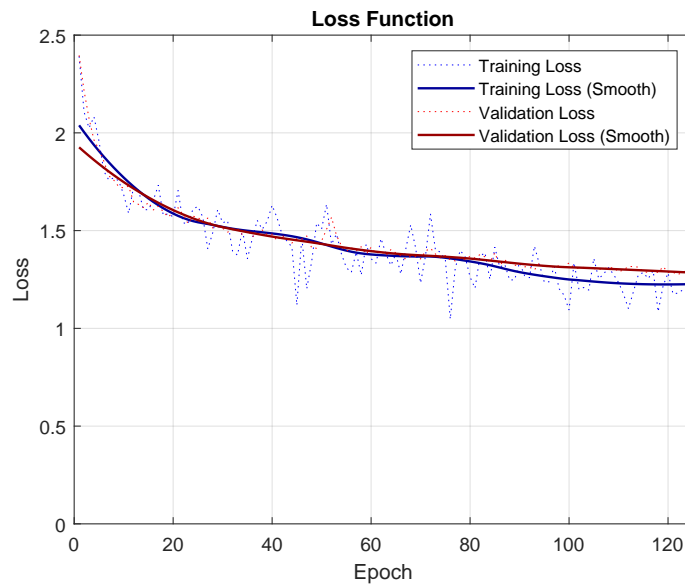
**Table 17.** Class metrics for the experiment with UNet on UOS2.

Class	Experiment Base			Experiment All-Purpose		
	<i>PA</i>	<i>UA</i>	<i>IoU</i>	<i>PA</i>	<i>UA</i>	<i>IoU</i>
UN	0.56	0.22	0.19	0.51	0.22	0.18
PA	0.42	0.28	0.20	0.46	0.21	0.17
SH	0.36	0.67	0.30	0.37	0.54	0.28
FO	0.59	0.60	0.43	0.52	0.48	0.34
BU	0.81	0.50	0.45	0.74	0.39	0.34
AR	0.77	0.94	0.73	0.69	0.85	0.62
GR	0.62	0.37	0.30	0.58	0.28	0.23
RO	0.36	0.16	0.12	0.40	0.13	0.11
WA	0.61	0.22	0.19	0.63	0.17	0.16
FR	0.34	0.10	0.08	0.45	0.08	0.07
VI	0.78	0.59	0.51	0.76	0.56	0.48
OT	-	-	-	0.09	0.38	0.08

To prove that the tuning has been done correctly, progress from the loss function for training and validation can be seen in Figure 13. Training loss and validation loss both start to stabilize.

Observing the visualization of the predictions from both experiments (Figures 14 and 15), it is clear that UOS2 is far more complex than UOPNOA. There are many different classes in a single image and their area is considerably smaller. Taking this into consideration, the results provided are good. When examining the predictions globally, they are very similar to the ground truth.

Finally, when comparing the use of the class “Other”, the predictions are similar although there is a significant difference in results.



**Figure 13.** Loss function for the “Base” experiment with UNet for UOS2.

### 3.3.2. Unet with Simplified Classes

The purpose of these experiments is to test the effects of using fewer classes by merging similar classes together. In addition, the effects of including the all-purpose class “Other” with fewer and more stable classes is studied. To make these new classes, the confusion matrix from the experiments is used to distinguish which classes are the most similar. Only classes that are similar from a logical standpoint are merged. Classes that do not obtain good results and have little relevance are not used.

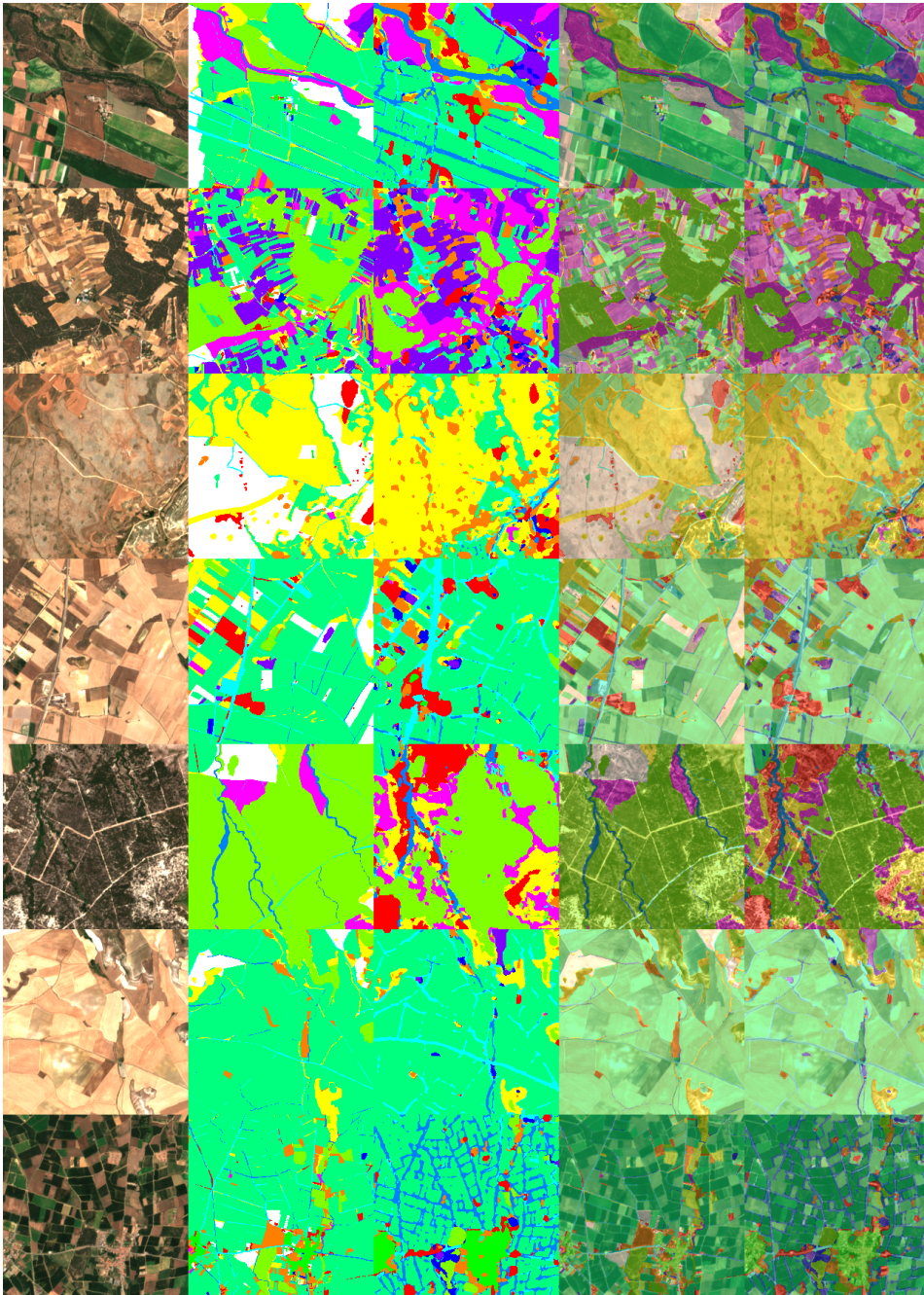
Optimal configurations from the previous experiments with eleven classes are reused (Tables 18 and 19). Experiments were conducted to verify that these configurations are still optimal.

**Table 18.** Network parameters for UNet with simplified classes on UOS2.

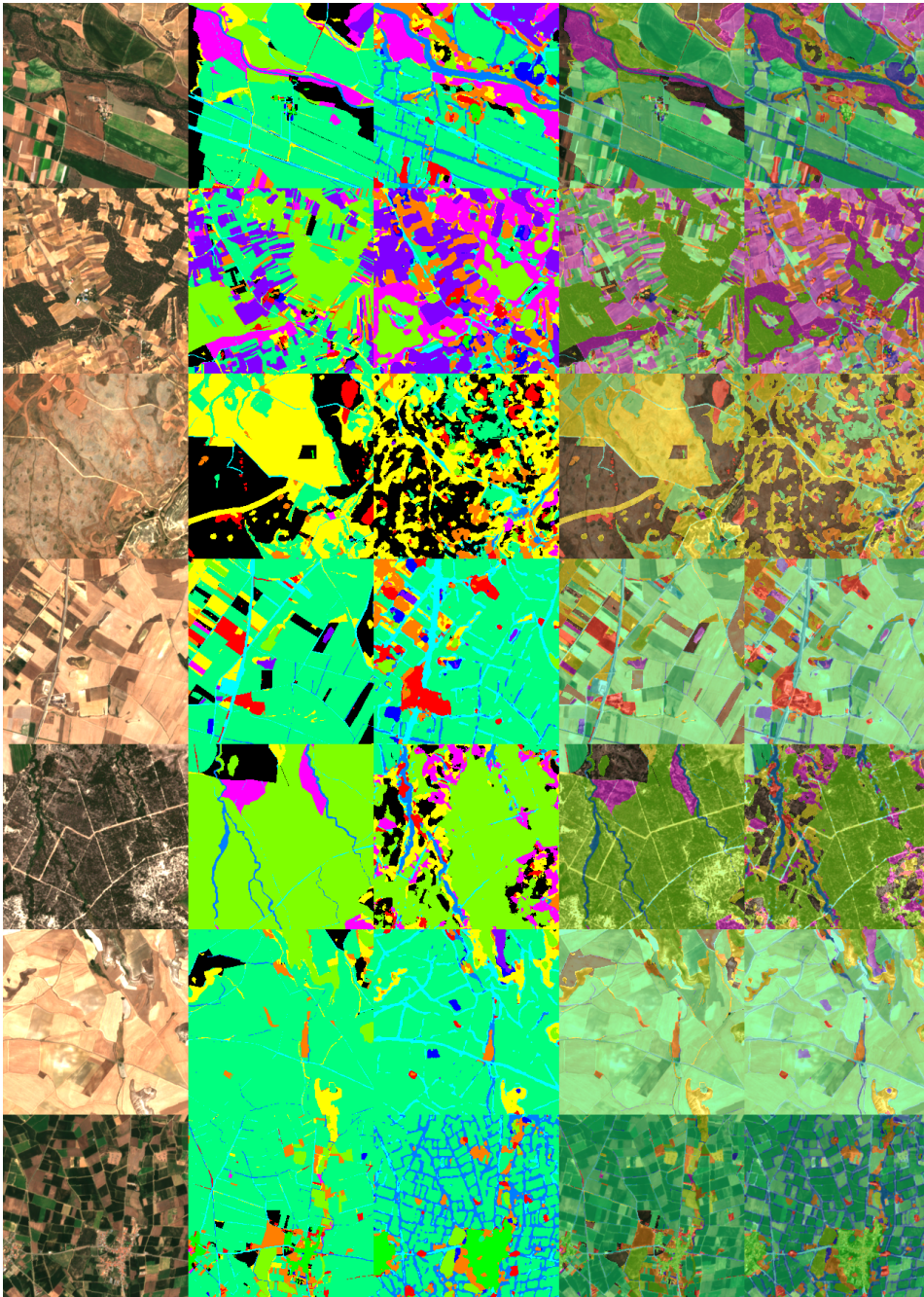
Network Parameters	
Input size	256 × 256 × 10
Classes	4
Depth	4
Filters on first level	32
Padding	Yes
Class weighting	Median frequency

**Table 19.** Training parameters for UNet with simplified classes on UOS2.

Training Parameters	
Solver	Adam
Epochs	125
Batch size	32
Learning rate	0.0005
Gradient clipping	No
L2 regularization	No
Shuffle	Yes



**Figure 14.** Visualization of the predicted results for UNet evaluated in UOS2 (Experiment Base). (1st col.) Original images, (2nd col.) ground truth masks, (3rd col.) predictions, (4th col.) overlapping of original images with ground truth masks, (5th col.) overlapping of original images with predictions.



**Figure 15.** Visualization of the predicted results for UNet evaluated in UOS2 with “Other” class (Experiment All-Purpose). (1st col.) Original images, (2nd col.) ground truth masks, (3rd col.) predictions, (4th col.) overlapping of original images with ground truth masks, (5th col.) overlapping of original images with predictions.

After training the model, an evaluation with the test data was carried out. The results obtained can be seen in Table 20. “Base” and “All-Purpose” experiments (with and without class “Other”) are carried out to compare with previous experiments.

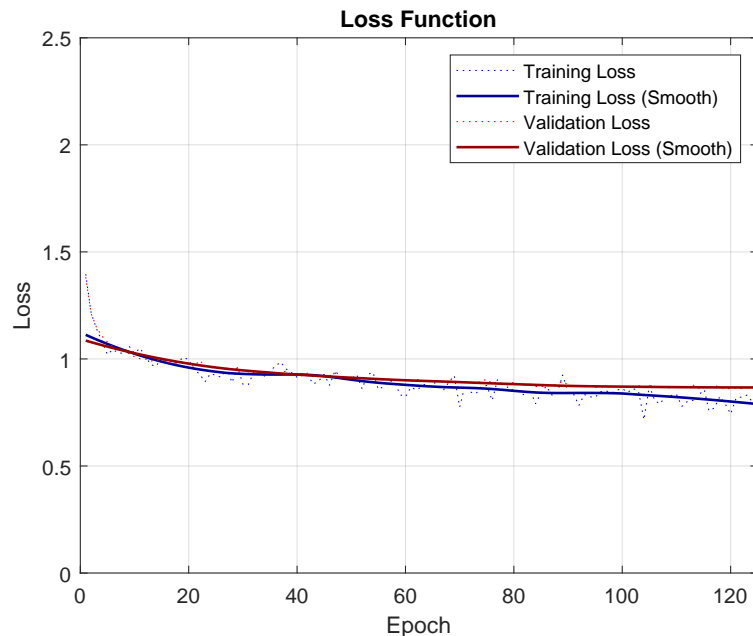
Globally, the results coincide with previous experiments in that the “Base” experiment outperforms the “All-Purpose” experiment. Merging and reducing the number of classes drastically improves the accuracy of the predictions, obtaining an improvement of almost 22% and 25% on *PA* and *UA*, respectively, for the best experiments.

**Table 20.** Global metrics for the experiment with UNet with simplified classes on UOS2.

Experiment	<i>OA</i>	<i>PA</i>	<i>UA</i>	<i>IoU</i>	<i>F<sub>1</sub></i>
Base	0.822	0.786	0.677	0.576	0.727
All-Purpose	0.650	0.639	0.578	0.402	0.607

Looking at the results per class from Table 21, outstandingly high *PA* and *UA* can be seen for every class except BURO. As seen in previous experiments, when the “Other” class is used, all the remaining classes lose accuracy.

To prove that the tuning has been done correctly, progress from the loss function for training and validation can be seen in Figure 16. Validation loss stabilizes but training loss keeps lowering; if more epochs are executed, overfitting of the model would start to occur.

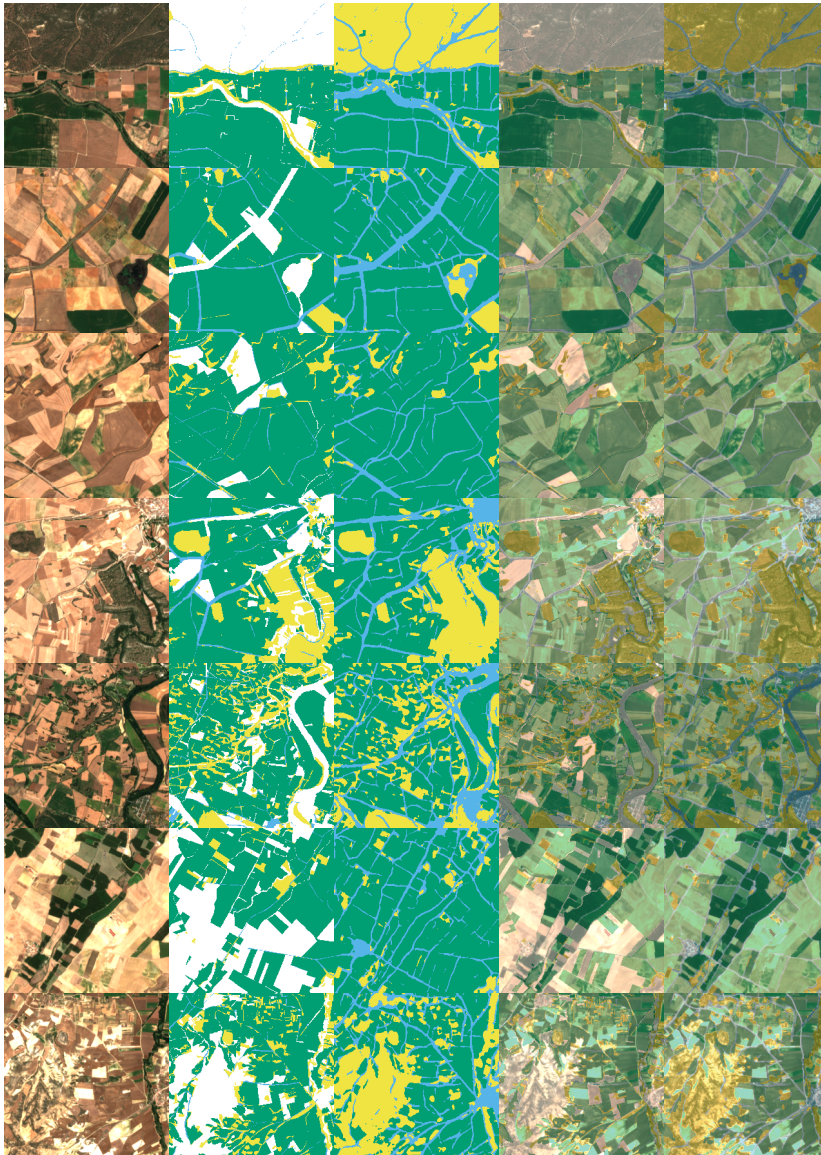


**Figure 16.** Loss function for the “Base” experiment with UNet for UOS2 with simplified classes.

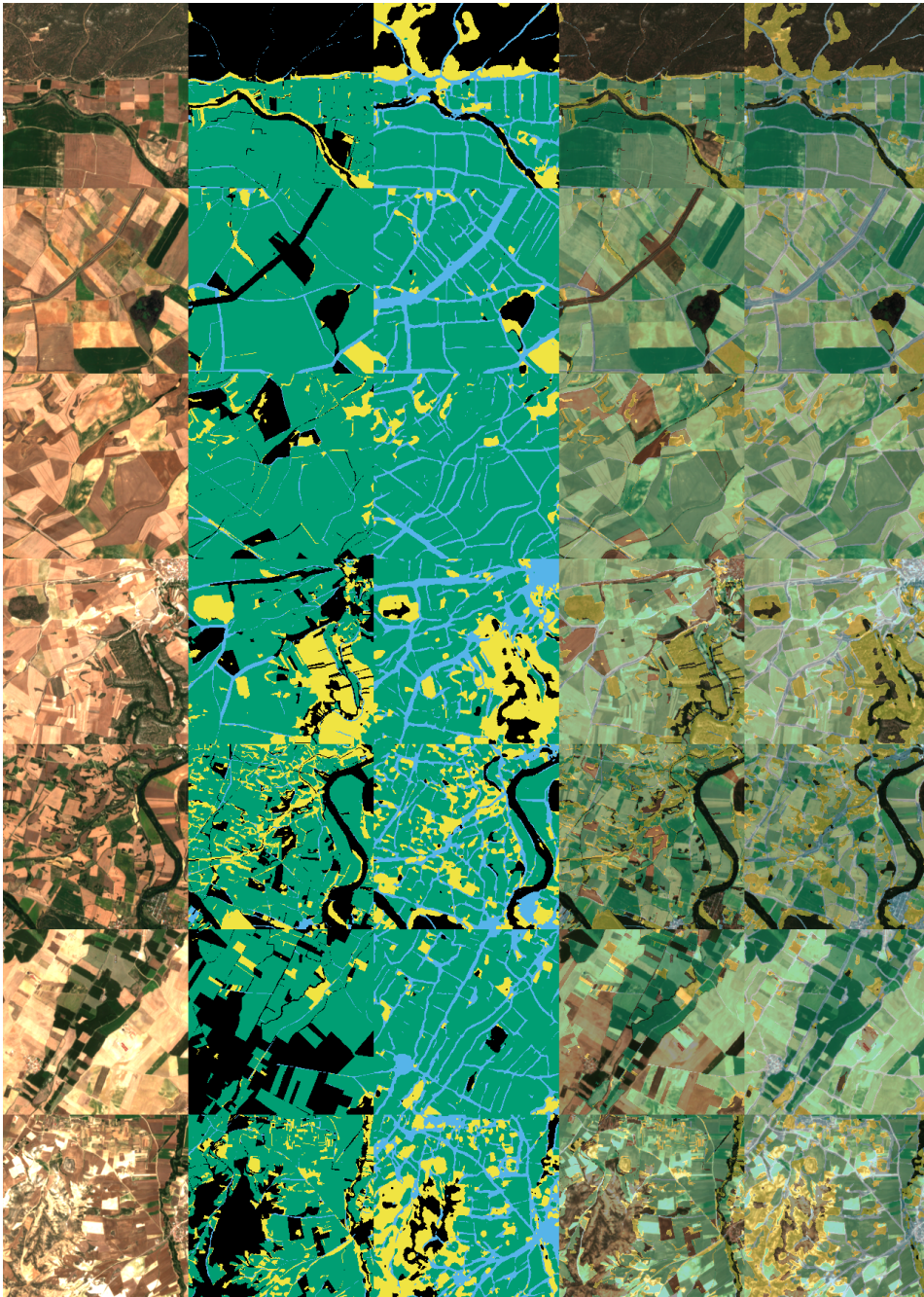
**Table 21.** Class metrics for the experiment with UNet with simplified classes on UOS2. These classes consist of PASHGR (PA+SH+GR—all pastures), BURO (BU+RO—all infrastructures), ARVI (AR+VI—arable lands and vineyards), and OT.

Class	Experiment Base			Experiment All-Purpose		
	<i>PA</i>	<i>UA</i>	<i>IoU</i>	<i>PA</i>	<i>UA</i>	<i>IoU</i>
PASHGR	0.82	0.81	0.69	0.71	0.52	0.43
BURO	0.70	0.25	0.23	0.70	0.17	0.16
ARVI	0.83	0.95	0.80	0.77	0.85	0.68
OT	-	-	-	0.36	0.75	0.21

To prevent confusion between the classes in Figures 17 and 18, a change in the color keys of classes has been made. In this visualization it is interesting to note that the class BURO, which performs the worst, has predictions around the boundaries of the plots. It seems to confuse the boundaries with roads. Furthermore, roads that are not classified as such in the original ground truth are predicted, outperforming the ground truth. This means that the low accuracy in this class can be seen as an error in the ground truth, which is logical given that not every dirt road is registered in the SIGPAC. This behavior may be beneficial to the predictions even though numerically the UA is low.



**Figure 17.** Visualization of the predicted results for UNet evaluated in UOS2 for four classes (Experiment Base). (1st col.) Original images, (2nd col.) ground truth masks, (3rd col.) predictions, (4th col.) overlapping of original images with ground truth masks, (5th col.) overlapping of original images with predictions.



**Figure 18.** Visualization of the predicted results for UNet evaluated in UOS2 with four classes and “Other” class (Experiment All-Purpose). (1st col.) Original images, (2nd col.) ground truth masks, (3rd col.) predictions, (4th col.) overlapping of original images with ground truth masks, (5th col.) overlapping of original images with predictions.

### 3.4. Deployment

After training the models, a prototype for inference with a microservice architecture that follows modern standards using Flask was created. To ensure that all the required dependencies are met and to make deployment easier, Docker was used to containerize the microservice. Then, the different infrastructures were deployed to test performance. The model selected for this deployment is the UNet architecture with simplified classes from the “Base” experiment. The deep learning tool used for the inference is PyTorch.

Semantic segmentation networks can only operate with the same resolution used for training when the inference is done. This means that the only way to predict larger images is by cropping them and then fusing the predictions together. Each cropped image will take the same amount of time because every pixel in the image is predicted, so the same number of predictions are made for every image fed to the network. For this reason, the method used to test performance is based on the number of images requested per petition. These images have the same resolution as those used for training.

The majority of the tools used in Geographic Information Systems are designed to work with geojson as this format is easier to work with than a mask. To add realism to the service, a process to convert the output from the network to a georeferenced geojson with every region and land use type predicted on it, is added at the end of each petition. This adds many computations to the prototype as it must polygonize the mask from the output of the inference and convert it to a geojson. To limit the size the geojson, the Ramer–Douglas–Peucker algorithm [38] is executed to simplify the numbers of points defining the polygons. This process is adapted to make use of multiple CPU cores.

In Figure 19, performance by infrastructure is presented. “Local:A-B” are the machines used to train the networks in this work. These machines are connected via 1Gbit LAN. The rest of the infrastructures are provided by Microsoft Azure. Infrastructures “NC6”, “A4 v2”, “F4s v2”, and “D4as v4” are IaaS. CaaS and FaaS implementations from Azure are also evaluated. In all the cases, the client is a local machine from outside Azure’s network with a connectivity of approximately 250 Mbps.

For the evaluation of performance, times are defined as follows: “Load” is the time to access the libraries, loading the images into the memory, etc. “Prediction” is the time needed for the model to make the predictions. “Results” is the time required to convert the predictions into a geojson format. For this task, polygonization of the predictions masks, a simplification of their results using the algorithm RDP, and the generation of geojsons are timed. “Network” is the time needed to upload the images to be predicted and the time needed to download the predictions and their geojsons.

Performance for one image is presented in Figure 19. To compare performance between infrastructures, the latency ratio is shown on top of each stacked bar. This ratio is calculated as the total time of a given experiment divided by the total time of the best experiment. Cost-performance metrics for ten images are presented in Figure 20.

In Table 19, a minimal improvement when using GPU can be seen. “Load” times are negligible. “Prediction” depends on the single core speed and the number of cores. However, using a GPU is always faster. “Results” have the greatest impact on time, depending mainly on single core speed. “Network” is not affected by the infrastructure, except in the case of “Local:A-B” where both computers are in the same local network.

Table 20 shows almost no improvement when using a GPU, but the cost is multiplied. A tradeoff in cost and performance must be done since CaaS is more economical, but IaaS solutions such as “D4as v4” and “F4s v2” have better performance. “Local:A-B” seems to be the best option even when accounting for electricity cost and amortization over five years. Setting up and maintaining this kind of infrastructure can be excessively complex. However, cloud options get upgrades to the hardware from time to time. FaaS cost performance is calculated as if there were always a petition running for the entire hour. This means that the cost is zero if there are no petitions, but it would be greater if there were multiple simultaneous petitions causing multiple instances of FaaS to execute at the



same time, multiplying its cost. The rest of the infrastructures are priced for availability and not use: they can await petition at the same cost.

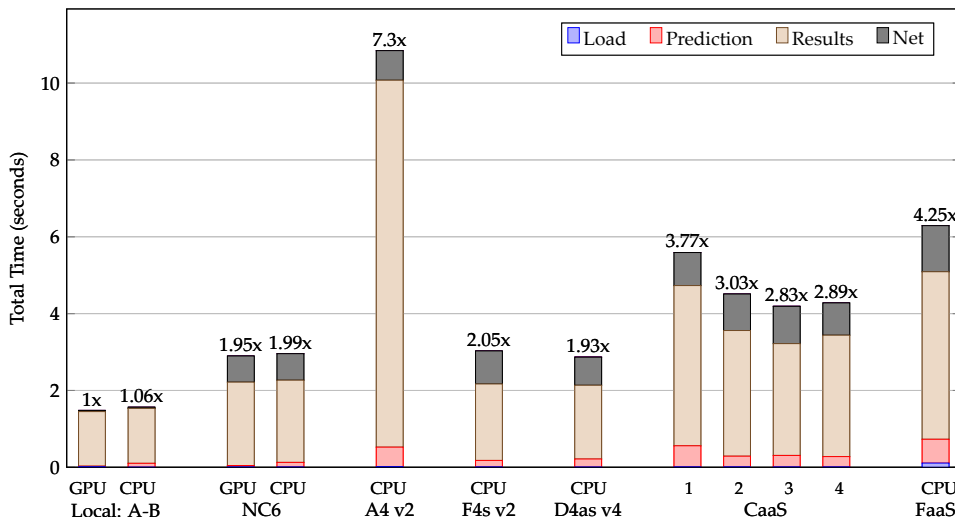


Figure 19. Performance in seconds per infrastructure.

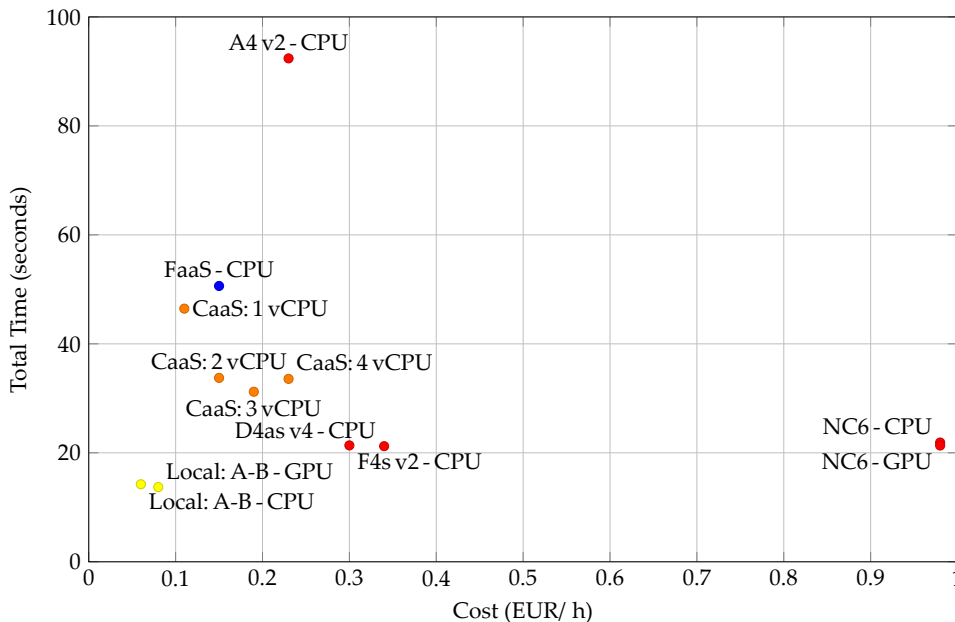


Figure 20. Cost performance per infrastructure. Prices and the computing power are subject to change (April 2021).

#### 4. Conclusions

Land use classification in aerial imagery, especially satellite imagery, is a complex task. Thanks to recent advances in convolutional neural networks, a specific technology called semantic segmentation seems the most appropriate to improve on actual results. UNet is the most widely used semantic segmentation network thanks to its great flexibility, but when compared to DeepLabV3+, the latter performs far better. As a counterpoint,

DeepLabV3+ lacks an implementation that can use more than three bands, so it is the best option for RGB. However, for satellite imagery, where most of the data is presented as extra multi-spectral bands, it falls short. This gives the advantage to UNet for satellite imagery. Taking this into consideration, the most important factor for good accuracy in aerial datasets is their GSD. In the case of UOPNOA, impressive results were achieved with a GSD of 0.25 m/pixel (PA and UA of 78% and 75% respectively for DeepLabV3+ and 61% and 47% for UNet). However, in UOS2 using only RGB bands, its results are completely unusable with a PA and UA of less than 10%. UOS2 has a GSD of 10–60 m/pixel, which means that the difference in resolution is from 40 to 240 times worse.

When using UOS2 with at least six bands, including RGB and multi-spectral bands, its accuracy rises greatly (up to 48% PA and 37% UA), obtaining far better results but still falling short when compared to UOPNOA. However, this difference is greatly reduced when using more bands. While it is true that the GSD is of fundamental importance, the number and type of bands used are equally important. In this way, satellite imagery can equal the results of aircraft imagery. It is interesting to note that using more than six bands, even as many as thirteen, gives no significant improvement in the classification of land use.

Merging together similar or confusing classes improves the predictions noticeably (up to 78% PA and 67% UA for UNet with UOS2). This proves that it is better to use fewer, well defined classes. This can be verified with the experiment for UNet in UOS2 that uses only three classes, as this is one of the experiments with the best results even when compared with DeepLabV3+ on UOPNOA.

Using an all-purpose class is counterproductive, causing all the remaining classes to lose accuracy. There is a large variability associated with pixels that do not belong to any of the target classes. Semantic segmentation models will have to deal with these “unknown” classes when used in practice, unless the user removes these pixels beforehand. This could be done by selecting only the region of a desired plot, although this reduces the attractiveness of the approach. This approach is only useful if the input images have no pixels that belong to the target classes. Therefore, while this is acceptable for research purposes, only specific use cases can benefit from it.

The use of the two newly created datasets for land use classification in aerial imagery, UOPNOA and UOS2, are proven to be good for comparisons and evaluation of different models thanks to their great complexity and variability.

Finally, for performance with an inference prototype, network time alone takes more time than the predictions, even using only CPU. This is extremely important to take into account when choosing an infrastructure to offer services, like the one presented in this work. The use of a GPU is not recommended as it increases cost greatly, with no significant improvement in performance.

**Author Contributions:** Conceptualization, O.D.P., D.G.L., D.F.G., and R.U.; methodology, O.D.P.; software, O.D.P.; validation, D.G.L., D.F.G., and R.U.; formal analysis, O.D.P., D.F.G., and R.U.; investigation, O.D.P. and D.G.L.; resources, D.F.G. and R.U.; data curation, O.D.P.; writing—original draft preparation, O.D.P.; writing—review and editing, O.D.P., D.F.G., and R.U.; visualization, O.D.P.; supervision, Á.A.; project administration, D.F.G.; funding acquisition, D.F.G. and Á.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by SERESCO S.A under the contract FUIO-20-018 and also by the project RTI2018-094849-B-I00 of the Spanish National Plan for Research, Development and Innovation.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** UOPNOA and UOS2 datasets are publicly available in the platform Zenodo with the following DOI (last accessed on 10 June 2021): [doi.org/10.5281/zenodo.4648002](https://doi.org/10.5281/zenodo.4648002).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Kussul, N.; Lavreniuk, M.; Skakun, S.; Shelestov, A. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 778–782. [CrossRef]
- Shelestov, A.; Lavreniuk, M.; Vasiliev, V.; Shumilo, L.; Kolotii, A.; Yailymov, B.; Kussul, N.; Yailymova, H. Cloud Approach to Automated Crop Classification Using Sentinel-1 Imagery. *IEEE Trans. Big Data* **2020**, *6*, 572–582. [CrossRef]
- Inglada, J.; Arias, M.; Tardy, B.; Hagolle, O.; Valero, S.; Morin, D.; Dedieu, G.; Sepulcre, G.; Bontemps, S.; Defourny, P.; et al. Assessment of an Operational System for Crop Type Map Production Using High Temporal and Spatial Resolution Satellite Optical Imagery. *Remote Sens.* **2015**, *7*, 12356–12379. [CrossRef]
- Clemente, J.; Fontanelli, G.; Ovando, G.; Roa, Y.; Lapini, A.; Santi, E. Google Earth Engine: Application Of Algorithms for Remote Sensing Of Crops In Tuscany (Italy). In Proceedings of the 2020 IEEE Latin American GRSS & ISPRS Remote Sensing Conference (LAGIRS), Santiago, Chile, 21–26 March 2020; IEEE: New York, NY, USA, 2020; pp. 195–200.
- Stoian, A.; Poulain, V.; Inglada, J.; Poughon, V.; Derksen, D. Land Cover Maps Production with High Resolution Satellite Image Time Series and Convolutional Neural Networks: Adaptations and Limits for Operational Systems. *Remote Sens.* **2019**, *11*, 1986. [CrossRef]
- Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
- Everingham, M.; Eslami, S.M.A.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
- Sistema de Información Geográfica de Parcelas Agrícolas (SIGPAC). Available online: <https://www.mapa.gob.es/es/agricultura/temas/sistema-de-informacion-geografica-de-parcelas-agricolas-sigpac/> (accessed on 26 February 2021).
- Sentinel-2: Satellite Imagery, Overview, and Characteristics. Available online: <https://eos.com/sentinel-2/> (accessed on 26 February 2021).
- Plan Nacional de Ortofotografía Aérea. Available online: <https://pnoa.ign.es/> (accessed on 26 February 2021).
- Tatsumi, K.; Yamashiki, Y.; Torres, M.A.C.; Taibe, C.L.R. Crop classification of upland fields using Random forest of time-series Landsat 7 ETM+ data. *Comput. Electron. Agric.* **2015**, *115*, 171–179. [CrossRef]
- Shelestov, A.; Lavreniuk, M.; Kussul, N.; Novikov, A.; Skakun, S. Exploring Google Earth Engine Platform for Big Data Processing: Classification of Multi-Temporal Satellite Imagery for Crop Mapping. *Front. Earth Sci.* **2017**, *5*, 17. [CrossRef]
- Mandal, D.; Kumar, V.; Rao, Y.S. An assessment of temporal RADARSAT-2 SAR data for crop classification using KPCA based support vector machine. *Geocarto Int.* **2020**, 1–13. [CrossRef]
- Cutler, A.; Cutler, D.R.; Stevens, J.R., Random Forests. In *Ensemble Machine Learning: Methods and Applications*; Springer: Boston, MA, USA, 2012; pp. 157–175.5. [CrossRef]
- Cristianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*; Cambridge University Press: Cambridge, UK, 2000; doi:10.1017/CBO9780511801389. [CrossRef]
- Zhou, Z.; Li, S.; Shao, Y. Crops classification from sentinel-2A multi-spectral remote sensing images based on convolutional neural networks. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*; IEEE: New York, NY, USA, 2018; pp. 5300–5303.
- Li, R.; Duan, C.; Zheng, S. MACU-Net Semantic Segmentation from High-Resolution Remote Sensing Images. *arXiv* **2020**, arXiv:2007.13083.
- Matvienko, I.; Gasanov, M.; Petrovskaia, A.; Jana, R.B.; Pukalchik, M.; Oseledets, I. Bayesian aggregation improves traditional single image crop classification approaches. *arXiv* **2020**, arXiv:2004.03468.
- Zhang, P.; Ke, Y.; Zhang, Z.; Wang, M.; Li, P.; Zhang, S. Urban land use and land cover classification using novel deep learning models based on high spatial resolution satellite imagery. *Sensors* **2018**, *18*, 3717. [CrossRef] [PubMed]
- Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
- Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *ECCV*; Springer: Berlin/Heidelberg, Germany, 2018.
- Bragagnolo, L.; Rezende, L.; da Silva, R.; Grzybowski, J. Convolutional neural networks applied to semantic segmentation of landslide scars. *CATENA* **2021**, *201*, 105189. [CrossRef]
- Karki, S.; Kulkarni, S. Ship Detection and Segmentation using U-net. In *2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*; IEEE: New York, NY, USA, 2021; pp. 1–7.
- Li, M.; Stein, A. Mapping land use from high resolution satellite images by exploiting the spatial arrangement of land cover objects. *Remote Sens.* **2020**, *12*, 4158. [CrossRef]
- Du, S.; Du, S.; Liu, B.; Zhang, X. Incorporating DeepLabv3+ and object-based image analysis for semantic segmentation of very high resolution remote sensing images. *Int. J. Digit. Earth* **2021**, *14*, 357–378. [CrossRef]
- Su, H.; Peng, Y.; Xu, C.; Feng, A.; Liu, T. Using improved DeepLabv3+ network integrated with normalized difference water index to extract water bodies in Sentinel-2A urban remote sensing images. *J. Appl. Remote Sens.* **2021**, *15*, 018504. [CrossRef]
- Centro de Descargas del CNIG (IGN). Available online: <https://centrodedescargas.cnig.es/CentroDescargas/index.jsp> (accessed on 21 May 2021).
- Sentinel Hub. Available online: <https://www.sentinel-hub.com/> (accessed on 21 May 2021).

29. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv* **2014**, arXiv:1412.7062.
30. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [[CrossRef](#)] [[PubMed](#)]
31. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
32. Liu, C.; Chen, L.C.; Schroff, F.; Adam, H.; Hua, W.; Yuille, A.L.; Li, F.-F. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 82–92.
33. Fernandez-Moral, E.; Martins, R.; Wolf, D.; Rives, P. A new metric for evaluating semantic segmentation: Leveraging global and contour accuracy. In *2018 IEEE Intelligent Vehicles Symposium (iv)*; IEEE: New York, NY, USA, 2018; pp. 1051–1056.
34. Taghizadeh-Mehrjardi, R.; Mahdianpari, M.; Mohammadimanesh, F.; Behrens, T.; Toomanian, N.; Scholten, T.; Schmidt, K. Multi-task convolutional neural networks outperformed random forest for mapping soil particle size fractions in central Iran. *Geoderma* **2020**, *376*, 114552. [[CrossRef](#)]
35. Yang, Q.; Zhang, H.; Xia, J.; Zhang, X. Evaluation of magnetic resonance image segmentation in brain low-grade gliomas using support vector machine and convolutional neural network. *Quant. Imaging Med. Surg.* **2021**, *11*, 300. [[CrossRef](#)]
36. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
37. Qian, N. On the momentum term in gradient descent learning algorithms. *Neural Netw.* **1999**, *12*, 145–151. [[CrossRef](#)]
38. Ramer, U. An iterative procedure for the polygonal approximation of plane curves. *Comput. Graph. Image Process.* **1972**, *1*, 244–256. [[CrossRef](#)]

### 5.1.2. Semantic segmentation for non-destructive testing with step-heating thermography for composite laminates

- Pedrayes, O. D., Lema, D. G., Usamentiaga, R., Venegas, P., & García, D. F. (2022). *Semantic segmentation for non-destructive testing with step-heating thermography for composite laminates*. *Measurement*, 200, 111653.
- DOI: [10.1016/j.measurement.2022.111653](https://doi.org/10.1016/j.measurement.2022.111653)
- El índice de impacto de la revista *Measurement* en 2021 fue 5.131 (Q1, 83.59%) y el índice de impacto a 5 años, 4.639.
- Citas: 2



# Semantic segmentation for non-destructive testing with step-heating thermography for composite laminates

Oscar D. Pedrayes<sup>a,\*</sup>, Darío G. Lema<sup>a</sup>, Rubén Usamentiaga<sup>a</sup>, Pablo Venegas<sup>b</sup>, Daniel F. García<sup>a</sup>

<sup>a</sup> Department of Computer Science and Engineering, University of Oviedo, Campus de Viesques, Gijón, 33204, Asturias, Spain

<sup>b</sup> Aeronautical Technologies Centre (CTA), Juan de la Cierva 1, Miñano, 01510, Basque Country, Spain

## ARTICLE INFO

MSC:  
0000  
1111

### Keywords:

Quality control  
Destructive  
Defects  
Flaws  
Semantic segmentation  
Deep learning

## ABSTRACT

In this paper, semantic segmentation networks such as UNet and DeepLabV3+ are evaluated and compared against Random Forest and Support Vector Machines in the field of step-heating active infrared thermography for subsurface defect detection and localization. To collect information from an entire digital recording sequence into a particular image, post-processing methods such as PCT, PPT, Kurtosis, Skewness and TSR are used. Two datasets are created, one with 3-channel images using PCT, and one using all the above post-processing methods to condense the heating and cooling processes into 30-channel images. This evaluation study shows that DeepLabV3+ is able to detect most defects in specimens with a similar structure to training samples without false positives even for defects of different depth and area. UNet requires the use of 30-channel images to achieve results closer to DeepLabV3+. Random Forest and Support Vector Machines are unable to compete with the recent methods as they are unable to detect defects correctly.

## 1. Introduction

Quality control is of significant interest in industry, causing continuous efforts to improve on previous methods. Non-destructive testing (NDT) is a set of analysis methods to inspect, test, and evaluate materials, components, or systems without harming the object. This approach has advantages over destructive testing (DT): it can be used to analyze every item instead of one for each batch and the repeatability of the tests make it possible to repair the products, which leads to lower cost since items need not be replaced after testing. Additionally, since the object is not damaged during the test, NDT tests can also be applied to detect failures as a maintenance method during the lifetime of the product, improving long-term use, and safety [1].

NDT methods can be divided into Contact and Non-Contact. Contact methods are ultrasonic testing, eddy current testing, magnetic testing, and penetrant testing. Non-Contact methods are air-coupled ultrasonic, radiography testing, thermography, shearography, and visual inspection [2].

Today, the most common analysis of defects is done manually by an expert in the field. The results of non-contact NDT inspections, and of some automated contact techniques as well, are generally represented through images. In these situations, the experts usually use image post-processing techniques to make their job faster. Even so, this approach is still costly and time-consuming compared to the potential of a solution based on deep learning.

In recent years, deep learning approaches have made significant advances in the field of infrared thermography [3–7]. Infrared thermography does not need coupling media facilitating the production and speed of scans. This approach has no harmful side effects (such as radiation in X-ray evaluation), improving safety and inspection rates in prolonged use cases. Infrared thermography can be grouped into passive and active. Passive infrared thermography uses the differences in the temperature of the product under natural conditions, that is, without applying heat to the object [8]. Active infrared thermography evaluates temperature differences during and after a heating process. It is important to mention that there is no agreement on the appropriate stimulation and post-processing methodology for a given material and flaw type [9]. This heating process can be done using photographic flashes, halogen lamps, ultrasonic transducers, or other methods [10]. Depending on the method used to heat the item, active infrared thermography can be classified in [6]:

1. Pulsed Thermography (PT): the object is heated for a short time, typically with a flash lamp or a coil for Eddy Current Pulse Thermography [11].
2. Step-Heating Thermography (SHT): the object is heated for longer periods than PT, reaching deeper defects.

\* Corresponding author.

E-mail address: [UO251056@uniovi.es](mailto:UO251056@uniovi.es) (O.D. Pedrayes).

3. Lock-in Thermography (LT): the object is heated by a modulated heatwave. The temperature changes are compared to the original heatwave revealing defects.

This paper evaluates multiple state-of-the-art methods for image segmentation. Image segmentation is the task of grouping the pixels of an image by creating a segmentation mask. High-level segmentation algorithms generate an easily interpretable classification such as bicycle or road using low-level features, including contrast levels, edges, textures, etc. The most common approaches to image segmentation are shown below:

- Threshold segmentation is the simplest method and consists of classifying pixels with respect to a threshold value.
- Edge-based segmentation is one of the most common approaches. This method identifies edges of different objects in an image using differences in texture, contrast, gray level, color, saturation and other features.
- Region-based segmentation algorithms find groups of pixels by locating seed points. The seed points increase or decrease in size and can merge together to produce different regions.
- Watershed segmentation treats the image as if it were a topographic map. It considers the brightness of a pixel as its height and finds the lines that run along the top of those ridges.
- Clustering algorithms divide the image into clusters of pixels that have similar characteristics. It separates the data elements into clusters where the elements in one cluster are more similar compared to the elements present in other clusters.
- Convolutional networks generate low-level feature maps in an automated fashion. This means that these features do not need to be easily understood by humans. After generating the low-level feature maps, neural networks recognize the relationships between the different features to classify the pixels of the image. Neural networks for the classification of each pixel in an image are known as semantic segmentation networks. If the distinction between multiple instances or objects of the same class is added, it is known as instance segmentation. And if both ideas are combined, so that there are classes without instances and classes with instances, it is known as panoptic segmentation.

Semantic segmentation is one of the most recent methods for image segmentation and has proven to be of great use in other fields such as autonomous driving [12] or crop classification [13]. This approach seems the most suitable given the growing trend for using deep learning models with active infrared thermography [6,7]. Semantic segmentation networks are evaluated for defect localization in composites using step-heating thermography. Given its popularity and the flexibility of its structure to adapt to changes in the required inputs, the semantic segmentation network UNet [14] is used for this evaluation. In addition, the semantic segmentation network DeepLabV3+ [15] is also evaluated, given its more recent and complex architecture. Then, as a basis for comparisons, the older methods Random Forest (RF) and Support Vector Machines (SVM) are used as well.

In a thermographic NDT inspection, the raw results consist of a sequence of thermal images that contain the temperature evolution history of each pixel in the observed scene. There is an explicit limitation in VRAM when using convolutional neural networks, requiring information from every frame about the heating and cooling processes to be compiled into different channels of a particular multichannel image. Post-processing techniques are used to accomplish this: Principal Component Thermography (PCT), Pulsed Phase Thermography (PPT), Kurtosis, Skewness, and Thermographic Signal Reconstruction (TSR). These methods help compile information from the whole sequence into different channels of a particular image and improve the signal-to-noise ratio (SNR). These methods will be discussed in more depth in the "Post-processing methods" subsection. In this way, the network can recognize patterns of all the frames from a sequence simultaneously.

The most obvious advantage of UNet is that it is capable of processing images with more than three channels. In this study, UNet is evaluated with images consisting of 30 channels using the methods described. However, to provide a fair comparison, images with three channels are also tested. This allows for a comparison with another more recent semantic segmentation network DeepLabV3+ [15], and other older methods such as Random Forest (RF) [16] and Support Vector Machines (SVM) [17].

Recent works tend to use simple or manually created convolutional network architectures [18,19] and older methods for object detection such as FasterRCNN [20]. There are few papers that use more recent, complex semantic segmentation architectures [4,21]. Those that do use a more modern architecture typically use UNet or one of its variations, but it appears that none of them explore the use of more than three channels against the use of only three channels per image. Moreover, the use of DeepLabV3+ in the field of defect detection is scarce [22], and to the best of the authors' knowledge, non-existent on the subject of subsurface defect detection.

The composite laminate evaluated in this work is a carbon-fiber-reinforced polymer (CFRP) laminate. Known for its strength-to-weight ratio and rigidity, it is often used in aircraft, cars, or bicycle frames [23]. NDT methods are preferred for CFRP since this material is costly, and an impact can create delamination inside the material, provoking subsurface damage invisible on the surface. Creating large datasets in NDT thermography is a costly and time-consuming process. For this reason, many papers use only a few specimens for their studies [24–26], so approaches that do not require large datasets (as is the case with UNet), or the need to use a pretrained model (as is the case with DeepLabV3+), are required. In this study, only one specimen is used to generate the datasets for training. By rotating the specimen 10°, up to 36 different digital recordings are generated with non-repeating data. Since the specimen has a different illumination and background for each digital recording, and it has to be heated and cooled again, the resulting data can be considered new and non-repeating, unlike other methods that consist of rotating the images. To further validate the trained models, another two new specimens on which to perform the testing are added.

The dataset containing the training and testing samples for semantic segmentation is described in the "Dataset" subsection and is released for public usage in the following DOI: <https://doi.org/10.5281/zenodo.5426792>.

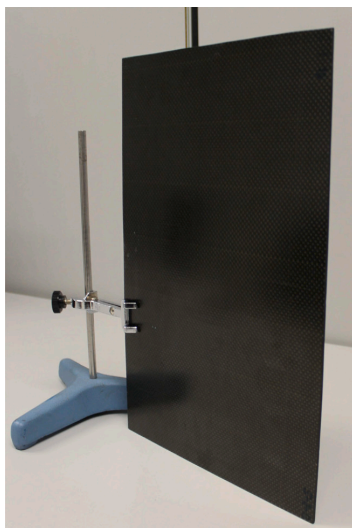
## 2. Materials and methods

### 2.1. Carbon-fiber-reinforced polymer laminate

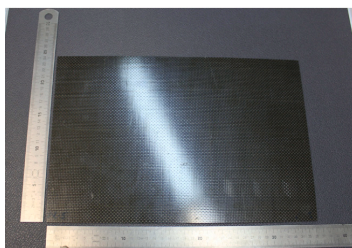
CFRP is a composite material composed of a reinforced carbon fiber, and a matrix to bind the reinforcements together. Fig. 1 shows: a general photograph of the specimen to be used for training (Fig. 1(a)); a photograph showing its measurements (Fig. 1(b)); and a photograph showing the location of the defects (Fig. 1(c)). The 360 mm × 240 mm specimen is 2.5 mm thick, following a 12-ply structure as seen in Fig. 2.

The specimen has artificially induced flaws. In this specimen there are two different types of defects: Polytetrafluoroethylene (PTFE) thin films, and steel chips defects. There are 12 defects, 9 are PTFE thin films and 3 are steel chips defects. The PTFE films simulate delaminations, which are common defects in composite materials produced by the separation of adjacent plies, while the steel inserts simulate accidental inclusion of small pieces of cutting tools used during the manufacturing process of the material.

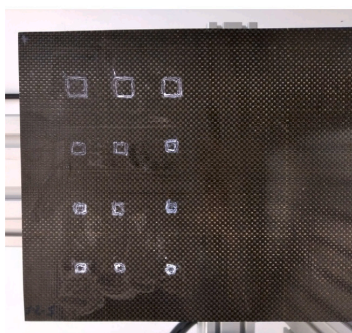
There are three different sizes of PTFE (12 mm × 12 mm, 7 mm × 7 mm, and 5 mm × 5 mm) each at 3 different depths (0.63 mm, 1.46 mm, and 2.08 mm), and only one size of steel chip (5 mm × 5 mm) located at 0.63 mm, 1.46 mm, and 2.08 mm. The bottom three defects are steel chips defects and the rest are PTFE defects. The height of the defects was measured with a calibrated caliber obtaining a value of



(a) Image



(b) Measurements



(c) Defects

Fig. 1. Photographs of the specimen.

0.06 mm. In Fig. 2, the location of the defects in the specimen is shown. The depths and layers of the defects are presented from shallowest to deepest, from left to right: the first column of defects has a depth of 0.63 mm, the second column 1.46 mm and the third column 2.08 mm.

A greater surface area of the defect implies a greater heat flow affected by the presence of the defect; and consequently, it implies a greater variation of temperatures in the areas near the defect. On the other hand, the shallower the depth, the lower the lateral heat dissipation effect. As a consequence of the presence of a defect, the

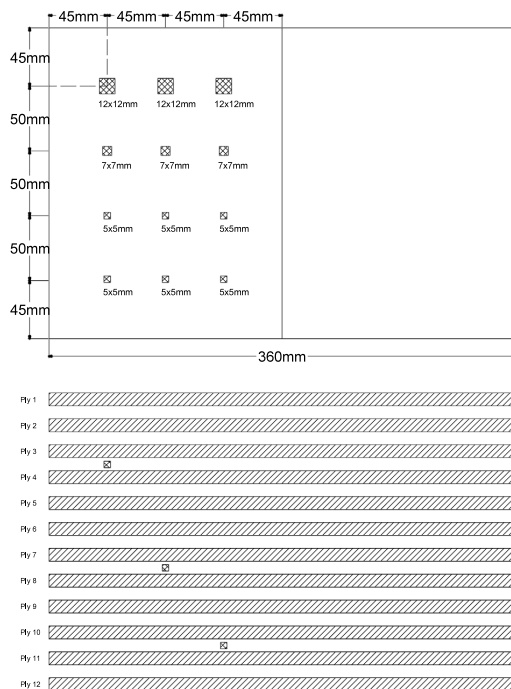


Fig. 2. Arrangement of the defects and dimensions of the specimen.

heat flow towards the surface will be less degraded, making the thermal effect on the surface more evident.

### 2.2. Infrared thermography using step heating

The CFRP laminate is heated using two halogen lamps (eurolite PAR-64 Profi floorspot model of 1 000 W) for ten seconds. After the ten seconds, the two lamps are turned off to let the object cool down for another ten seconds. This process is recorded (using an IR detector NETD of less than 55 mK, and optics of 25 mm F/1 lenses) for a total of twenty seconds at 50 FPS at a resolution of 640 × 480 pixels, resulting in a total of 1,000 frames for each digital recording. The camera used to record the digital recordings is a Xenix Gobi 640 GigE model with a spectral range between 8-14 μm and a pixel resolution of 480 × 640. In Fig. 3 a diagram of the setup for the recordings with the location, distance and angle of the infrared camera, halogen lamps, and specimen is shown.

The heating time necessary to reveal the presence of defects was roughly defined by numerical simulation in a preliminary stage, and subsequently, the definite heating time was verified by experimental assessment. This time span produced the maximum number of defects to be detected preventing the sample from overheating.

Subsurface defects heat and cool down at different rates than the rest of the object. The active stimulation is applied to exploit this feature as a way to obtain the maximum possible contrast between the defects and the rest of the object. For each digital recording, the CFRP laminate is cooled down to room temperature before the process starts, to avoid heating the object at different temperatures.

Fig. 4 shows this heating and cooling process for a pixel with defect and a nearby pixel without defect (see Fig. 5). In addition, these same reference points are added but with the specimen rotated 120° (see Fig. 6). No absolute values are needed, only the differences between



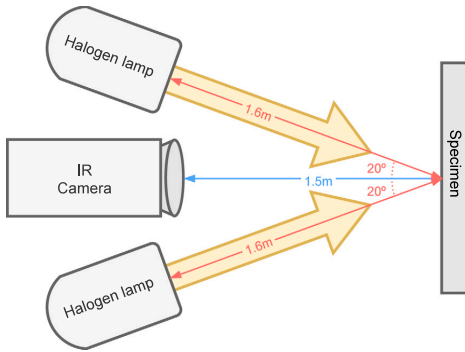


Fig. 3. Diagram of the setup of the recordings.

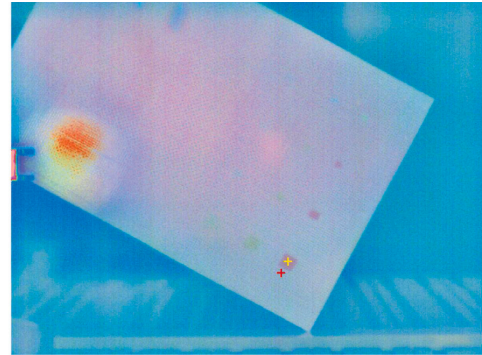


Fig. 6. Pixels used for reference in Fig. 4 for the specimen rotated 120°.

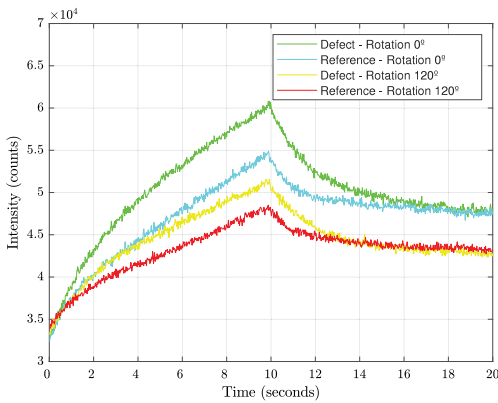


Fig. 4. Heating and cooling signal intensities for the time sequence of the CRFP laminate. Signal color correspond to those of Figs. 5 and 6.

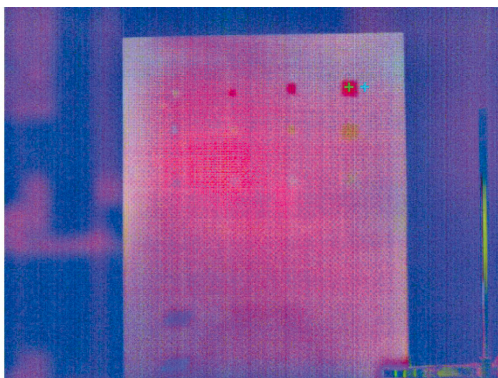


Fig. 5. Pixels used for reference in Fig. 4 for the specimen rotated 0°.

nearly pixels are required to locate the defects. When rotating the specimen it can be observed that the response is not the same. This is because the thermal energy is not transmitted uniformly throughout the specimen, which is of great interest in order to create a dataset that allows the network to generalize.

### 2.3. Post-processing methods

Image post-processing methods are used to summarize the information of a full digital recording into a particular multichannel image. This data compression is necessary to be able to use UNet and DeepLabV3+ due to their computational cost. To study the effect of this compression, two different approaches are evaluated.

The first approach converts the heating process of the digital recording into an image with 3 different channels using only the Principal Component Thermography (PCT) [27] method. This approach is tested with all the methods (Random Forest, Support Vector Machines, UNet, and DeepLabV3+).

The second approach takes advantage of both the heating and cooling sequences and uses 15 channels for each, resulting in images with 30-channel. Each channel stores post-processed images generated by the following methods: PCT [28], PPT [29,30], Kurtosis [31], Skewness [32] and TSR [33,34], as detailed in their respective subsections. This approach can only be tested with UNet.

#### 2.3.1. Principal Component Thermography (PCT)

PCT is applied to each pixel time history, calculating a linear transformation to the initial data from the eigenvectors of the associated covariance matrix. Using this method a distinction between defect and non-defect is more easily visible.

In this study, for the 3-channel images, each post-processed image corresponds to components 1st, 3rd and 4th of the PCT of the heating sequence (500 frames). The second component is not used for the 3-channel images since the signal to noise ratio is higher in the 3rd and 4th components [23].

For the 30-channel images, the first four channels of the PCT are used in both the heating (500 frames) and cooling (500 frames) sequences separately, thus giving a total of eight channels.

#### 2.3.2. Pulsed Phase Thermography (PPT)

PPT is a method to calculate the phase of thermographic data per pixel time history based in the Discrete Fourier Transform (DFT) algorithm [35]. The DFT algorithm is usually used in image post-processing to filter out periodic noise. It can be used to obtain an image that only represents the edges. For the 30-channel images, the phase of the minimum frequency is used, obtaining a post-processed image for the heating process and another for the cooling process. Eq. (1) is used to calculate PPT, where  $T$  is the temperature,  $n$  the frequency increment,  $N$  the number of frames,  $i$  the imaginary number,  $Re_n$  is the real part of the DFT, and  $Im_n$  the imaginary one.

$$F_n = \sum_{i=1}^{N-1} T(i) e^{\frac{2\pi i n i}{N}} = Re_n + i Im_n \quad (1)$$

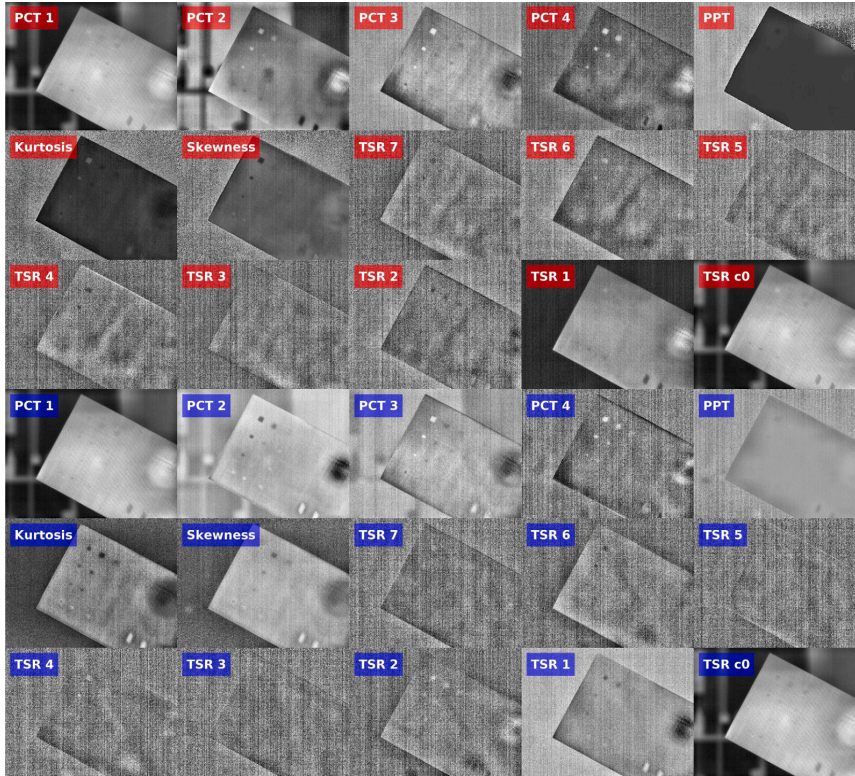


Fig. 7. Example of a 30-channel images. Images with a red label are obtained from the heating sequence. Images with a blue label are obtained from the cooling sequence..

Eq. (2) is used to calculate the phase.

$$\phi = \text{atan} \left( \frac{Im_n}{Re_n} \right) \quad (2)$$

### 2.3.3. Kurtosis

Kurtosis measures the degree of peakedness of a distribution. If the distribution is the same as the normal distribution it has a value of zero, if it is higher it has a positive value, and if it is lower a negative value. In this case this measure is calculated per pixel time history using the heating and cooling sequences, obtaining two channels for the 30-channel images.

Eq. (3) is used to calculate Kurtosis.  $T$  is the temperature data from the pixel time history,  $\bar{T}$  is the mean of the temperature data,  $s$  the standard deviation, and  $N$  the number of frames.

$$Kurtosis = \frac{\sum_{i=1}^N (T_i - \bar{T})^4 / N}{s^4} \quad (3)$$

### 2.3.4. Skewness

Skewness measures the lack of symmetry. A positive skew means that the longest tail of the distribution is at the right of the histogram and the reverse for the negative skew. A distribution that is fully symmetric has a value of zero. The skewness is calculated per pixel using every frame in the heating process or cooling process. For the 30-channel images, this results in two channels, one for the heating process and another for the cooling process.

Eq. (4) is used to calculate Skewness.  $T$  is the temperature data from the pixel time history,  $\bar{T}$  is the mean of the temperature data,  $s$  the standard deviation, and  $N$  the number of frames.

$$Skewness = \frac{\sum_{i=1}^N (T_i - \bar{T})^3 / N}{s^3} \quad (4)$$

### 2.3.5. Polynomial fit

Polynomial fit, also known as Thermographic Signal Reconstruction (TSR) when calculated using logarithmic expressions, is a method for estimating thermal diffusivity by removing noise from a thermal signal based on a sequence. This method is calculated per pixel time history and is commonly used for defect detection. It is considered that a degree of 7 generally provides optimal results for defect detection in laminates [36]. This generates one post-processed image for the coefficients of each degree plus its coefficient zero. Taking this into consideration, for the 30-channel images, eight channels for the heating process and another eight channels for the cooling process are created.

Eq. (5) is used to calculate the Polynomial fit.  $T$  is the temperature pixel time history,  $n$  is the degree and  $t$  is the time or frame of the thermogram.

$$T(t) = a_0 + a_1 t + a_2 t^2 + \dots + a_n t^n \quad (5)$$

## 2.4. Dataset

Using the method described in Section 2.2, 36 digital recordings are generated. All the digital recordings record the same CFRP laminate using different rotations, which alters lighting, lamp reflections and background among other things. This process is done to obtain more data for training and to improve variability. For each digital recording the CFRP laminate is rotated  $10^\circ$ . From each digital recording, two images (one with 3 channels and the other one with 30 channels) are generated using the post-processing methods mentioned in Section 2.3. In Fig. 7 an example of every post-processed image from one of the 30-channel images is shown. The objective of this study is not visualization

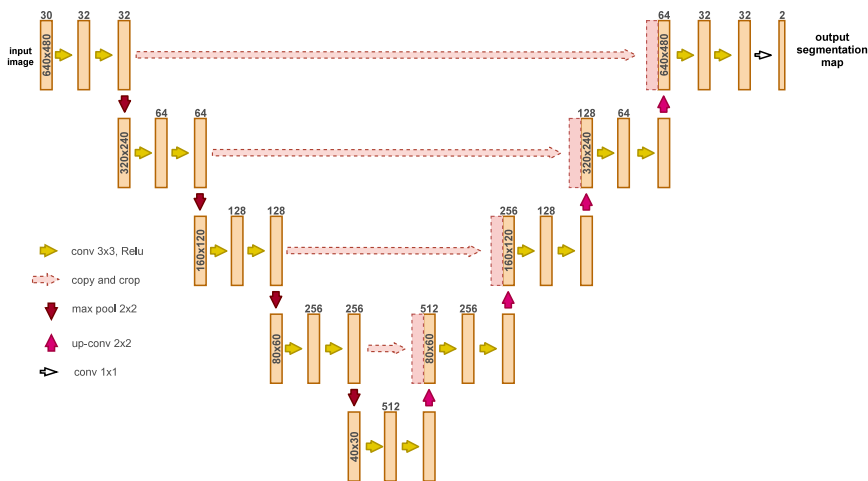


Fig. 8. The UNet architecture used for optimal results. (This graphic is inspired from the UNet architecture paper [14]).

but detection and localization. Fig. 7 merely provides an understanding of the inputs to be fed into the neural networks.

Ground truth masks are generated by experts in the field, who verify that the defects are correctly classified.

Two datasets are created, one that uses images with 3 channels and another that uses images with 30 channels. In this way, a comparison can be made to determine if more information results in better accuracy. Each dataset consists of 36 images. The first 30 are used for training and the last 6 for testing and visualization. Both datasets can be found at the following DOI: <https://doi.org/10.5281/zenodo.5426792>.

Classes are divided in “defect” and “other”. The objective is binary classification so the “defect” class is the target class, and the “other” class is the non-target class that refers to everything else, including the rest of the specimen and the background of the digital recording.

### 2.5. Analysis of the evaluated architectures

#### 2.5.1. Unet

UNet is one of the first and most referenced networks in semantic segmentation with over 29,000 cites of its original paper in Google Scholar. Its original purpose was for binary classification to segment cells in biomedical imagery and to train and produce precise predictions with as few training images as possible [14]. The name “UNet” comes from its u-shaped architecture as the result of a symmetric encoder–decoder. UNet was quickly adapted to work with all kinds of imagery and class number as it offers a high degree of flexibility thanks to its simple layout. This has caused the rapid development of new variations. An overview of the UNet architecture used in this evaluation study can be seen in Fig. 8.

#### 2.5.2. Deeplab

DeepLab is a semantic segmentation architecture made by Google. DeepLabV1 [37] presents atrous convolutions to tune the resolution at which features are calculated. DeepLabV2 [38] details Atrous Spatial Pyramid Pooling (known as ASPP) to increase the accuracy of predictions at different scales. DeepLabV3 [39], tunes the ASPP module and uses a Batch Normalization module to simplify the setup of the data eliminating the need for a manual normalization. DeepLabV3+ [15] is the fourth and most recent version of DeepLab. It converts its architecture to an encoder–decoder architecture. There is an auto machine learning version called AutoDeepLab [40] which is based on the DeepLabV3+ architecture. An overview of the DeepLabV3+ architecture used in this work can be seen in Fig. 9. The DCNN module is the

backbone network used and it usually is a variation of ResNet, Xception or MobileNet.

### 2.6. Network parameters

This section provides a brief description of the network parameters used to modify the architectures.

UNet and DeepLabV3+ have some common network parameters: the input size, which controls the resolution and channels of the input images, the number of classes to use in the experiment, and the use of padding to fill each convolution to keep the resolution of the final feature map the same size as the input.

UNet has two controllable network specific parameters consisting of the depth of the architecture, which is based on the number of max pooling layers, and the number of filters at each level, which is controlled by the number of filters at the first level and then multiplied by two at each level.

DeepLabV3+ has a specific controllable network parameter called output stride. This parameter controls the separation between each step of a convolution. It is calculated as the result of the division between the input image resolution and the final feature map. For example, an input image that has a resolution of  $512 \times 256$  pixels and a final feature map of  $32 \times 16$  would result in an output stride of 16.

In all cases, the initialization of the convolutional filter weights follows the Kaiming He et al. [41] algorithm.

### 2.7. Training parameters

This section lists the training parameters used to train the models. Optimal training parameters are re-evaluated for every change in the network parameters described in Section 2.6.

As a first step, the optimal batch size, learning rate and number of epochs are investigated. This process is repeated for each solving algorithm available, which in this case are Adam or Stochastic Gradient Descent with Momentum (SGDM). On the other hand, the value of the L2 regularization is studied separately, to apply a penalty to the loss function in order to decrease the complexity of the model and reduce overfitting.

Then, if the Precision and Recall metrics are unbalanced, different class balancing approaches are evaluated. Methods such as inverse frequency weighting (IFW), mean frequency weighting (MFW) and manually chosen custom weights are evaluated.

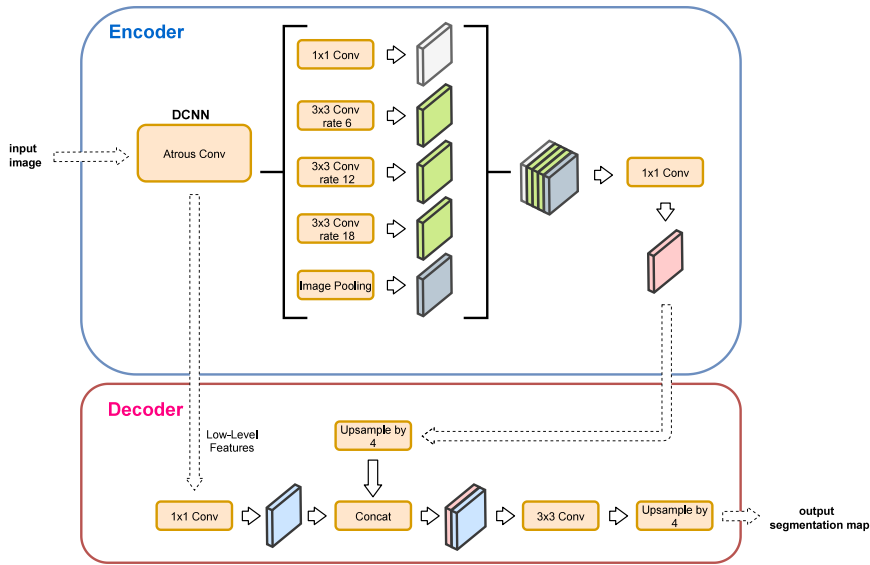


Fig. 9. The DeepLabV3+ architecture used in the experiments. This graphic is inspired from the DeepLabV3+ paper [15].

In this work, the use of a gradient clipping value, to constrain the maximum possible value of the gradient, is not necessary, since the exploding gradient problem is not present in the training process.

Finally, to add new data samples, improve variability and reduce overfitting, data augmentation methods are applied to the training set. These methods consist of enlarging or flipping the images. In addition, the dataset is shuffled before each epoch to minimize overfitting.

## 2.8. Performance metrics

This section provides a brief description of the metrics used [42] to evaluate the performance of the trained models.

- True positive (TP): correctly classified pixels.
- True negative (TN): pixels correctly classified as belonging to other classes.
- False positive (FP): pixels classified wrongly as the target class.
- False negative (FN): pixels wrongly classified as belonging to other classes.
- Precision (P): Percentage of correctly classified pixels from the total number of predictions for a particular class.

$$P = \frac{TP}{TP + FP} \quad (6)$$

- Recall (R): Percentage of correctly classified pixels from the total number of pixels for a particular class.

$$R = \frac{TP}{TP + FN} \quad (7)$$

- F-score ( $F_1$ ): Value that combines Precision and Recall making it easier to compare models. A good model should have a balance between Precision and Recall. This metric should not be used alone as it does not indicate whether the two metrics are balanced. This metric is equivalent to the Dice Coefficient with two classes.

$$F_1 = \frac{2 \times P \times R}{P + R} \quad (8)$$

- Intersection-Over-Union (IoU): Value that measures the similarity between ground truth and prediction. This metric is equivalent to

the Jaccard Index.

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{TP}{TP + FN + FP} \quad (9)$$

## 2.9. Training procedure

The network and training parameters must be tuned to reach the best possible results. In this evaluation, these hyperparameters are calibrated manually for each network, obtaining the best configuration for each parameter one by one. The effects of changing multiple parameters at the same time has not been studied in depth. However, a manual process that would research every combination of parameters is not feasible. In this regard, there is still leeway to improve results but the time required is far too great for a small improvement in accuracy.

To obtain realistic results, the datasets are divided in training and testing images. From the total of 36 images, the first 30 are used for training and the last 6 for testing and visualization.

To select the best experiments both Precision and Recall are evaluated. When both metrics are high and are balanced it is considered as a good result. If they are unbalanced, the accuracy of the model is compromised. A high Precision and a low Recall means that the model is predicting few pixels but those that are predicted are correct. If the Recall is high and the Precision low, it means that the model is predicting more pixels than there are in the ground truth. Only the metrics of the target class are provided because the non-target class is irrelevant.

To offer a better representation and facilitate the understanding of the metrics, visualization examples of the six testing images are provided for the best experiment of each architecture. This can help to give a better idea of how the model is predicting the defects.

The hardware used to train the models of the experiments consist of a GPU NVIDIA RTX 2080 Ti and a I7-9700 K CPU.

## 3. Results and discussions

### 3.1. Random forest and support vector machines

Random Forest runs several decision tree algorithms. Each decision tree gives a classification and the choice with the most “votes” is the

**Table 1**  
Metrics for the experiments with Random Forest and Support Vector Machines.

Method	Precision	Recall	IoU	F <sub>1</sub>
RF	.103	.132	.061	.116
SVM	.037	.604	.036	.069

final prediction. Support Vector Machines search for a hyperplane with the widest margin between the two classes that best separates two different classes of data points.

Experiments are carried out with Random Forest and Support Vector Machines as a basis for making comparisons. Both use the same feature vector, which is calculated using thirteen features. The first three features consist of the red, green and blue (RGB) values of each pixel of the input image, which correspond to the first, third and fourth components of PCT.

The fourth feature is the local binary pattern (LBP), a texture descriptor used in computer vision, calculated by thresholding the neighborhood of every pixel into a binary number using a  $3 \times 3$  grid, and converting the result to a decimal number [43]. To calculate the neighborhood components, a radius of 24 is used totaling in 192 neighbors. LBP is applied to a gray scale version of the 3-channel images in order to obtain more spatial context. LBP is calculated with Eq. (10), where  $P$  is the total number of neighbors,  $R$  the radius,  $c$  is to the central pixel and  $g$  is the value of a pixel.

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)^{2^p} \text{ with } s(x) = \begin{cases} 1, & \text{if } x \geq 0; \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

The last nine features consist of multiple Haralick texture features, which are texture descriptors used in computer vision for image classification. All Haralick features are based on the gray-level co-occurrence matrix which shows the frequency at which each gray level occurs in a pixel at a fixed geometric location with respect to other pixels. The features used are: angular second moment, contrast, correlation, sum of square: variance, inverse difference moment, sum average, sum entropy, and entropy. Equations for all the Haralick texture features are in [44]. Haralick features are applied to the gray scale version of the 3-channel images in order to obtain more spatial context.

A subsampling to the feature vector of each image is done in order to reduce training time and memory usage. This generates 1,000 observations per image with 13 features per observation. With 30 images to train, 30,000 observations are used for training.

For Random Forest the number of estimators and their maximum depth is manually optimized. In addition, different class weights are tested for both Random Forest and Support Vector Machines. The optimal experiments for each methods are listed in Table 1. The best Random Forest experiment uses 1000 estimators and a maximum depth of 10. In the case of Support Vector Machines a radial basis function kernel is used, and the gamma value is calculated as the inverse of the multiplication of the number of features by the variance. In both cases the class weight for the non-target class and for the target or defect class is balanced using the proportional inverse of the class frequencies. Results from these experiments can be seen in Table 1.

According to Table 1, both experiments obtain low metrics: below 12% in F<sub>1</sub>-Score. To prove that these values are too low, visualizations of Random Forest and Support Vector Machines are shown in Figs. 10 and 11 respectively.

In SVM, the edges of the specimen are classified as defects, this is due to the great variance between the specimen and the background. In RF, although this can be observed in some cases, it is much less obvious. Moreover, both models predict many more pixels from the most shallow defects as these are the ones with the most variance.

There is a circular area detected at the bottom of both RF and SVM. This area is the reflection of the heating lamps. By observing Fig. 11, it is clear that SVM is very sensitive to these artifacts, more

**Table 2**  
Network parameters for UNet.

Network parameters		
Parameter	3-channel images	30-channel images
Input size	640 × 480 × 3	640 × 480 × 30
Classes	2	2
Depth	4	4
Filters on first level	32	32
Padding	Yes	Yes

**Table 3**  
Training parameters for UNet.

Training parameters		
Parameter	3-channel images	30-channel images
Solver	Adam	Adam
Epochs	1000	1000
Batch size	8	4
Learning rate	0.001	0.001
Class weighting	0.35–0.65	0.20–0.80
Gradient clipping	No	No
L2 regularization	0.0001	0.0001
Data augmentation	Mirror in X/Y	Mirror in X/Y
Shuffle	Yes	Yes

**Table 4**  
Metrics for the experiments with UNet.

Experiment	Precision	Recall	IoU	F <sub>1</sub>
3-channel images	.689	.717	.542	.703
30-channel images	.764	.726	.593	.745

so than RF. These reflections can be avoided by positioning the camera properly, although this is not always possible in real inspections due to lack of space. It is very common to find reflections in real inspections. Therefore, it seems reasonable to include them in the study and analyze the robustness of the models in their presence. Although the effects can be minimized using Plexiglas filters.

Theoretically, better results could be achieved by improving the feature vector. The selection of features has the most significant impact on how well these methods perform. However, in this study, common features for image segmentation are used [43,44].

### 3.2. UNet

This section presents the optimal segmentation results with UNet for 3-channel images and 30-channel images. Both experiments have the same optimal hyperparameters ( Tables 2 and 3) with the exception of batch size and class weights. Since the images with 3 channels take less memory than the images with 30 channels, the maximum batch size can be increased from four to eight images. In the case of the class weights, the optimal weights differ between datasets from a value of 0.65 to 0.80.

The depth of the UNet architecture coincides with the original implementation but the number of filters on the first level has been reduced by two. This affects the whole architecture dividing the numbers of filters by two. Using fewer filters means faster training times and increased batch sizes. There is no need for gradient clipping since there is no exploding gradient problem. L2 regularization works best when using the 0.0001 default. All the training data is shuffled before every epoch to prevent overfitting.

The metrics from the testing of the “30-channel images” and “3-channel images” experiments can be seen in Table 4.

Table 4 shows a great difference between using 3 and 30 channels. In this case the 30-channel images experiment has an almost 5% higher F<sub>1</sub>-Score. Both experiments surpass 70% in F<sub>1</sub>-Score and obtain a balance between Precision and Recall.

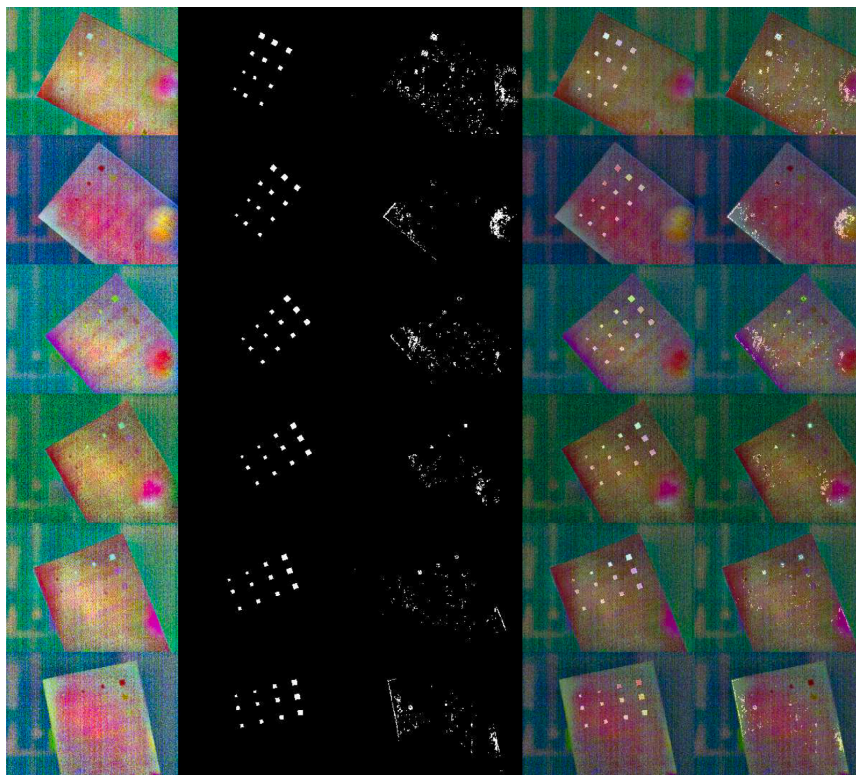


Fig. 10. Visualization of the predicted results for Random Forest. (1st col.) Original images, (2nd col.) ground truth masks, (3rd col.) predictions, (4th col.) original images and ground truth masks, (5th col.) original images with predictions.

The 3-channel images experiment takes 00 h:31 m:31 s to train with the specified hardware, whereas the 30-channel images experiment takes 02 h:15 m:15 s. The extra channels make the architecture more complex.

To accompany these results a visualization of the testing images can be seen in Figs. 12 and 13. In these figures a great difference between models can be seen. The experiment for 3-channel images (Fig. 12) detects all defects although the ones with more depth have a much smaller area than the ground truth and there are some false positives. Fig. 13 has much less noise but it has trouble detecting all the defects in some of the images.

### 3.3. DeepLabV3+

This section presents the best experiment with DeepLabV3+ with 3-channel images. Table 5 shows the network architecture parameters. In this case the backbone architecture that performs the best is Xception65. Xception71 has more layers and therefore more VRAM is needed for the same batch size. A smaller batch size, even when using a network with more layers, performs worse. For this same reason, an output stride of 16 is preferred.

Table 6 shows the training parameters. In this case, DeepLabV3+ has a more complex architecture than UNet so the maximum batch size possible for eleven gigabytes of VRAM is four images. The value of the optimal class weight for the 3-channel images is the same as UNet. There is no need for gradient clipping since there is no exploding gradient problem. Furthermore, this architecture works better with a smaller learning rate than UNet. The best L2 regularization value coincides with

Table 5

Network parameters for DeepLabV3+.	
Network parameters	
Input size	640 × 480 × 3
Classes	2
Backbone	Xception65
Output stride	16
Padding	Yes

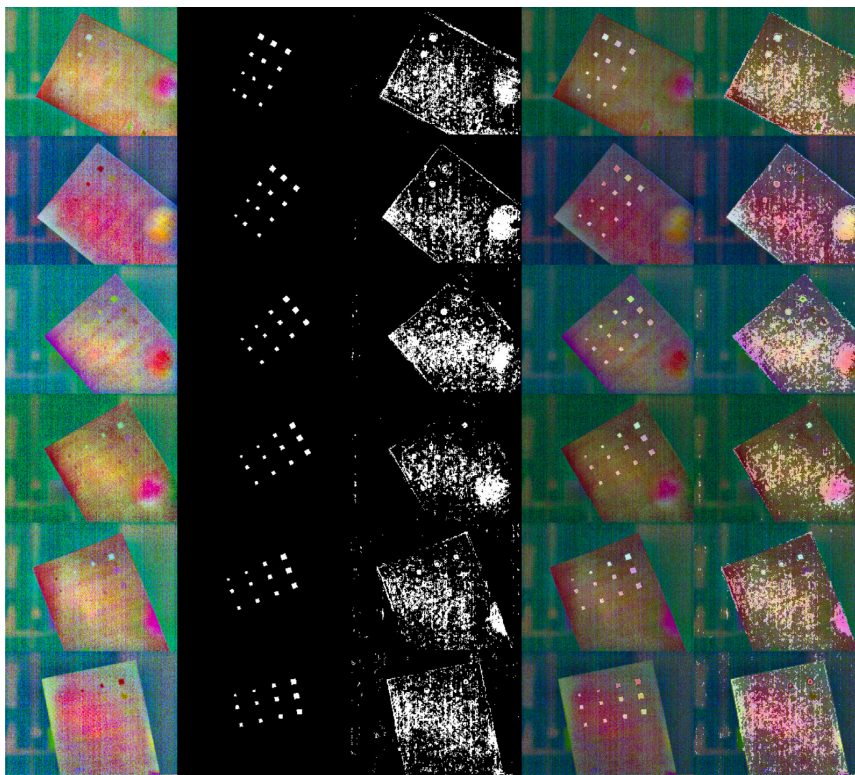
Table 6

Training parameters for DeepLabV3+.	
Training parameters	
Solver	Adam
Epochs	1000
Batch size	4
Learning rate	0.0005
Class weighting	0.35–0.65
Gradient clipping	No
L2 regularization	0.00004
Data augmentation	Scale 0.5–2.0 with 0.25 steps
Shuffle	Yes

that recommended by the developers. The whole training set is shuffled before every epoch to prevent overfitting.

To achieve faster training times and allow the model to generalize better, the training starts from a pre-trained model on the ImageNet dataset [45].

The metrics from the testing of the “3-channel images” experiment can be seen in Table 7.



**Fig. 11.** Visualization of the predicted results for Support Vector Machines. (1st col.) Original images, (2nd col.) ground truth masks, (3rd col.) predictions, (4th col.) original images with ground truth masks, (5th col.) original images with predictions.

**Table 7**  
Metrics for the experiment with DeepLabV3+.

Experiment	Precision	Recall	IoU	F <sub>1</sub>
3-channel images	.760	.786	.629	.773

In Table 7 the results show high values for the metrics, above 77% in F<sub>1</sub>-Score and with Precision and Recall balanced. This experiment obtains even better results than the 30-channel images experiment with UNet, which is impressive given the difference between the 30-channel images and 3-channel images experiments in UNet.

This experiment with DeepLabV3+ takes 01 h:18 m:27 s, more than 2.5 times longer than UNet under the same conditions. However, is still almost two times faster than the 30-channel experiment with UNet.

To accompany these results a visualization of the testing images can be seen in Fig. 14. In these figures a great difference between models is observed with respect to those of UNet. This model detects almost every defect and has virtually no noise. It has most trouble detecting 5 mm×5 mm defects at maximum depth. However, in the majority of the testing images all the defects are found.

#### 3.4. Discussion

Neither Random Forest nor Support Vector Machines can detect defects in CFRP laminates using the image post-processing methods described. The metrics (Table 8 and Fig. 15) and visualization images (Figs. 10, 11, 12, 13 and 14) make it clear that these methods are not reliable enough, at least with the features selected, to detect defects with

high confidence. They do not generalize well enough. Thermographic data generally has high levels of noise and low levels of contrast. These characteristics give high variance to the features for the same defect, making them difficult to detect for conventional models such as RF and SVM.

In the case of UNet, results are much improved. With the 3-channel images the metrics might be considered low. However, the defects are all distinguishable in the visualization images although there is some noise in the predictions. When it comes to the 30-channel images, the result metrics show a clear improvement. The noise of predictions is vastly reduced and the visualization images show that almost all the defects are found.

DeepLabV3+ performs better than UNet even when only the 3-channel images can be used. This evaluation provides the best results, nearing 80% of F<sub>1</sub>-Score. The visualization images are clearer than those produced by UNet and almost all of the defects are found. DeepLabV3+ is more computationally complex than UNet under the same conditions, requiring more than twice as much training time. However, DeepLabV3+ is still almost twice as fast as UNet with 30 channels.

For semantic segmentation models, unlike SVM and RF models, lamp reflection is not classified as a defect. This is desirable in real inspections, where reflections are often unavoidable. This indicates that the manually created features are not enough to learn that the reflection is not a distinguishing feature of the defects. However, UNet and DeepLabV3+ are able to “learn” that the reflections are not a distinct part of the defects. This is possible because by rotating the specimen, the reflection is not always in the same part of the specimen.

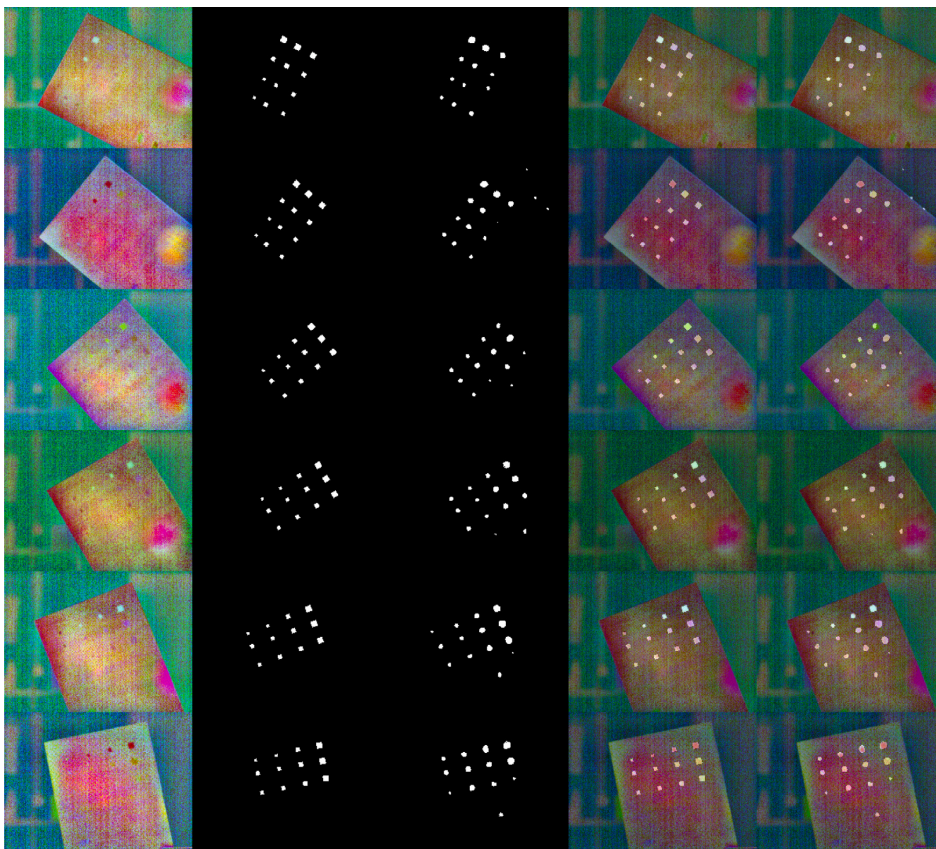


Fig. 12. Visualization of the predicted results for UNet evaluated with 3 channels. (1st col.) Original images, (2nd col.) ground truth masks, (3rd col.) predictions, (4th col.) original images with ground truth masks, (5th col.) original images with predictions.

Table 8

Metrics for all the methods.

Experiment	Precision	Recall	IoU	F <sub>1</sub>
RF (3-channel)	.103	.132	.061	.116
SVM (3-channel)	.037	.604	.036	.069
UNet (3-channel)	.689	.717	.542	.703
UNet (30-channel)	.764	.726	.593	.745
DeepLabV3+ (3-channel)	.760	.786	.629	.773

#### 4. Other samples

This section evaluates new specimens with different internal structures. The objective of these evaluations is to observe how far the semantic segmentation models generalize. For this purpose, the predictions of these specimens are run with the previous models, trained with the specimen presented in Section 2.1.

The defects of these specimens are artificially generated, however, they may be slightly offset from the original scheme. For this reason, an ultrasonic inspection is performed to find and check the real positions of the defects. The ground truth of these two new specimens is generated by a manual procedure. First, a probe is passed over the surface of the specimen, scanning the signal it receives in a similar way to an oscilloscope. In this way, it is possible to detect signal changes that are indicative of a defect. This defective area is marked with a pencil on the specimen itself. Finally, using the thermographic image and the

Table 9

Metrics for specimen 2.

Experiment	Precision	Recall
UNet (3-channel images)	.56	.47
DeepLabV3+ (3-channel images)	.83	.41

RGB image in which the pencil marks can be observed, a ground truth mask is generated manually by observing and overlapping both images.

The first specimen has a similar structure to the training specimen. However, this specimen has half the depth (1.125 mm) and a smaller number of layers (6 plies). In this case, the depth of the defects is 0.75 mm, 0.56 mm, and 0.19 mm from left to right. The top three defects are steel chips defects and the rest are PTFE defects. (See Fig. 16).

As can be seen in Fig. 17, all the defects of the part are successfully detected with DeepLabV3+. It appears that the smaller defects have a predicted area greater than the area of the ground truth defects. UNet is also able to detect almost all of the defect but the predicted image has more noise. In Table 9 metrics for this evaluation are obtained. These metrics present lower precision than expected due to this increase in the area of small defects.

The second specimen has a very different structure from the training specimen. This specimen not only has greater depth (20 plies and a total depth of 3.825 mm), but also, ply 7 is of greater depth and



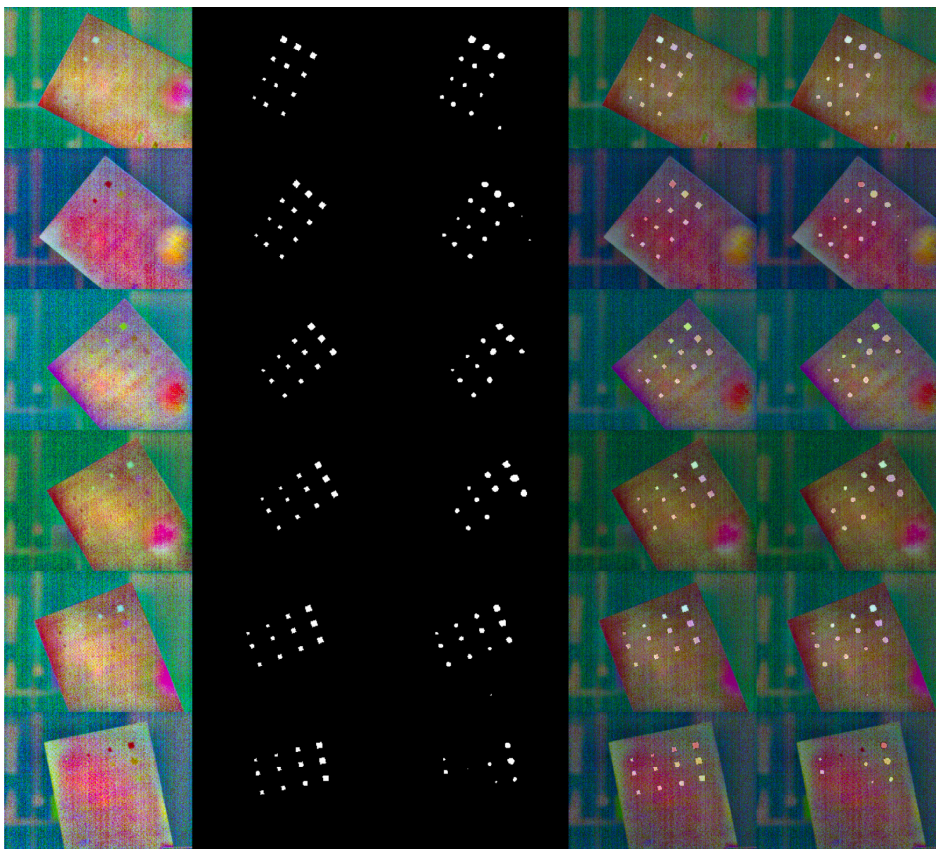


Fig. 13. Visualization of the predicted results for UNet evaluated with 30 channels. Only the first three channels are shown in the image, which consist of the first, third and fourth components, using exactly the same channels as the 3-channel images. (1st col.) Original images, (2nd col.) ground truth masks, (3rd col.) predictions, (4th col.) original images and ground truth masks, (5th col.) original images with predictions.

Table 10

Metrics for specimen 3.

Experiment	Precision	Recall
UNet (3-channel images)	.56	.50
DeepLabV3+ (3-channel images)	.84	.32

reflectivity (See Fig. 18). This ply is 20 mm thick and is called “Rohacell core” and is a registered trademark of structural foams that have high mechanical performance (<https://www.rohacell.com/en>). These foams have been used in the aeronautical sector for a long time to lighten composite materials and are currently used for the same purpose in other industries, such as the automotive and wind sectors.

There are no defects deeper than that of the Rohacell core because with thermography it is not possible to detect defects due to the fact that it is a great thermal insulator. The depth of the defects are 0.38 mm, 0.75 mm, 1.125 mm from left to right. The bottom three defects are steel chips defects and the rest are PTFE defects.

As can be seen in Fig. 19, most of the defects are not detected successfully. It seems that the new layer aggressively affects the reflectivity and therefore the behavior of the model for defect detection. In Table 10 the metrics for this evaluation are obtained. These metrics obviously present very poor results.

As a result of these evaluations, it can be observed that as long as the tested specimen has a similar structure to that of the training specimen,

high quality detections can be achieved even if the depth of the specimen is not exactly the same as in the training specimen. However, if the specimen structure is severely altered, by adding an inner layer with different reflectivity, or very drastic depth changes, the semantic segmentation models are not able to find all the defects in the specimen. In particular, the Rohacell core changes the boundary conditions of the heat transfer problem, which affects the results obtained in the inspections.

### 5. Conclusion

This paper studies different solutions from the computer vision branch for pixel-based defect detection in CFRP specimens. It evaluates older and more common models such as Random Forest and Support Vector Machines against state-of-the-art approaches such as convolutional neural networks for semantic segmentation.

Semantic segmentation networks are capable of detecting subsurface defects far outperforming older methods such as Random Forest or Support Vector Machines. In addition, semantic segmentation has a great advantage over object detection thanks to its ability to detect defects of any shape, not only square defects.

More complex and modern networks like DeepLabV3+ tend to perform better, but increasing the amount of data per sample given to the model is almost as effective, as seen with UNet. Using 30-channel images instead of 3-channel images significantly improves

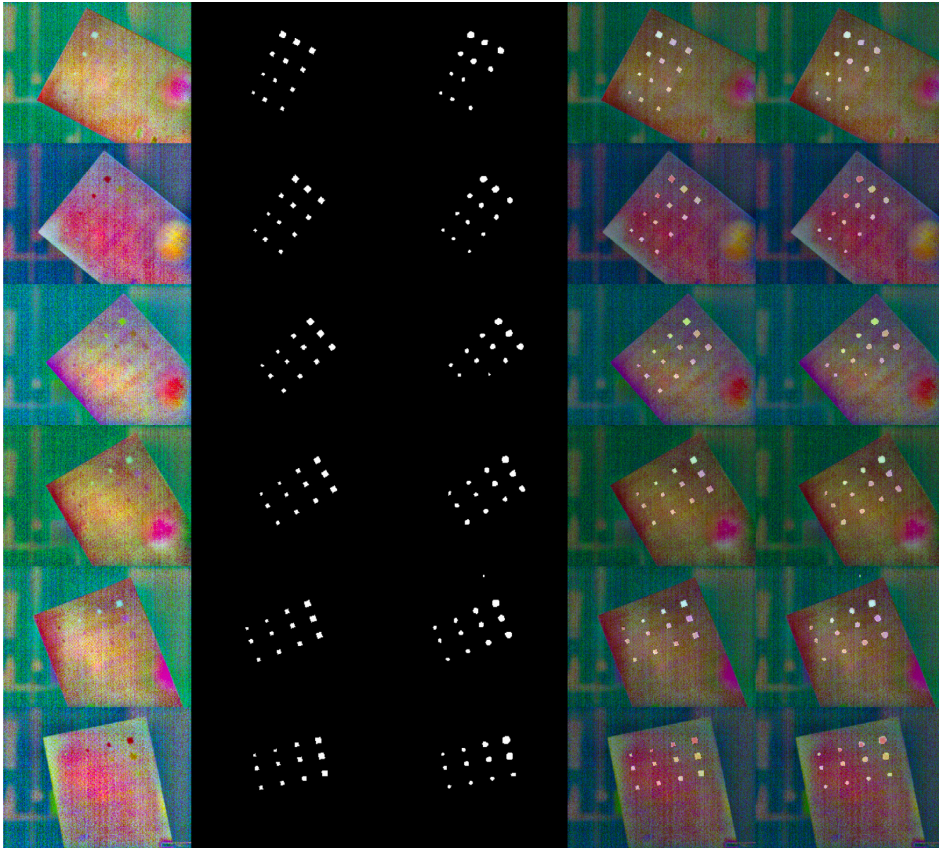


Fig. 14. Visualization of the predicted results for DeepLabV3+. (1st col.) Original images, (2nd col.) ground truth masks, (3rd col.) predictions, (4th col.) original images with ground truth masks, (5th col.) original images with predictions.

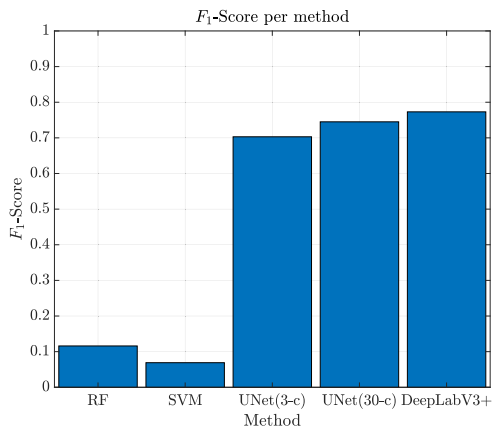


Fig. 15. Bar graph for all the methods.

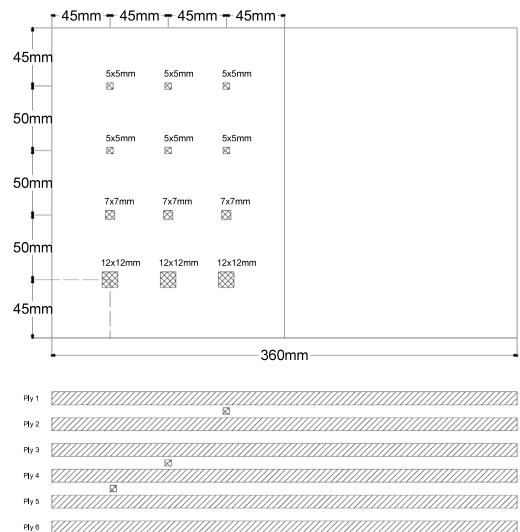


Fig. 16. Diagram of specimen 2.

predictability. To ensure reproducibility and further investigations, the dataset generated for this article is publicly available in the following DOI: <https://doi.org/10.5281/zenodo.5426792>.

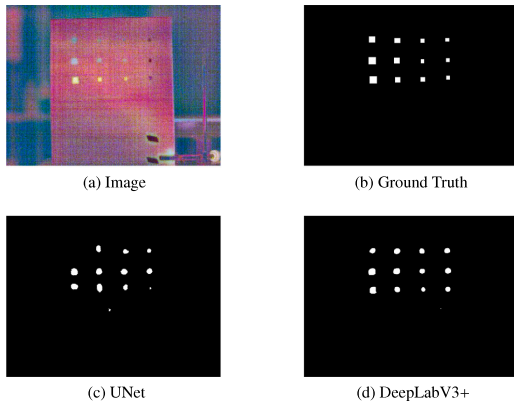


Fig. 17. Specimen 2.

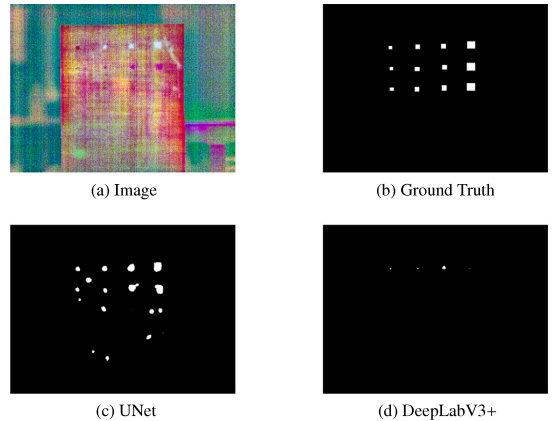


Fig. 19. Specimen 3.

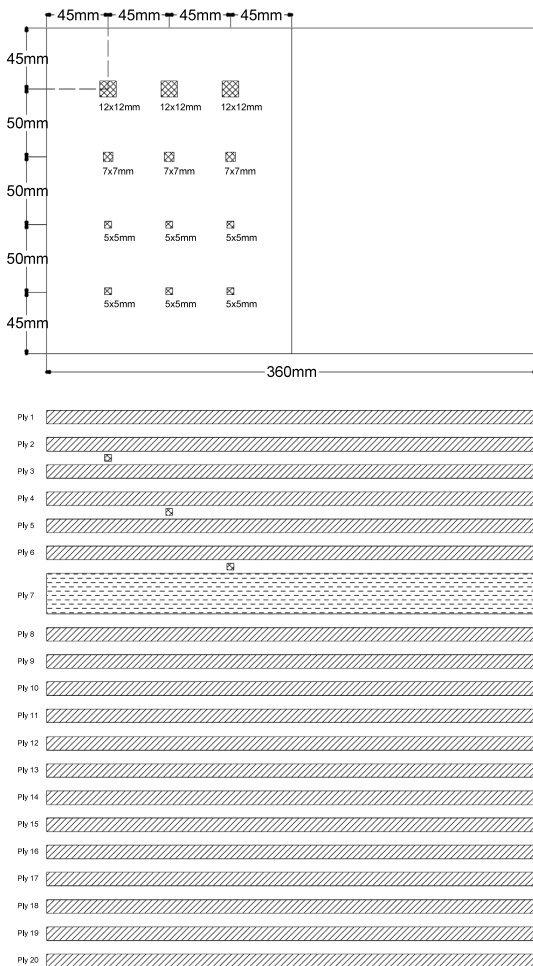


Fig. 18. Diagram of specimen 3.

To increase the validity of this study, evaluations of the DeepLabV3+ and UNet trained models for the 3-channel dataset are performed on new specimens with different internal structures. This evaluation proves that as long as the specimen has a similar internal structure, defect detection with strong results is possible.

Without performing testing on composite specimens with naturally occurring flaws, it is not possible to validate this technique. Almost all defects are easily detected and without false positives in this test for artificially induced defects with DeepLabV3+. The structure of the specimens needs to be similar to the training samples. A larger and more varied dataset would produce improved results.

It is apparent that these technologies could provide a solid support to help experts who have to check each specimen manually. Considering how fast the field of computer vision is evolving, it would be no surprise if deep learning algorithms become the norm for subsurface defect detection.

This study shows that there is still room for improvement in this field. For example, a GPU with more than eleven gigabytes of VRAM could slightly improve the results offered in this evaluation work by increasing the batch size. In addition, if the architecture of DeepLabV3+ were modified to accept 30-channel images, it could improve its results, although this would further limit the VRAM required.

**CRedit authorship contribution statement**

**Oscar D. Pedrayes:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Darío G. Lema:** Investigation, Resources, Supervision. **Rubén Usamentiaga:** Investigation, Resources, Data curation, Writing – review & editing, Visualization, Supervision, Project administration, Funding acquisition. **Pablo Venegas:** Investigation, Resources, Data curation, Writing – review & editing, Visualization, Supervision. **Daniel F. García:** Investigation, Resources, Supervision, Funding acquisition.

**Declaration of competing interest**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Data availability**

Data is available in <https://doi.org/10.5281/zenodo.5426792>.

## Acknowledgments

This work has been partially funded by the project RTI2018-094849-B-I00 of the Spanish National Plan for Research, Development and Innovation, Spain.

## References

- [1] B. Kamsu-Foguem, Knowledge-based support in Non-Destructive Testing for health monitoring of aircraft structures, *Adv. Eng. Inform.* 26 (4) (2012) 859–869.
- [2] S. Gholizadeh, A review of non-destructive testing methods of composite materials, *Proc. Struct. Integr.* 1 (2016) 50–57.
- [3] B. Yousefi, D. Kalthor, R. Usamentiaga Fernández, L. Lei, C.I. Castanedo, X.P. Maldague, et al., Application of deep learning in infrared non-destructive testing, in: *QIRT 2018 Proceedings*, 2018.
- [4] Q. Luo, B. Gao, W.L. Woo, Y. Yang, Temporal and spatial deep learning network for infrared thermal defect detection, *NDT & E Int.* 108 (2019) 102164.
- [5] Q. Fang, X. Maldague, A method of defect depth estimation for simulated infrared thermography data with deep learning, *Appl. Sci.* 10 (19) (2020) 6819.
- [6] R. Marani, D. Palumbo, U. Galietti, T. D'Orazio, Deep learning for defect characterization in composite laminates inspected by step-heating thermography, *Opt. Lasers Eng.* 145 (2021) 106679.
- [7] Y. He, B. Deng, H. Wang, L. Cheng, K. Zhou, S. Cai, F. Ciampa, Infrared machine vision and infrared thermography with deep learning: a review, *Infrared Phys. Technol.* (2021) 103754.
- [8] P. Theodorakeas, E. Cheilakou, E. Ftikou, M. Kouli, Passive and active infrared thermography: An overview of applications for the inspection of mosaic structures, in: *J. Phys. Conf. Ser.*, vol. 655, IOP Publishing, 2015, 012061.
- [9] R. Usamentiaga, P. Venegas, J. Guerediaga, L. Vega, I. López, A quantitative comparison of stimulation and post-processing thermographic inspection methods applied to aeronautical carbon fibre reinforced polymer, *Quant. InfraRed Thermogr. J.* 10 (1) (2013) 55–73.
- [10] C. Ibarra-Castanedo, X. Maldague, Pulsed phase thermography reviewed, *Quant. Infrared Thermogr. J.* 1 (1) (2004) 47–70.
- [11] Y. Liu, G. Tian, B. Gao, X. Lu, H. Li, X. Chen, Y. Zhang, L. Xiong, Depth quantification of rolling contact fatigue crack using skewness of eddy current pulsed thermography in stationary and scanning modes, *NDT & E Int.* 128 (2022) 102630.
- [12] I. Sgibnev, A. Sorokin, B. Vishnyakov, Y. Vizilter, Deep semantic segmentation for the off-road autonomous driving, *Int. Archiv. Photogram. Remote Sens. Spatial Inform. Sci.* 43 (2020) 617–622.
- [13] O.D. Pedrayes, D.G. Lema, D.F. García, R. Usamentiaga, Á. Alonso, Evaluation of semantic segmentation methods for land use with spectral imaging using sentinel-2 and PNOA imagery, *Remote Sens.* 13 (12) (2021) 2292.
- [14] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2015, pp. 234–241.
- [15] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: *ECCV*, 2018, pp. 801–818.
- [16] C. Zhang, Y. Ma, *Ensemble Machine Learning: Methods and Applications*, Springer, 2012.
- [17] N. Cristianini, J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, 2000, <http://dx.doi.org/10.1017/CBO9780511801389>.
- [18] Y. Cao, Y. Dong, Y. Cao, J. Yang, M.Y. Yang, Two-stream convolutional neural network for non-destructive subsurface defect detection via similarity comparison of lock-in thermography signals, *NDT & E Int.* 112 (2020) 102246.
- [19] C. Schmidt, T. Hocke, B. Denkena, Artificial intelligence for non-destructive testing of CFRP prepreg materials, *Product. Eng.* 13 (5) (2019) 617–626.
- [20] H.-T. Bang, S. Park, H. Jeon, Defect identification in composite materials via thermography and deep learning techniques, *Compos. Struct.* 246 (2020) 112405.
- [21] Y. Dong, C. Xia, J. Yang, Y. Cao, Y. Cao, X. Li, Spatio-temporal 3D residual networks for simultaneous detection and depth estimation of CFRP subsurface defects in lock-in thermography, *IEEE Trans. Ind. Inf.* (2021).
- [22] Z. Nie, J. Xu, S. Zhang, Analysis on DeepLabV3+ performance for automatic steel defects detection, 2020, *ArXiv preprint arXiv:2004.04822*.
- [23] R. Usamentiaga, C. Ibarra-Castanedo, M. Klein, X. Maldague, J. Peeters, A. Sanchez-Beato, Nondestructive evaluation of carbon fiber bicycle frames using infrared thermography, *Sensors* 17 (11) (2017) 2679.
- [24] K. Zheng, Y.-S. Chang, K.-H. Wang, Y. Yao, Improved non-destructive testing of carbon fiber reinforced polymer (CFRP) composites using pulsed thermograph, *Polym. Test.* 46 (2015) 26–32.
- [25] D. Schumacher, N. Meyendorf, I. Hakim, U. Ewert, Defect recognition in CFRP components using various NDT methods within a smart manufacturing process, in: *AIP Conference Proceedings*, vol. 1949, AIP Publishing LLC, 2018, 020024.
- [26] R. Marani, D. Palumbo, V. Renò, U. Galietti, E. Stella, T. D'Orazio, Modeling and classification of defects in CFRP laminates by thermal non-destructive testing, *Composites B* 135 (2018) 129–141.
- [27] N. Rajic, *Principal Component Thermography*, Tech. Rep., Defence Science and Technology Organisation Victoria, Australia ..., 2002.
- [28] B. Milovanović, M. Gaši, S. Gumbarević, Principal component thermography for defect detection in concrete, *Sensors* 20 (14) (2020) 3891.
- [29] H.J. Nussbaumer, The fast Fourier transform, in: *Fast Fourier Transform and Convolution Algorithms*, Springer, 1981, pp. 80–111.
- [30] J. Bodzenta, A. Kaźmierczak, T. Kruczek, Analysis of thermograms based on FFT algorithm, *J. Physique IV* 129 (2005) 201–205.
- [31] F.J. Madruga, C. Ibarra-Castanedo, O.M. Conde, X.P. Maldague, J.M. López-Higuera, Enhanced contrast detection of subsurface defects by pulsed infrared thermography based on the fourth order statistic moment, kurtosis, in: *Thermose XXXI*, vol. 7299, International Society for Optics and Photonics, 2009, p. 72990U.
- [32] F. Madruga, C. Ibarra-Castanedo, O. Conde, J. Lopez-Higuera, X. Maldague, Automatic data processing based on the skewness statistic parameter for subsurface defect detection by active infrared thermography, in: *Proc. QIRT*, vol. 9, Citeseer, 2008, p. 6.
- [33] S. Shepard, J. Lhota, B. Rubadeux, D. Wang, T. Ahmed, Reconstruction and enhancement of active thermographic image sequences, *Opt. Eng.* 42 (2003) 1337–1342, <http://dx.doi.org/10.1117/1.1566969>.
- [34] D. Balageas, B. Chapuis, G. Deban, F. Passilly, Improvement of the detection of defects by pulse thermography thanks to the TSR approach in the case of a smart composite repair patch, *Quant. InfraRed Thermogr. J.* 7 (2) (2010) 167–187.
- [35] E.O. Brigham, *The Fast Fourier Transform and Its Applications*, Prentice-Hall, Inc., 1988.
- [36] D.L. Balageas, J.-M. Roche, F.-H. Leroy, W.-M. Liu, A.M. Gorbach, The thermographic signal reconstruction method: A powerful tool for the enhancement of transient thermographic images, *Biocybern. Biomed. Eng.* 35 (1) (2015) 1–9.
- [37] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Semantic image segmentation with deep convolutional nets and fully connected crfs, 2014, *ArXiv preprint arXiv:1412.7062*.
- [38] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, *IEEE Trans. Pattern Anal. Mach. Intell.* 40 (4) (2017) 834–848.
- [39] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, 2017, *ArXiv preprint arXiv:1706.05587*.
- [40] C. Liu, L.-C. Chen, F. Schroff, H. Adam, W. Hua, A.L. Yuille, L. Fei-Fei, Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 82–92.
- [41] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.
- [42] E. Fernandez-Moral, R. Martins, D. Wolf, P. Rives, A new metric for evaluating semantic segmentation: leveraging global and contour accuracy, in: *2018 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2018, pp. 1051–1056.
- [43] S.H. Khaleefah, S.A. Mostafa, A. Mustapha, M.F. Nasrudin, Review of local binary pattern operators in image feature extraction, *Indonesian J. Electric. Eng. Comput. Sci.* 19 (1) (2020) 23–31.
- [44] E. Miyamoto, T. Merryman, Fast Calculation of Haralick Texture Features, Human Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, USA. Japanese Restaurant Office, 2005.
- [45] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2009, pp. 248–255.

### 5.1.3. Detection and localization of fugitive emissions in industrial plants using surveillance cameras

- Pedrayes, O. D., Lema, D. G., Usamentiaga, R., & García, D. F. (2022). *Detection and localization of fugitive emissions in industrial plants using surveillance cameras*. *Computers in Industry*, 142, 103731.
- DOI: [10.1016/j.compind.2022.103731](https://doi.org/10.1016/j.compind.2022.103731)
- El índice de impacto de la revista *Computers in Industry* en 2021 fue 11.245 (Q1, 95.98 %) y el índice de impacto a 5 años, 9.613.
- Citas: 2



Contents lists available at ScienceDirect

## Computers in Industry

journal homepage: [www.elsevier.com/locate/compind](http://www.elsevier.com/locate/compind)

# Detection and localization of fugitive emissions in industrial plants using surveillance cameras

Oscar D. Pedrayes<sup>a,\*</sup>, Darío G. Lema<sup>a</sup>, Rubén Usamentiaga<sup>a</sup>, Daniel F. García<sup>a</sup><sup>a</sup> Department of Computer Science and Engineering, University of Oviedo, Campus de Viesques, Gijón 33204 Asturias, Spain

## ARTICLE INFO

## Article history:

Received 9 March 2022

Received in revised form 9 June 2022

Accepted 12 June 2022

Available online 21 June 2022

## PACS:

0000

1111

## MSC:

0000

1111

## Keywords:

Deeplab

Deep learning

Pollution

Semantic segmentation

Smoke

## ABSTRACT

Industrial plants commonly generate gas emissions that are not caused intentionally. These emissions are known as fugitive emissions. Early detection of fugitive emissions helps to find points of failure in the different processes and avoid sources of pollution, helping to reduce danger to the environment and to respect legislation. Despite the importance of the problem, there are no published solutions in the specialized literature about the location and automated detection of fugitive emissions in industrial plants. Therefore, this article proposes an effective approach based on convolutional neural networks for semantic segmentation. The proposed solution takes advantage of existing surveillance cameras to apply state-of-the-art image segmentation methods, in particular, the semantic segmentation network DeeplabV3 + . This work explores aspects such as the ability to differentiate gases like water vapor and clouds from fugitive emissions, the possibility of reusing models in different industrial plants, the differences between multi-class and binary classification, the importance of proportions in the number of images in each class, the use of weights to balance classes, the comparison of a standard size test versus a real use case test, and the feasibility of an area-based alarm system to warn of emissions. This paper describes a methodology to configure the proposed solution for a specific industrial facility.

© 2022 Published by Elsevier B.V.  
CC BY 4.0

## 1. Introduction

Fugitive emissions (Laconde, 2018) are greenhouse gas emissions that are not intentionally produced by a stack or vent. This type of emissions are much more complex, as they do not follow a stack trace, but are scattered, resulting in areas of low opacity and gaps. They are usually caused in industrial plants by the production, processing, transmission, storage and use of fuels. Other causes of fugitive emissions are uncontrolled elements such as wind stirring up accumulated dust or improperly stored products. In some cases, emissions accumulate and escape to the outside through openings other than chimneys. There are many causes of fugitive emissions, so detecting and locating these emissions is essential to finding the problem and correcting it.

Pollution prevention (Johnson, 1992; Freeman et al., 1992) is a priority if the environment is to be preserved. Fugitive emissions pollute the air, endangering the lives of people and animals living

nearby. In addition, depending on the composition of the emissions, they can contribute to the greenhouse effect. For example, methane emissions from oil and gas industries have an impact 25 times greater than that of carbon dioxide (Solomon et al., 2007).

As pollution regulation laws become more strict (Lee, 2021; Condren and Dunning, 2021), companies need to find new and innovative ways to control and prevent pollution in the most cost-effective manner in order to remain competitive.

Some of the most common low-cost air pollution sensors include:

- Electrochemical sensors (Bakker and Telting-Diaz, 2002): are based on a chemical reaction between gases in the air and the electrode in a liquid. These sensors are very sensitive to temperature and humidity variations.
- Photoionization detectors (Davenport and Adlard, 1984): ionize volatile organic compounds and measure the resulting electric current. Such sensors are more expensive and do not distinguish between gases.
- Optical particle counters (Liu et al., 1974): measure particulate matter by detecting the light scattered by the particles.

\* Corresponding author.

E-mail address: [UO251056@uniovi.es](mailto:UO251056@uniovi.es) (O.D. Pedrayes).

- Optical sensors (Bogue, 2015): can detect gas by measuring the absorption of infrared light.

The use of non-optical emission detection devices (Williams et al., 2014), have notorious disadvantages. These include the need to install sensors throughout the industrial plant, modifying the existing infrastructure. In addition, the sensors must be in close proximity to emissions in order to detect them. There may be high pollution levels in one area, and a few meters away, very low concentrations. Thus, the results depend to a large extent on where sensors are installed: if they are not installed in the right locations, their results will not be representative. Non-optical sensors can be highly sensitive to weather conditions, such as wind speed, temperature and humidity. Surveillance cameras do not have these drawbacks, however they depend on daylight making them useless at night. One way to use cameras at night is by using infrared sensors. For example, optical gas imaging cameras use different infrared spectral ranges to visualize and detect different types of gas emission, such as methane or ethylene (Naranjo et al., 2010).

The use of surveillance cameras for vision-based detection is an appropriate and low-cost approach when compared to other sensors, especially because the emissions to be detected are in the visible spectrum. For this reason, this work studies the possibility of using video surveillance cameras and adapting the existing infrastructure of the industrial plants themselves. One of the disadvantages of the proposed method is that the chemicals found in the emissions cannot be recognized. Thus, it is not possible to analyze the concentration of different chemical compounds, only their location and the area they occupy. This method is able to detect emissions based on examples of previous emissions. If a new type of emission were to occur, it would probably not be detected properly and the network would have to be retrained.

Surveillance cameras are often low resolution, low quality devices (Rofeim, 2019), badly focused on irrelevant locations. They must be evaluated on an individual basis to determine their usefulness for emission detection. Cleaning and maintenance may be required.

Video surveillance cameras typically cover the majority of industrial plant areas, with coverage of almost 100 % in many cases. For this reason, they are much less intrusive than the installation of sensors, especially for areas such as the air space directly above the industrial plant. Although surveillance cameras cannot detect the chemical composition of emissions, they may be able to detect their location and size.

There are deep learning approaches to detect smoke in RGB imagery (Park and Song, 2019), some of which use ultraviolet (Osorio et al., 2017; Wang et al., 2020) or infrared (Wang et al., 2022) imagery. However, to the best of the authors' knowledge, there is no existing research about image segmentation for fugitive emission detection in industrial plants. This may be because companies are reluctant to make data showing levels of contamination public. In addition, private research may not be shared in order to have an advantage over competitors.

Creating a dataset for this purpose is complex as it might interfere with the normal operation of the company, labeling the images is costly and the images may be of a sensitive nature. The only dataset found that could be of use is a dataset from Project RISE (Hsu et al., 2020), which obtains its data from outside an industrial plant rather than using existing infrastructure from the plant. Furthermore, the dataset is not labeled for semantic segmentation.

At present, there is no research in this field and certainly none with state-of-the-art technologies based on fully convolutional neural networks for semantic segmentation. This paper focuses on fugitive emissions, using a solution based on state-of-the-art image segmentation methods. The use of surveillance cameras is proposed

to gather these complex images with low opacity emissions and gaps.

From this proposal several questions arise:

1. The use of pixel-wise segmentation to obtain a mask detailing the location of the emission;
2. The distinction of fugitive emission from other gases such as water vapor and/or clouds despite not being able to obtain their composition;
3. The evaluation of the difference between a binary classification and a multi-class classification and its effects on the fugitive emission class;
4. The execution of trained models on a dataset with realistic emission/non-emission proportions in order to test their effectiveness if put into production;
5. The feasibility of transferring already trained models with images from an industrial plant for application in images from other industrial plants;
6. The study of the minimum number of images necessary for the training of a model.
7. The transfer of models from other industrial plants to reduce the number of images needed to train.
8. The use of detections as an alarm to warn of emissions.

These questions are addressed in Section 3. To answer these questions, three different datasets have been developed, each focusing on a single industrial plant. Each dataset is composed of 1000 images with emission and up to 11,500 images without emission. The non-emission images are simpler to obtain for binary classification because they do not require the creation of a ground truth mask.

As a predictor tool, the semantic segmentation network known as DeepLabV3 + (Chen et al., 2018) is used. This network, developed by Google for Tensorflow, is the state of the art in this field of research, but has never been used for fugitive emissions localization. It obtains excellent results in other areas such as autonomous driving (Sgibnev et al., 2020), land cover classification (Pedrayes et al., 2021), or brain tumor detection (Choudhury et al., 2018) among other fields for pixel-wise segmentation. To obtain the best possible results with this network, a thorough hyper-parametric tuning process is necessary. In this paper, the most important aspects to fine-tune the network for these specific datasets are detailed.

## 2. Methods and materials

### 2.1. Datasets

This paper shows images from three different industrial plants (Plant1, Plant2 and Plant3). Each industrial plant provides images from a single camera. These cameras were already existing surveillance cameras to monitor buildings with risk of fugitive emission. However, the monitoring process was manual. In the figures of this paper, the regions corresponding to the buildings are colored with the red corresponding to the building class. For the images of industrial plants 1 and 2, a crop is applied to avoid showing the silhouette of the building. This is to protect the anonymity of the company that provides this information. However, all experimentation has been done with the original images.

A dataset for each industrial plant is generated. Training and test sets are divided in 75 % and 25 % respectively. All the images are taken at random times on four different days, excluding night-time since the cameras do not have enough visibility. The minimum time separation between two images is 5 s. The cameras have a 4 K/8MP sensor, a framerate of 30fps and a horizontal angle of 95–10 degrees. Quality is degraded because zoom levels are adjusted to frame the







images to the corresponding buildings. The lens of video surveillance cameras are designed to have a larger field of view in order to cover as much ground as possible. The area of interest is only a specific area of the image itself. In addition, they are low cost and low maintenance cameras because they are located in inaccessible places, so it is common that there is blurring due to dirt on the lenses. All the images consist of the red, green, and blue (RGB) bands and have a resolution of  $2048 \times 1536$  pixels or  $1024 \times 768$  pixels. To alleviate the amount of memory needed in VRAM and to keep all image sizes constant, all images are scaled to  $512 \times 384$  pixels using the bicubic interpolation.

A real use case has a proportion of 1:34 emission to non-emission images. During a normal day on average, only one image with fugitive emissions was captured for every 34 images with no emission. To test how models trained with different proportions performed in a real environment, a test set of 250 fugitive emission images, and 8500 non-fugitive emission images was created and evaluated. This real test is generated for Plant1 and Plant2.

To study the importance of the proportion used to train a model that will normally be used under other proportions (realistic proportions), multiple variations are designed for each dataset. Each variation uses a different number of non-emission images. In Table 1, the different proportions for the datasets for Plants 1 and 2 along with the number of images that correspond to emissions and non-emissions for training and test sets are shown. The fugitive emission images are maintained throughout all the variations to improve comparability. The dataset from Plant 3 has only the proportion 2:1 since it does not have enough images.

Plant1 has the following classes: building, cloud, sky, fire, water vapor chimneys, and fugitive emission. Plant2 has: building, cloud, sky, water vapor chimneys, and fugitive emission. Plant3 has: building, cloud, sky, and fugitive emission. The labeling was carried out by experts using software tools and then reviewed by the operators of the industrial plants.

Classes in the ground truth masks have the following colors associated: .

-  Building
-  Water vapor chimney
-  Cloud
-  Fire
-  Fugitive emission
-  Sky

Fugitive emissions are the target class for this evaluation study. This is the only class of interest, therefore, a comparison between multi-class classification and binary classification and its effects on the target class can be studied.

## 2.2. Network architecture

DeepLabv3+ (Chen et al., 2018) is the most recent version of DeepLab (Chen et al., 2014; Chen et al., 2017a; Chen et al., 2017b), a convolutional neural network architecture for semantic segmentation developed by Google. The architecture of this network is based on an encoder-decoder structure with an Atrous Spatial Pyramid Pooling (ASPP) module in the encoder part. This evaluation study

**Table 1**  
Proportions for the different datasets. (emission:non-emission).

Proportion	Train	Test
2:1	750:375	250:125
1:1	750:750	250:250
1:2	750:1500	250:500
1:4	750:3000	250:1000

uses the official implementation from Google's Github, which can be accessed with the following link: <https://github.com/tensorflow/models/tree/master/research/deeplab>. This implementation is based on TensorFlow. A diagram of the DeepLabV3+ architecture can be seen in Fig. 1.

## 2.3. Metrics

One of the most common metrics to determine the quality of a prediction in semantic segmentation is the  $F_1$ -Score. As seen in Eq. (1), it is calculated as a combination of both Precision and Recall and is equivalent to the Dice Coefficient with two classes.

$$F_1 = \frac{2 \times P \times R}{P + R} \quad (1)$$

Precision is calculated using Eq. (2), as the correctly classified pixels from the total predicted pixels. Recall is calculated using Eq. (3), as the pixels classified correctly from the pixels that correspond to that particular class in ground truth. If both metrics are very different (unbalanced) it usually means that the predictions will tend to over classify pixels from that particular class (low Precision and high Recall), or that it will only predict pixels that are too obvious (high Precision and low Recall).

$$P = \frac{TP}{TP + FP} \quad (2)$$

$$R = \frac{TP}{TP + FN} \quad (3)$$

Intersection-Over-Union or IoU is equivalent to the Jaccard Index and is used to measure the area of similarity of a prediction to its ground truth. This metric is common for measuring the quality of prediction in the context of image segmentation. It is calculated using Eq. (4), as the ratio between the true positives and the sum of all the pixels that are not true negatives.

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{TP}{TP + FN + FP} \quad (4)$$

## 2.4. Training

To get the best possible results, hyper-parameters must be tuned accordingly for the dataset used. Optimal hyper-parameters were studied for the three datasets separately. As they are similar, the optimal hyper-parameters coincide.

The following hyper-parameters were tuned to improve predictions: different input sizes ( $2048 \times 1536$ ,  $1024 \times 768$ ,  $512 \times 384$ , and  $256 \times 192$ ) and their effect on batch size; number of classes using multi-class and binary predictions; class weighting comparing the Median Frequency Weighting (MFW) (Eigen and Fergus, 2015) method against custom weights; learning rate and epochs; output strides of 8, 16, and 32; different backbone networks (Resnet50, Xception45, Xception65, Xception71, MobileNetV2, MobileNetV3Small, MobileNetV3Large); L2 regularization; and solver algorithms such as Adam or Stochastic Gradient Descent with Momentum (SGDM).

Data augmentation was used to obtain more training data in order to reduce overfitting, and to improve predictions. The augmentation process consists of zooms of the images with varying zoom values ranging from 0.5 to 2.0 at intervals of 0.25. The dataset was shuffled for epoch to prevent overfitting.

As testing all possible combinations of hyper-parameters with a single computer would be too time consuming, each hyper-parameter was adjusted individually. Batch size was re-evaluated for every change that involves VRAM usage. Architecture changes, such



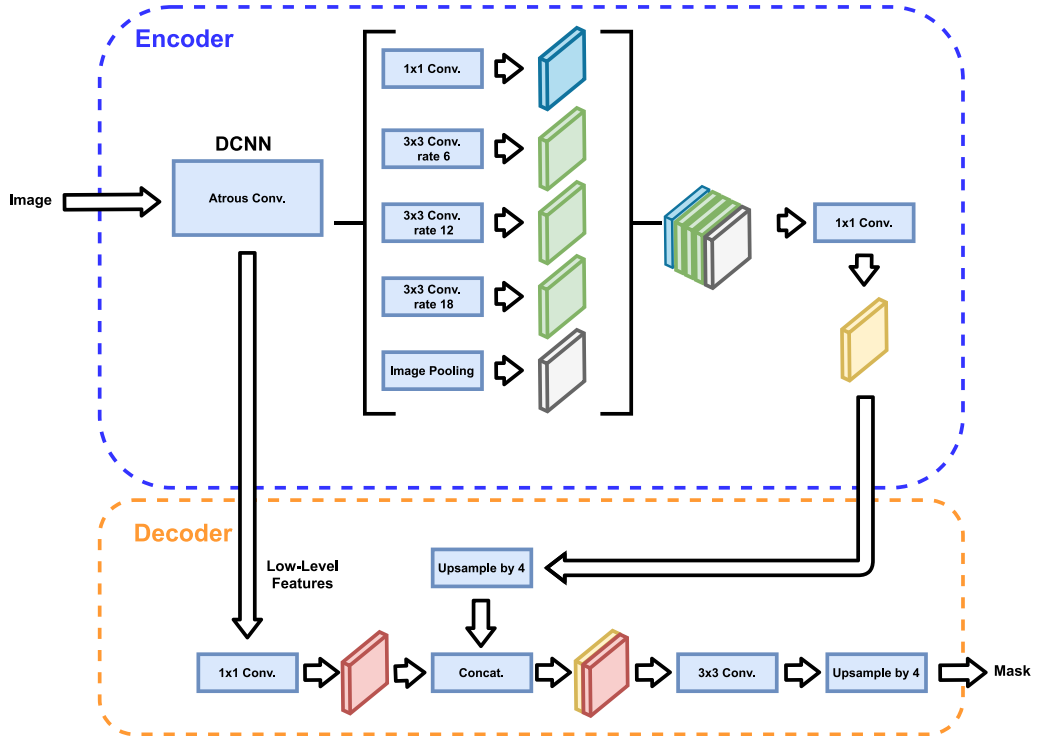


Fig. 1. DeepLabV3+ architecture.

as the backbone network, can reduce VRAM usage, allowing for larger batch sizes.

The best configuration found for binary classification for all three datasets can be seen in Table 2. As the three datasets are similar, the best configurations are the same, except for the value of the custom class weights, which may vary slightly. For multi-class classifications, hyperparameters are the same except for class number, and class weighting. In the multi-class classification no custom class weighting was performed, only MFW weighting was used. This is due to the added difficulty of the compound effect of the different classes affecting each other. Obtaining a configuration of custom weighting values would be too time consuming.

Models were trained using an NVIDIA RTX 2080 Ti GPU with 11 GB of VRAM. Training the DeepLabV3+ network from scratch can take weeks, mainly due to the backbone network. For this reason, pretrained models with the ImageNet dataset (Deng et al., 2009) are used for each of the backbone networks. From these pretrained models only the backbone network part of the architecture is loaded. The pretrained models are provided by Google in its original GitHub for DeepLab.

### 3. Results and discussions

#### 3.1. Multi-class experimentation

First, to determine if an image segmentation of fugitive emissions in industrial plants can be obtained from surveillance camera images, multi-class experiments were carried out. This section

shows the results for experiments using DeepLabV3+ for multi-class segmentation. These experiments use the MFW method for class weighting balancing.

Metrics from Table 3 show high F<sub>1</sub>-Score values, surpassing the 80% barrier in every class for all three datasets proving that a functional prediction mask can be obtained. This solves question (1) from the "Introduction" in Section 1. Fugitive emission is one of the classes with the lowest F<sub>1</sub>-Scores and its metrics are usually slightly biased towards Recall over Precision. These metrics are to be expected since the fugitive emission class is much harder to see and has greater variability in both area and color than other classes. This makes both the network and the ground truth prone to error. In the same way, since it is hard to distinguish between them, classes Cloud and Sky might increase their metrics by being merged into the same class.

From these results it can be established that it is possible to differentiate clouds, water vapor chimneys, and fugitive emissions, solving question (2).

Visualization for the experiments of Table 3 can be seen for Plant1 in Fig. 2, for Plant2 in Fig. 3, and Plant3 in Fig. 4. The first column of the three figures shows the input image; the second column, the original multi-class ground truth mask; the third column, the multi-class classification prediction from the model; the fourth column, a superposition of the ground truth mask with the input image; and the fifth column, the superposition of the predicted multi-class mask from the model over the input image.

Results for Plant1 (Fig. 2) are very good visually. Most predictions are indistinguishable from the ground truth, except the fourth image

**Table 2**  
Training parameters for DeepLabV3+.

Training parameters	
Input size	512 × 384 × 3
Classes	2
Backbone	Xception65
Output stride	16
Padding	Yes
Solver	Adam
Epochs	80
Batch size	6
Learning rate	0.00005
Class weighting	Non-Target: ~ 0.40 - Target: ~ 0.60
Gradient clipping	No
L2 regularization	0.0004
Data augmentation	Scale 0.5–2.0 with 0.25 steps
Shuffle	Yes

**Table 3**  
Metrics for the multi-class classification experiments.

Dataset	Class	Precision	Recall	IoU	F <sub>1</sub>
Plant1	Building	0.995	0.997	0.992	0.996
	Vapor	0.915	0.900	0.831	0.907
	Clouds	0.955	0.834	0.802	0.890
	Fire	0.794	0.837	0.688	0.815
	<b>Emission</b>	0.836	0.832	0.715	0.834
Plant2	Sky	0.926	0.951	0.884	0.938
	Building	0.997	0.988	0.985	0.992
	Vapor	0.744	0.934	0.707	0.847
	Clouds	0.800	0.931	0.755	0.861
	<b>Emission</b>	0.767	0.895	0.704	0.826
Plant3	Sky	0.984	0.935	0.921	0.959
	Building	0.992	0.994	0.986	0.993
	Clouds	0.779	0.710	0.591	0.743
	<b>Emission</b>	0.881	0.939	0.833	0.909
	Sky	0.954	0.936	0.895	0.945

which detects fire even when the ground truth shows none. In this case, it appears that the ground truth may be labeled incorrectly, as a small transparent fire can be seen in the original input image. This means that the model, for this particular case, is outperforming the ground truth. Obviously, this is impossible to detect just by looking at the metrics, but it does mean that the model may be performing better than the numbers show. Results from Plant2 (Fig. 3) are also visually excellent. Most predictions are indistinguishable from the ground truth, except the second image, which detects fugitive emissions when the ground truth does not. Finally, results from Plant3 (Fig. 4) are also very good visually. However, the second image shows that the emission is not fully detected.

### 3.2. Binary experimentation

Although multi-class segmentation can successfully localize fugitive emissions in industrial plants, in order to reduce the complexity of the ground truth masks, binary segmentation is evaluated and compared with multi-class segmentation. Binary segmentation has great advantages over multi-class segmentation. It is much cheaper and easier to build ground truth masks with a single class rather than multiple classes that require several different regions per image. In addition, the process required to adjust the network is much simpler, since parameters such as class weighting are easier to set. This is because there are only two classes, so modifying one only affects the other rather than affecting multiple classes at the same

time. This allows for an in-depth study of class weighting for class balancing. Table 4 only show results for the emission class. The “non-emission” class is not detailed since it is not needed.

Results from the MFW class weighting experiments for binary segmentation shown in Table 4 are comparable to those of the multi-class experiment from Table 3. These experiments are more biased towards Recall, obtaining very high Recall but lower Precision. Even though there is a disparity between Recall and Precision, F<sub>1</sub>-Score values are not far from those in the multi-class experiments (see Table 3). Using binary classification instead of multi-class makes the imbalance between Recall and Precision bigger, lowering the quality of the segmentations. Plant3 is the exception because multi-class and binary results are practically the same. This indicates that these differences are highly dependant on the dataset used and its complexity, given that Plant3 has no elements such as water vapor chimneys or fire. In answer to question (3) from the “Introduction” in Section 1, merging all non-target classes into a single class causes that class to be more unbalanced than the target class than when they are separated. In other words, the difference in the number of pixels between the two classes increases, causing Recall to remain high, but Precision to decrease.

### 3.3. Class weighting experimentation

The results of the binary segmentation experiments show that their metrics are biased towards Recall over Precision. This clearly limits the quality of the prediction: Recall and Precision must be balanced. Class weighting has a great impact on the training process of the model. Standard methods for class weighting, such as Inverse Frequency Weighting (IFW) (Cui et al., 2019) and Median Frequency Weighting (MFW) are often used. However, as there is a problem balancing Recall and Precision, custom weights are studied in order to reach a balance. Plant3 is not included in this study as its Precision and Recall are already acceptably balanced. Binary segmentation benefits from the fact that only the best weights for two classes need be found. This greatly simplifies the study of how class weighting affects the results obtained.

In Table 5 multiple values for class weighting are tested to study the behavior of the network under different weightings. To make the class weighting experimentation easier, all values are normalized between 0 and 1. In this way, if a value of 0.60 is used for the target class, it is assumed that the remaining 0.40 is used for the non-target class.

Metrics from Table 5 show that custom weights for binary classification can balance Recall and Precision and improve overall accuracy, increasing its F<sub>1</sub>-Score. Values from the custom weights for binary classification achieve even better results than those of multi-class classification with MFW. Balancing both Recall and Precision has a great impact on the quality of the model. After balancing the Recall and Precision correctly, there is no benefit to adding classes other than the target classes to segment. In this case, only the fugitive emission class is of value. For this reason, and because of the advantages mentioned above, the rest of the study will be focused on binary segmentation for the fugitive emission class.

Visualization for the experiments with the highest F<sub>1</sub>-Scores in Table 5 can be seen for Plant1 in Fig. 5, for Plant2 in Fig. 6, and Plant3 (from Table 4) in Fig. 7. The three figures show: in the first column, the input image; in the second column, the original multi-class ground truth mask; in the third column, the binary classification prediction from the model; in the fourth column, a superposition of the ground truth mask with the input image; and in the fifth column, the superposition of the predicted binary mask from the model over the input image.

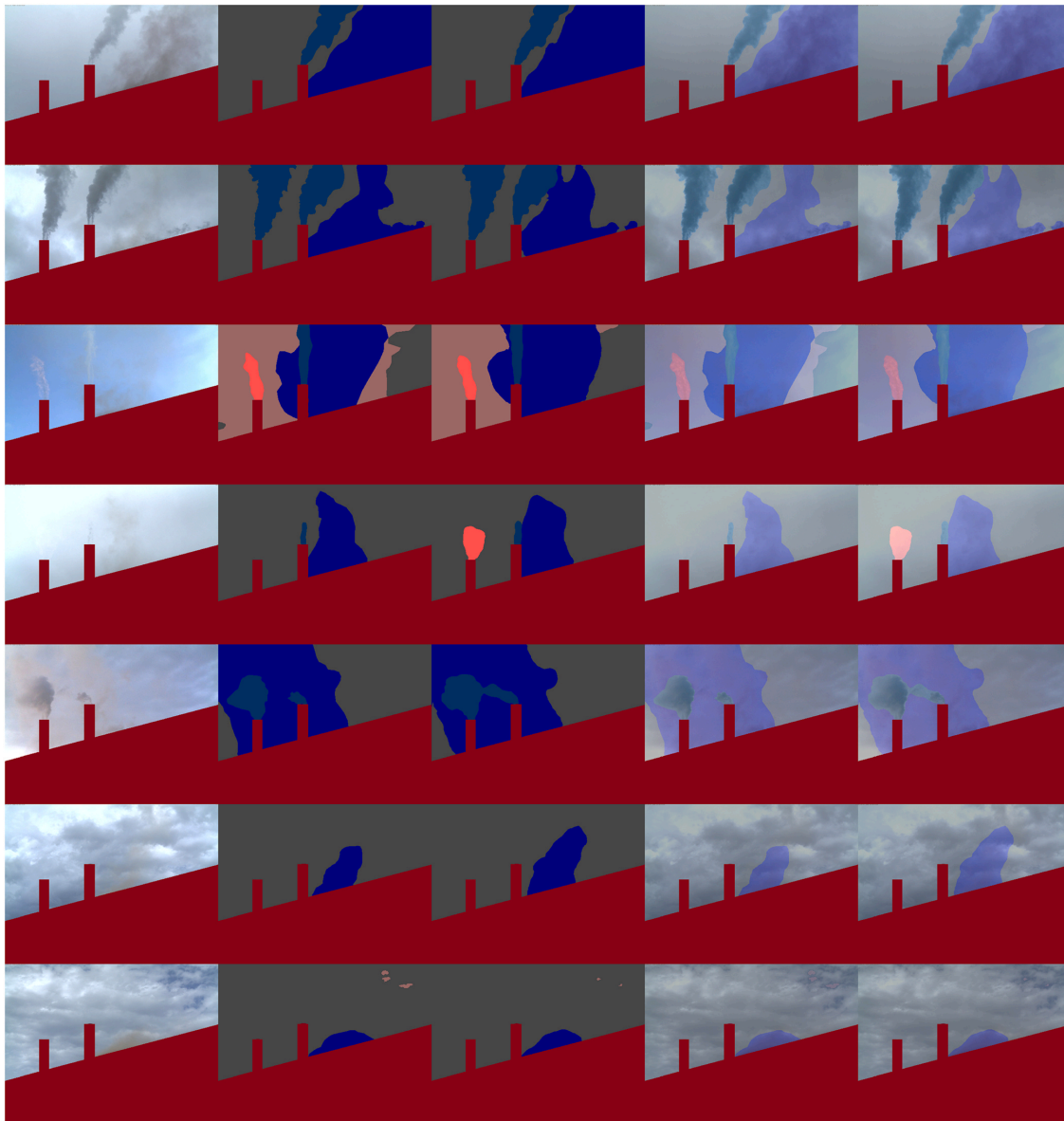


Fig. 2. Plant1 multi-class classification test visualization.

The results for Plant 1, 2 and 3 are visually very good, as can be seen in (Fig. 5, 6, and 7). In Plant1 most predictions are indistinguishable from the ground truth, except the second image, which detects fewer fugitive emissions than the ground truth. In Plant2 most predictions are also indistinguishable from the ground truth, except the second image, which detects fugitive emissions when the ground truth does not. Finally, Plant3 has nearly perfect visual results.

### 3.4. Training with different class proportions and a real test

Binary segmentation with custom class weighting has robust results. To further validate these results, a larger test set is evaluated. This test set aims to replicate the proportions in which images exhibit fugitive emissions over a full day. In this case, for every 34 images without emission, one image with emission is produced. For the sake of simplicity, this is referred to as a 1:34 proportion.

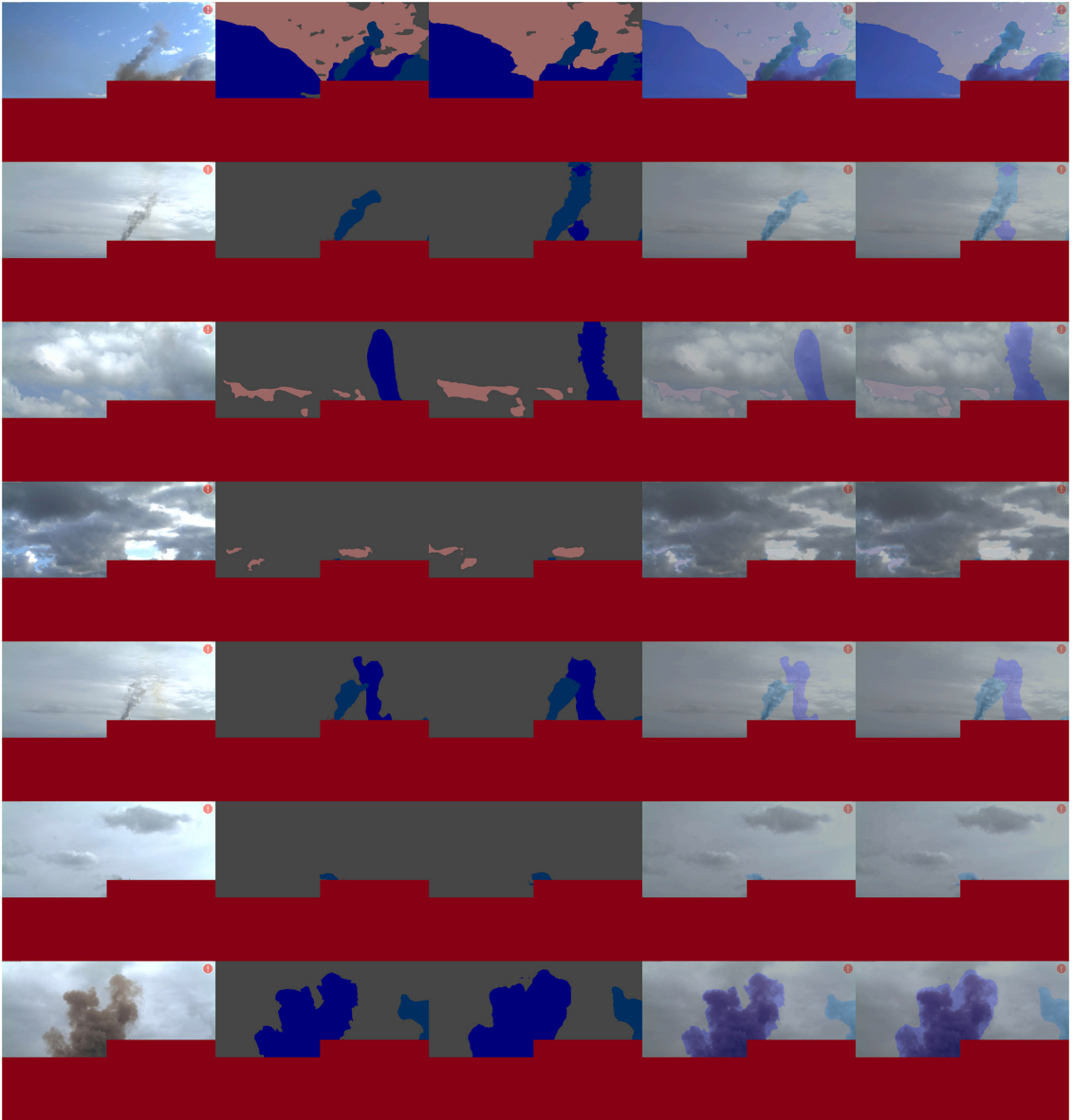


Fig. 3. Plant2 multi-class classification test visualization.

In the case of Plant3 there are not enough images to create this kind of test set. For Plant1 and Plant2, 250 images with emission and 8500 non-emission images are used.

When a network trained with a set of 2:1 proportion is tested with the real proportion of 1:34, Precision is greatly reduced due to new False Positives (see Table 6). However, the Recall remains the same since no new images with fugitive emission were added to the

test. This effect can be seen in the first row for Plant1 and the fifth row for Plant2 in Table 6. In order to solve this problem, experiments with varying class proportions were carried out. The objective was to determine if a change in proportions during training affects the Precision of the predictions when adjusting the class weighting of the classes to balance Recall and Precision. Table 6 shows the best custom class weighting experiment for each proportion.

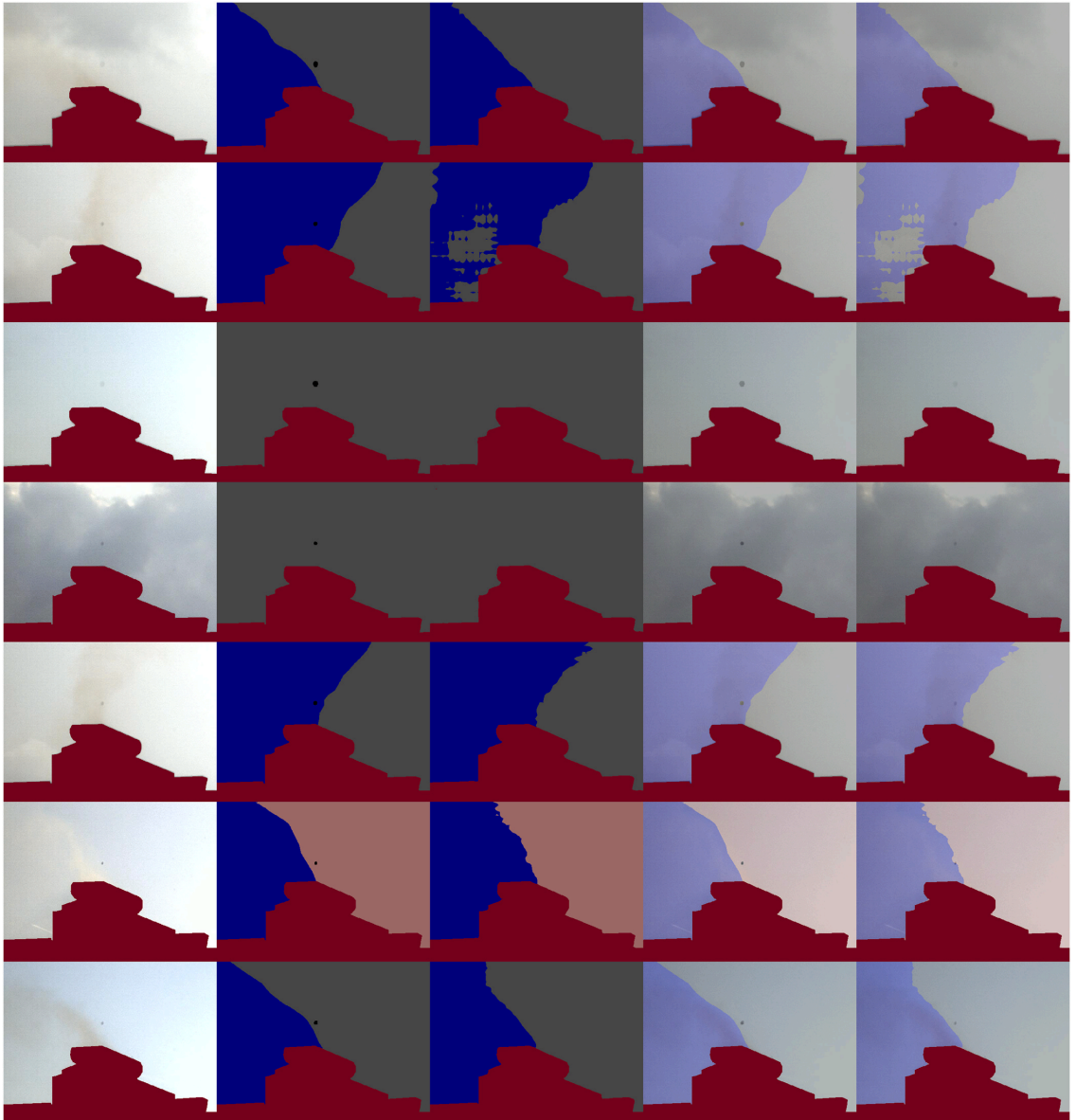


Fig. 4. Plant3 multi-class classification test visualization.

Table 4  
Metrics for the binary classification experiments.

Dataset	Precision	Recall	IoU	F <sub>1</sub>
Plant1	0.634	0.958	0.617	0.763
Plant2	0.648	0.965	0.633	0.775
Plant3	0.885	0.938	0.836	0.911

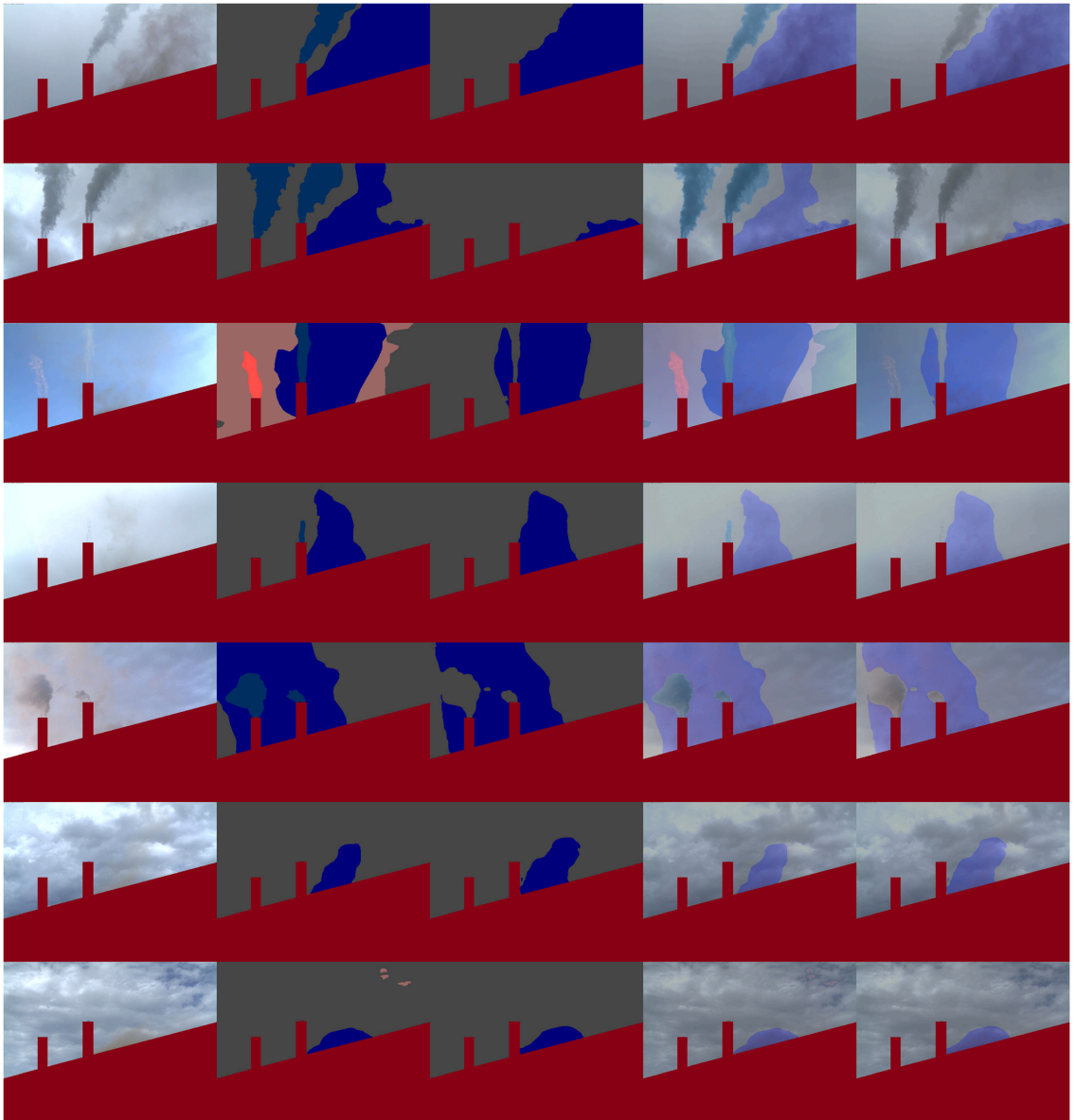
In Table 6 it is observed that a proportion of one emission image per two non-emission images (1:2) is the optimal proportion in both datasets.

A higher non-emission proportion gives the model far more stability and fewer false positives. This discovery is vital since the standard training tests do obtain excellent results. To answer question (4), if no real test were done, the erroneous assumption that the

**Table 5**  
Metrics for the binary classification with custom class weighting experiments.

Dataset	Weight	Precision	Recall	IoU	F <sub>1</sub>
Plant1	(MFW) 0.88	0.634	0.958	0.617	0.763
Plant1	0.70	0.765	0.884	0.695	0.820
Plant1	0.60	0.826	0.841	0.715	<b>0.833</b>
Plant2	(MFW) 0.923	0.648	0.965	0.633	0.775
Plant2	0.80	0.779	0.891	0.711	0.831
Plant2	0.70	0.842	0.842	0.727	<b>0.842</b>
Plant2	0.60	0.866	0.779	0.695	0.820

models could generalize as well as the 2:1 test in a real scenario could be made. Training using a realistic proportion of 1:34 is impractical, as the time required to set up and train such a model would be too time consuming; the number of images required is beyond the scope of this research. For this same reason, no experiments are performed with proportions higher than 1:4, even though such experiments would be interesting to confirm this statement or to find an upper limit.



**Fig. 5.** Plant1 binary classification test visualization.

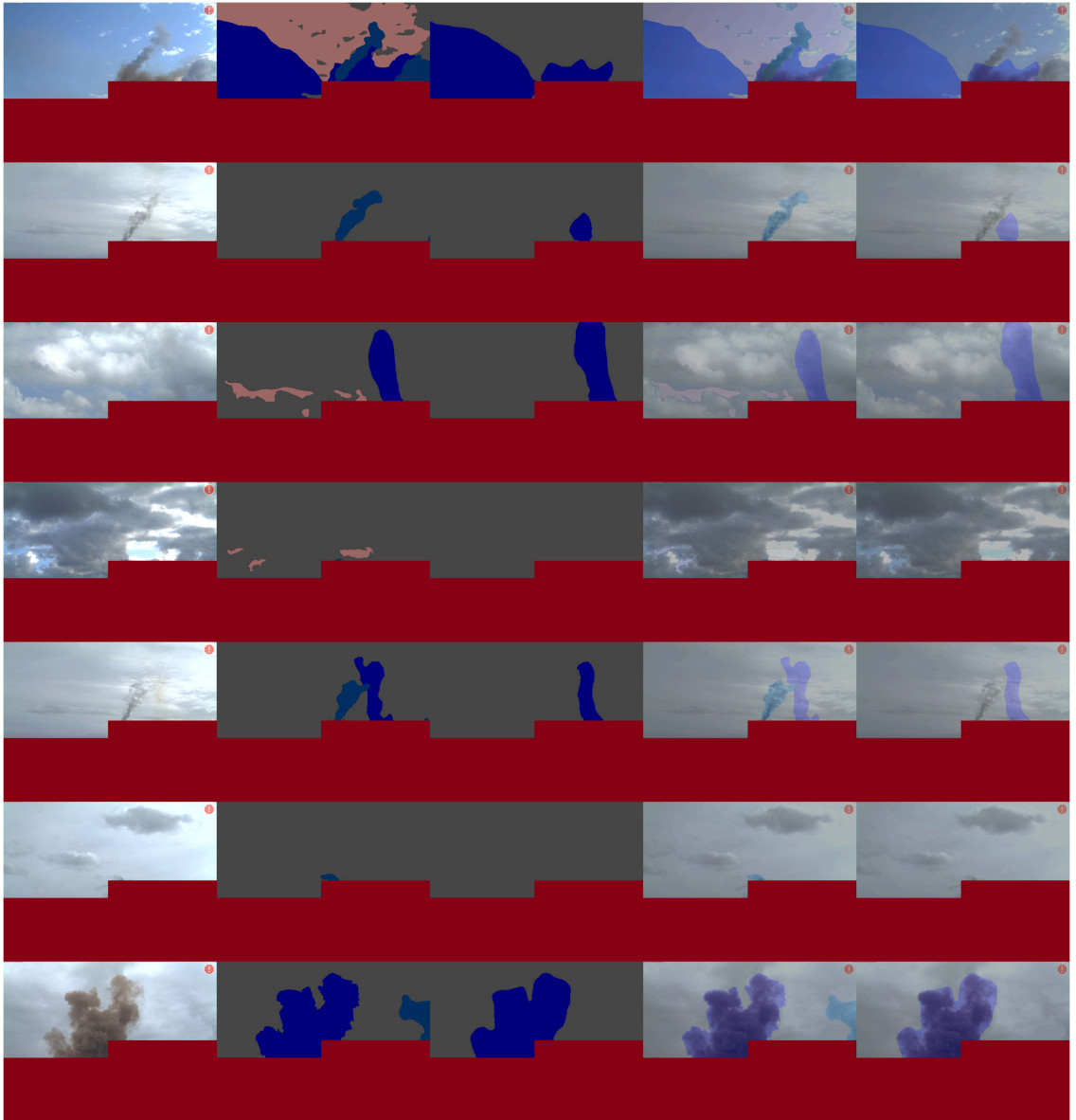


Fig. 6. Plant2 binary classification test visualization.

### 3.5. Generalization between datasets

Since a large amount of parameter configuration is required, it is reasonable to try to take advantage of models trained in one industrial plant to be used in another. Experiments using a model trained with images from one industrial plant and testing with images from another are carried out and their results are presented in Table 7.

Metrics from Table 7 show that testing with different datasets does not produce good quality predictions. To answer question (5), models are not directly transferable between datasets. Clearly, models cannot be used indistinctly in different industrial plants.

### 3.6. Reduced training

As it is not possible to reuse a model already trained in another industrial plant, it is interesting to study the minimum number of images needed to train a model and obtain metrics similar to using a large dataset. This could greatly reduce time and costs. Experiments with fewer images for Plant2 and Plant3 are evaluated in Table 8. For these experiments, a proportion of 2:1 is maintained to facilitate comparisons. Testing sets are not modified to make results directly comparable.

To answer question (6), the reduced training shows that using only 100 images is enough to obtain acceptable results, as shown in

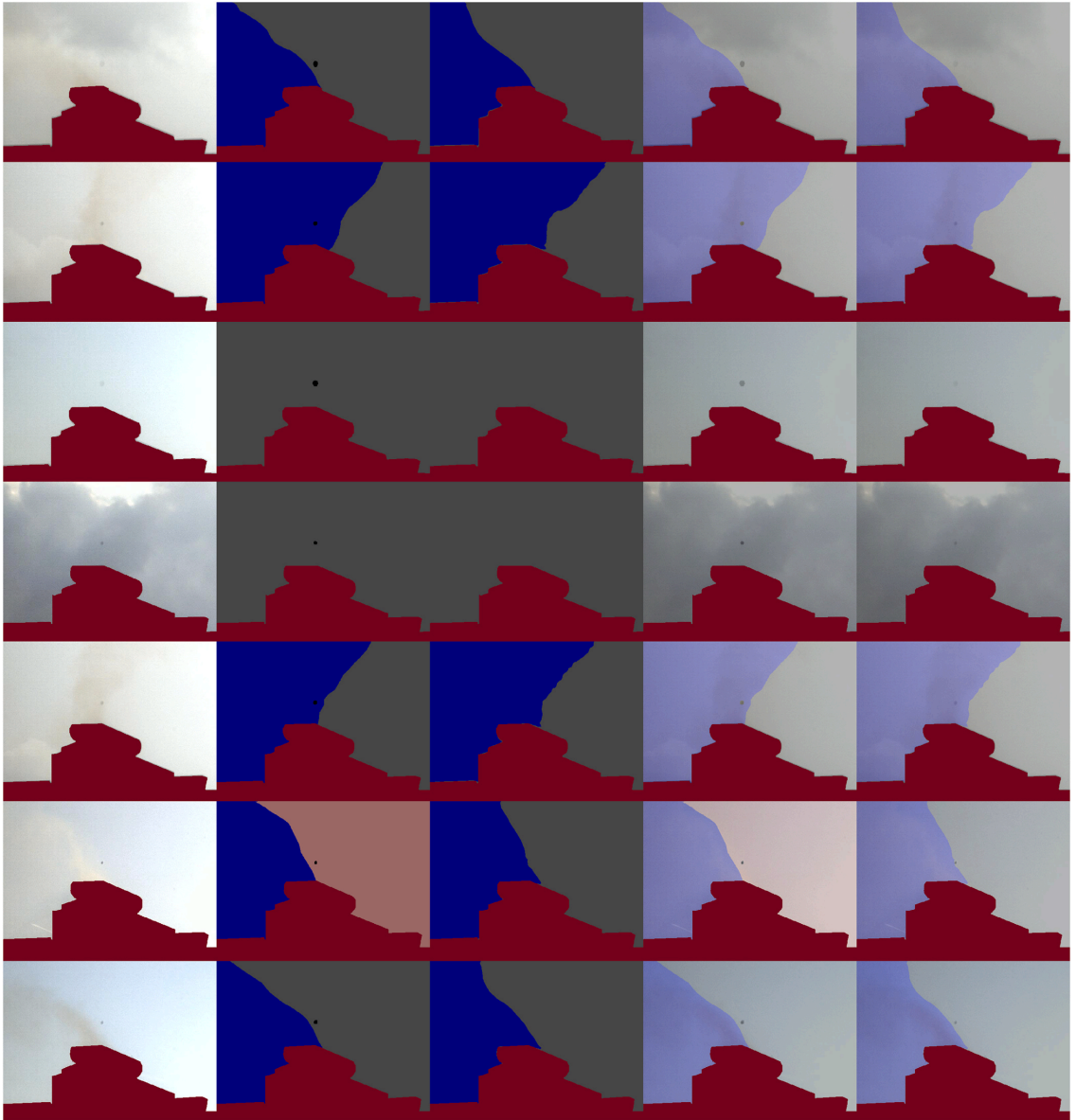


Fig. 7. Plant3 binary classification test visualization.

**Table 6**  
Metrics for the proportion experiments.

Parameters			Tests with train proportions				Tests with real proportions (1:34)			
Dataset	Train prop.	Weight	Precision	Recall	IoU	F <sub>1</sub>	Precision	Recall	IoU	F <sub>1</sub>
Plant1	2:1	0.60	0.826	0.841	0.715	0.833	0.585	0.841	0.527	0.690
Plant1	1:1	0.51	0.831	0.787	0.678	0.808	0.678	0.787	0.573	0.728
Plant1	1:2	0.60	0.833	0.815	0.700	0.824	0.819	0.815	0.691	0.817
Plant1	1:4	0.50	0.783	0.850	0.688	0.815	0.540	0.850	0.493	0.660
Plant2	2:1	0.70	0.842	0.842	0.727	0.842	0.365	0.842	0.342	0.509
Plant2	1:1	0.55	0.852	0.826	0.723	0.839	0.786	0.826	0.675	0.806
Plant2	1:2	0.635	0.849	0.816	0.713	0.832	0.835	0.816	0.702	0.825
Plant2	1:4	0.57	0.843	0.823	0.714	0.833	0.815	0.823	0.694	0.819



**Table 7**  
Metrics for the dataset cross testing experiments.

Dataset	Precision	Recall	IoU	F <sub>1</sub>
Plant1 → Plant2	0.728	0.380	0.333	0.499
Plant1 → Plant3	0.446	0.129	0.111	0.200

**Table 8**  
Metrics for reduced training set.

Dataset	Images	Precision	Recall	IoU	F <sub>1</sub>
Plant2	100	0.668	0.834	0.590	0.742
Plant2	250	0.697	0.837	0.614	0.761
Plant3	100	0.847	0.895	0.770	0.870
Plant3	250	0.862	0.890	0.779	0.876

**Table 9**  
Metrics for transfer learning experiments.

Dataset	Images	Precision	Recall	IoU	F <sub>1</sub>
Plant1 → Plant2	10	0.698	0.620	0.489	0.657
Plant1 → Plant2	50	0.777	0.683	0.571	0.727
Plant1 → Plant2	100	0.788	0.757	0.629	0.772
Plant1 → Plant2	250	0.729	0.815	0.625	0.770
Plant1 → Plant3	10	0.801	0.801	0.668	0.801
Plant1 → Plant3	50	0.808	0.927	0.760	0.863
Plant1 → Plant3	100	0.837	0.871	0.745	0.854
Plant1 → Plant3	250	0.841	0.916	0.781	0.877

**Table 8.** These metrics are almost 5 % lower than the experiments from **Table 4**, which use 1125 images. However, a 5 % reduction using a dataset ten times smaller may be acceptable when time and cost restraints exist.

**3.7. Transfer learning**

Since at least 100 images are needed to achieve robust training, it is interesting to study whether this number can be further reduced by reusing a model already trained in another industrial plant. Even if it cannot be used directly, perhaps another model can serve as a checkpoint to reduce training time on other datasets. In this section, experiments are carried out to observe if a model trained for one industrial plant can be used as a base for training with a new dataset for another industrial plant (i.e., transfer learning). This would mean training with fewer images. In both cases, “Transfer learning” and “Reduced training” are trained with the exact same images to allow for comparisons.

Transfer learning results from **Table 9** show an improvement of 3% in the F<sub>1</sub>-Score for Plant2 when compared to the reduced training of 100 images from **Table 8**. However, in the case of Plant3 this does not improve its metrics, and even reduces its F<sub>1</sub>-Score by 2 % in the 100-image experiment. This shows that the models might be too dependant on the dataset used and cannot be generalized for the application of emission segmentation. Since it is already possible to train with only 100 images, transfer learning may not be necessary. Thus, in answer to question (7), transfer learning cannot be used to reduce the number of images needed.

**3.8. Emission detection**

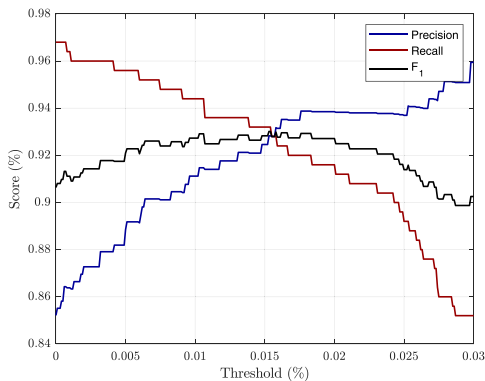
Once a robust model for fugitive emission segmentation has been obtained, it seems reasonable to use these segmentations to detect fugitive emissions. To raise an alarm when a fugitive emission is detected it is necessary to apply semantic segmentation to each frame of the video feed. This alarm is used to indicate whether or not there are fugitive emissions in the image (i.e., it is a binary alarm). In this way, the predictions are used for image classification. The realistic test set is used to study its behavior.

In order to avoid raising the alarm for false positives consisting of extremely small regions, an area threshold for the segmented areas is needed. This threshold is measured as the minimum percentage of pixels with emission out of the total pixels of the image. In this section, the optimal area thresholds are calculated using the models from the realistic test from Plant1 and Plant2.

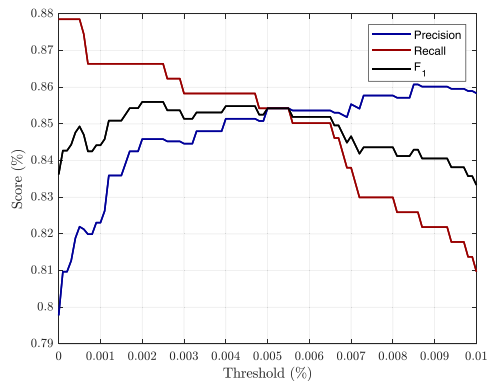
For each threshold evaluated, using steps of 0.01 %, the images of the dataset that will be considered as true emissions are determined. Then the predicted images with and without emission are analyzed as a function of the threshold used, obtaining the metrics of Recall, Precision and F<sub>1</sub>-Score. These metrics are represented in **Figs. 8(a)** and **8(b)**. It should be noted that although the metrics used are the same, in this case they are calculated per image and not per pixel.

Plant1 has an optimal threshold of 1.56 % of the area of the total image (**Fig. 8(a)**). Plant2 has an optimal threshold of 0.5 % of the area of the total image (**Fig. 8(b)**).

Metrics from **Table 10** show that over 85 % and almost 93 % F<sub>1</sub>-Score values are achieved for Plant1 and Plant2 respectively. These experiments demonstrate that this method can be successfully used for image classification, answering question (8). In addition, the detected segmentation regions can be used to determine different levels of severity based on area or shape. For example, a fugitive emission severity level from 1 to 10 could be declared using 10% area



(a) Plant1



(b) Plant2

**Fig. 8.** Threshold study.

**Table 10**  
Metrics for the emission detection alarm.

Dataset	Threshold	Precision	Recall	F <sub>1</sub>
Plant1	0.0050	0.854	0.854	0.854
Plant2	0.0156	0.928	0.928	0.928

increments of the emissions. In other words, a severity level of 3 could be 30 % of the sky area occupied by emissions.

#### 4. Conclusion

The method proposed in this paper to segment and detect fugitive emissions in images taken from surveillance cameras in industrial plants, reaches results above 80 % F<sub>1</sub>-Score. Clouds, water vapor chimneys, and fugitive emissions can be distinguished with no problem. These results are achieved despite the low resolution and quality of the surveillance cameras used to acquire the images, some of which include images with dirt on the lenses. This means that the location and detection of fugitive emissions using surveillance cameras is possible. Overcoming other traditional methods such as sensors that measure volume concentrations or chemical compounds in the air, as these sensors require to be placed in areas where a large flow of emissions is expected, i.e. planned emissions such as emissions produced by stacks. This is a great advantage since a much larger area can be covered than with such sensors. Another advantage over non-optical sensors is that the result of the detection is a new 2D image with classification at pixel level. This type of image is very easy to interpret and verify visually.

Surveillance cameras are a type of optical sensor. These cameras only detect the visible spectrum. If another optical sensor were used that could detect a wider spectral range, results would most likely improve. However, this would also increase the cost dramatically. One of the advantages of this method is that surveillance cameras are low cost sensors, and they usually already exist in the industrial plant.

It is observed that there is no need to add more classes to the training process, since there is no significant improvement over using only the target classes, as long as the Precision and Recall are balanced. This can save labeling time and is far more cost efficient when creating new datasets.

This is relevant since a model trained for a particular camera of a particular industrial plant does not generalize well enough to be used for another industrial plant with the same type of fugitive emissions. The differences between datasets are far too great for the model to produce good results. Furthermore, transfer learning does not provide any improvement over training from a pre-trained model from Imagenet. Transfer learning between different industrial plants is highly dependant on the similarity of the datasets and cannot be used in a generalized manner. However, transfer learning is not needed since with only 100 training images results are close to a full training with a dataset over ten times larger.

It is common to use metrics obtained from a test with fewer images than the training set. It is usually a test that does not follow the class proportions of a real use case, as it commonly focuses on training the target classes. This paper confirms that, whenever possible, a test with as many images as possible, using a realistic proportion, should be used to ensure that the results of a model are fully validated.

Different class proportions during training have a significant effect in real tests even when class weighting is used. Regardless of the class weighting strategy used, the class proportion is of much greater importance. A proportion that is closer to the real scenario works better than training with all the classes in the same proportion. However, custom class weighting is still essential to balance Recall and Precision even if more realistic image proportions are

used. Results from the best model, in a real test, obtain over 80 % F<sub>1</sub>-Score, with balanced Precision and Recall.

Finally, when applying an area threshold to the predictions and used as an image classification approach to serve as an alarm method for emission detection, the F<sub>1</sub>-Score can be as high as 90 %. This proves that this method can be used to detect fugitive emissions and thus help to control pollution in industrial plants.

Given the great importance of pollution control in industrial plants to preserve the environment, the need to continue this research into new ways of detecting and locating fugitive emissions is essential. It becomes apparent that the approach of using semantic segmentation as a basis for further analysis is appropriate, especially considering the speed at which this field is improving. The main contribution of this paper is the study of different possibilities for detecting and locating fugitive emissions with the new state-of-the-art semantic segmentation technologies and their possible uses for emission alerts.

#### CRedit authorship contribution statement

**Oscar D. Pedrayes:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization **Darío G. Lema:** Investigation, Resources, **Supervision Rubén Usamentiaga:** Investigation, Resources, Data Curation, Writing - Review & Editing, Visualization, Supervision, Project administration, Funding acquisition, Funding acquisition **Daniel F. García:** Investigation, Resources, Supervision, Funding acquisition.

#### Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Acknowledgments

This work has been partially funded by the project RTI2018-094849-B-I00 of the Spanish National Plan for Research, Development and Innovation.

#### References

- Bakker, E., Telling-Diaz, M., 2002. Electrochemical sensors. *Anal. Chem.* 74, 2781–2800.
- Bogue, R., 2015. Detecting gases with light: a review of optical gas sensor technologies. *Sens. Rev.* 35, 133–140.
- Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2017a. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 834–848.
- Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2014. Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv:1412.7062.
- Chen, L.C., Papandreou, G., Schroff, F., Adam, H., 2017b. Rethinking atrous convolution for semantic image segmentation. arXiv:1706.05587.
- Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: *Proceedings of the ECCV*. pp. 833–851.
- Choudhury, A.R., Vanguri, R., Jambawalikar, S.R., Kumar, P., 2018. Segmentation of brain tumors using deeplabv3. In: *Proceedings of the International MICCAI Brainlesion Workshop*, Springer.154–167.
- Condren, M., Dunning, H., 2021. Experts push for stricter air pollution standards in new environment legislation. <http://www.imperial.ac.uk/news/231324/experts-push-stricter-pollution-standards-environment/>.
- Cui, Y., Jia, M., Lin, T.Y., Song, Y., Belongie, S., 2019. Class-balanced loss based on effective number of samples. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 9268–9277.
- Davenport, J., Adlard, E., 1984. Photoionization detectors for gas chromatography. *J. Chromatogr. A* 290, 13–32.
- Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE*. pp. 248–255.

- Eigen, D., Fergus, R., 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2650–2658.
- Freeman, H., Harten, T., Springer, J., Randall, P., Curran, M.A., Stone, K., 1992. Industrial pollution prevention! a critical review. *J. Air Waste Manag. Assoc.* 42, 618–656.
- Hsu, Y.C., Huang, T.H., Hu, T.Y., Dille, P., Prendi, S., Hoffman, R., Tshlars, A., Pachuta, J., Sargent, R., Nourbakhsh, I., 2020. Project rise: recognizing industrial smoke emissions. arXiv:2005.06111.
- Johnson, S.M., 1992. From reaction to proaction: The 1990 pollution prevention act. *Columbia. J. Environ. Law* 17, 153.
- Laconde, T., 2018. Fugitive emissions: a blind spot in the fight against climate change. Fugitives emissions-sector profile. INIS 51.
- Lee, H., 2021. Air quality watchdog oks tighter regulations for la county oil refineries. <https://www.dailybreeze.com/2021/11/05/air-quality-watchdog-oks-tighter-regulations-for-la-county-oil-refineries/>.
- Liu, B.Y., Berglund, R.N., Agarwal, J.K., 1974. Experimental studies of optical particle counters. *Atmos. Environ.* 8 (1967), 717–732.
- Naranjo, E., Baliga, S., Bernascolle, P., 2010. Ir gas imaging in an industrial setting. In: Proceedings of the Thermosense XXXII, SPIE. pp. 160–167.
- Osorio, M., Casaballe, N., Belsterli, G., Barreto, M., Gómez, Á., Ferrari, J.A., Frins, E., 2017. Plume segmentation from uv camera images for so2 emission rate quantification on cloud days. *Remote Sens.* 9, 517.
- Park, J.S., Song, J.K., et al., 2019. Fcn based gas leakage segmentation and improvement using transfer learning. Proceedings of the 2019 IEEE Student Conference on Electric Machines and Systems (SCEMS 2019). IEEE, pp. 1–4.
- Pedrayes, O.D., Lema, D.G., García, D.F., Usamentiaga, R., Alonso, Á., 2021. Evaluation of semantic segmentation methods for land use with spectral imaging using sentinel-2 and pnoa imagery. *Remote Sens.* 13, 2292.
- Rofeim, M., 2019. Why are your security cameras blurry or fuzzy? <https://diysecuritytech.com/why-are-your-security-cameras-blurry-or-fuzzy/>.
- Sgibnev, I., Sorokin, A., Vishnyakov, B., Vizilter, Y., 2020. Deep semantic segmentation for the off-road autonomous driving. The International Archives of Photogrammetry, Remote Sensing and Spatial. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* 43, 617–622.
- Solomon, S., Manning, M., Marquis, M., Qin, D., et al., 2007. Climate Change 2007-The Physical Science Basis: Working group I Contribution to the Fourth Assessment Report of the IPCC, volume 4. Cambridge University Press.
- Wang, J., Tchampi, L.P., Ravikumar, A.P., McGuire, M., Bell, C.S., Zimmerle, D., Savarese, S., Brandt, A.R., 2020. Machine vision for natural gas methane emissions detection using an infrared camera. *Appl. Energy* 257, 113998.
- Wang, J., Ji, J., Ravikumar, A.P., Savarese, S., Brandt, A.R., 2022. Videogasnet: deep learning for natural gas methane leak classification using an infrared camera. *Energy* 238, 121516.
- Williams, R., Kilaru, V., Snyder, E., Kaufman, A., Dye, T., Rutter, A., Russell, A., Hafner, H., 2014. Air Sensor Guidebook. US Environmental Protection Agency.

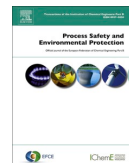
#### **5.1.4. Fully automated method to estimate opacity in stack and fugitive emissions: A case study in industrial environments**

- Pedrayes, O. D., Usamentiaga, R., & García, D. F. (2023). *Fully automated method to estimate opacity in stack and fugitive emissions: A case study in industrial environments*. *Process Safety and Environmental Protection*, 170, 479-490.
- DOI: [10.1016/j.psep.2022.12.023](https://doi.org/10.1016/j.psep.2022.12.023)
- El índice de impacto de la revista *Process Safety and Environmental Protection* en 2021 fue 7.926 (Q1, 85.66 %) y el índice de impacto a 5 años, 7.717.



Contents lists available at ScienceDirect

# Process Safety and Environmental Protection

journal homepage: [www.journals.elsevier.com/process-safety-and-environmental-protection](http://www.journals.elsevier.com/process-safety-and-environmental-protection)

## Fully automated method to estimate opacity in stack and fugitive emissions: A case study in industrial environments

Oscar D. Pedrayes, Rubén Usamentiaga<sup>\*</sup>, Daniel F. García

Department of Computer Science and Engineering, University of Oviedo, Campus de Viesques, Gijón 33204, Asturias, Spain

## ARTICLE INFO

## Keywords:

DeepLab  
Deep learning  
Pollution  
Semantic segmentation  
Smoke

## 2000 MSC:

0000  
1111

## PACS:

0000  
1111

## ABSTRACT

Fugitive emissions are those that are unplanned, i.e., they do not come out of a stack. These emissions are usually disperse and difficult to locate. By estimating the opacity of fugitive emissions they can be controlled or even prevented, helping to comply with environmental regulations. Most opacity estimation methods are based on stack emissions, which are straightforward, as they are always located in the same area. All methods in the literature for emission opacity estimation require a human operator to select the regions to be used as a reference. In this work, deep learning networks are proposed to improve the quality and automation of this process by selecting the regions completely and automatically. Furthermore, a new fugitive emission opacity estimation method is proposed. This method, called SBPB, is compared with other relevant methods in the literature, offering a solution with an average F1-Score metric 5 % higher than other methods on two real datasets with over 4000 images in total. This method provides a robust solution for fugitive emissions.

### 1. Introduction

A system capable of estimating the severity of an emission in real-time would be of great help in preserving the environment. Such a system could act in function of the severity of the emission, generating an alarm to solve the problem as quickly as possible. Moreover, it would comply with regulations that require industrial plants to monitor the severity of their emissions according to their size and opacity. To calculate the severity of an emission, one of the most important characteristics is opacity: the focus of this study.

Initially, emission plume opacity was assessed by visual comparison of the plume with Ringelmann charts (Ringelmann and Kudlich, 1967). These have five density reference levels inferred from a grid of black lines on a white surface, corresponding to different opacity values (Randolph and Foster, 1993). Later, the Method 9 standard (Randolph and Foster, 1993) was created to train human observers. Method 9 is a standard that details a method designed by the Environmental Protection Agency (EPA) of The United States to guide human observers in quantifying plume opacity in daylight conditions. Today, the most common way to obtain the opacity of a plume is still by human observers, usually trained with Method 9.

Standards and algorithms to determine the opacity of a plume do

exist. ASTM D7520 () is a testing standard to determine the opacity of visible emissions using a digital camera and analysis software, known as Digital Camera Opacity Technique, or DCOT. ATSM D7520–09 was approved in 2009 by the U.S. Department of Defense (DOD). This standard aims to establish a minimum level of performance for products using DCOT to determine plume opacity outdoors. EPA Alternative Method 082 (ALT-082) is a standard (Dolan, 2017) approved in 2011 as an alternative to EPA Method 9 and adds limitations to the ASTM D7520 specification. The objective of ALT-082 is to determine the accuracy and reliability of a visual opacity monitoring system consisting of a conventional digital camera and a stand-alone software application to determine plume opacity. This standard was designed for the opacity analysis software Digital Opacity Compliance System (DOCS) (McFarland et al., 2004, 2007, 2010). There is also an updated version known as Digital Opacity Compliance System Second Generation (DOCS II) (Rasmussen and Grieco, 2009).

Some algorithms simulate the EPA Method 9 standard using two cameras. One of them is focused on the background taken as a reference and the other on the plume (Lighty et al., 2007). Other algorithms, such as the Digital Opacity Method (DOM) (Du et al., 2007), rely on a physical model of a reference background. In this case, DOM uses only one camera, so obtaining the reference is more complicated. If the reference is a contrasting background, it is called the Contrast model, while if the

<sup>\*</sup> Corresponding author at: University of Oviedo Department of Computer Science and Engineering, Campus de Viesques, Gijón, 33204, Asturias, Spain.  
E-mail address: [rusamentiaga@uniovi.es](mailto:rusamentiaga@uniovi.es) (R. Usamentiaga).

<https://doi.org/10.1016/j.psep.2022.12.023>

Received 17 August 2022; Received in revised form 8 December 2022; Accepted 8 December 2022

Available online 9 December 2022

0957-5820/© 2022 The Author(s). Published by Elsevier Ltd on behalf of Institution of Chemical Engineers. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

## Nomenclature

**ALT-082** EPA Alternative Method 082.

**ASPP** Atrous Spatial Pyramid Pooling.

**ATSM** American Society for Testing and Materials.

**CNN** Convolutional Neural Network.

**DCOT** Digital Camera Opacity Technique.

**DOCS** Digital Opacity Compliance System.

**DOD** Department of Defense.

**DOM** Digital Optical Method.

**EPA** Environmental Protection Agency.

**FN** False Negative.

**FP** False Positive.

**GPU** Graphics Processing Unit.

**HSV** Hue, Saturation, Value.

**IoU** Intersection over Union.

**P** Precision.

**PCA** Principal Component Analysis.

**R** Recall.

**RGB** Red, Green, Blue.

**SBPB** Sky and Building Percentiles in the Blue channel.

**SGDM** Stochastic Gradient Descent with Momentum.

**TN** True Negative.

**TP** True Positive.

reference background is the sky itself, it is called the Transmission model. Using the Transmission model, Yuen et al. (Yuen et al., 2017) and Yuen et al. (Yuen et al., 2018) add new references to the sky reference: a reference to the background where the emission is dark, usually because of a building; a reference where the emission is lighter, the sky in the background; and a reference to the darker part in the background, the building itself.

In opacity estimation algorithms, a human operator usually selects the references by selecting boxes with the regions to be used. If these regions are obtained automatically, the whole process of opacity calculation can be automated. In (Prakasa, 2017), an algorithm for opacity calculation using regions obtained by k-nearest neighbors, and then revised by an operator, is proposed.

Opacity estimation algorithms are usually applied to plumes but not to more diffuse emissions. Fugitive emissions are emissions that are not generated by a stack, i.e., emissions that are not planned. For this reason, characterizing the severity of these emissions by calculating the opacity is essential before taking proportional actions. Fugitive emissions are commonly caused in industrial plants by failures in the production, processing, transmission, storage, and use of fuels. These emissions can severely pollute the environment, endanger the lives of people and animals in the area, and contribute to the greenhouse effect (Laconde, 2018). For example, carbon dioxide emissions have an impact approximately one order of magnitude less than methane emissions from the oil and gas industries (Solomon et al., 2007).

There is very little work in the literature on opacity estimation of fugitive emissions. DOCS has an algorithm for fugitive emissions which obtains an image before the emission occurs and an image while the emission exists to compare the two. Nevertheless, it requires a human operator to establish the references. This approach is not viable for a fully automated system, as it is impossible to know how long a fugitive emission may last. This results in images that may have excessively different light or weather conditions. The ideal method would be able to calculate the opacity of fugitive emissions from a single static image without the intervention of a human operator.

This paper compares different methods to estimate emission opacity and proposes a method, SBPB (Sky and Building Percentiles in the Blue channel), to determine fugitive emission opacity (Laconde, 2018). Unlike other methods, the proposed method is fully automated, so no human operator intervention is required. One of its major advantages is its robustness, enabling the use of uncalibrated cameras with self-adjusting exposure time. For input, a single image is needed, in which the opacity of each pixel is characterized separately. This approach provides more information about the emission than if only one numerical value were used to represent the opacity of the whole emission, which is especially important for fugitive emissions due to their non-uniform nature. For this purpose, a semantic segmentation model capable of obtaining the regions of fugitive emissions, buildings, and the sky is first trained. Then, the sky and building regions are used as a reference to calculate the opacity of fugitive emissions so that a

particular opacity value characterizes each emission pixel. A value for the total emission can be obtained by selecting a percentile. This value is used to make a numerical as well as a visual comparison with other methods in the literature. To make fair comparisons between the different methods, the same mask obtained from the semantic segmentation network is used as the basis for pixel selection in all the methods.

## 2. Methods and materials

### 2.1. Datasets

The images used in this study were obtained from two different industrial plants through surveillance cameras. The images were taken on sunny days, partially cloudy days, and very cloudy days. There are no images of heavy fog or rainy days. All of the images were taken in the summertime between dawn and dusk. Night-time images are not included because, as the area is not artificially lit, it is impossible to see the emission after dusk. The images of the datasets were provided by the owner of the industrial plants. In order to maintain their anonymity, the regions of the images belonging to the facilities have been censored using a Gaussian blur and a stripe mask. This censorship applies only to the images shown in this paper, not those for training or use.

Experts from the companies' industrial plants use on-site visual analysis to classify opacity. Datasets from two plants, totalling 4287 images were classified into three levels of opacity: Low, Medium, and High. Of the 2150 images from the first plant, 1400 were Low, 597 Medium, and 153 High. Of the 2137 images from the second plant, 1294 were Low, 593 Medium, and 150 High. Example images can be seen in Fig. 1.

The channels of these images consist of the red, green, and blue (RGB) bands and have a resolution of  $2048 \times 1536$  pixels or  $1024 \times 768$  pixels. To meet the semantic segmentation network requirements, all images are scaled to a constant smaller size using one of the most common interpolation algorithms: Bicubic interpolation. This reduces their computational cost and required VRAM usage. The best size is  $512 \times 384$  because it maintains its aspect ratio and keeps the highest manageable resolution.

The ground truth for semantic segmentation training has the following classes: building, vapour (water vapour from chimneys), cloud, fire, fugitive emission, and sky.

### 2.2. Processing pipeline

In this paper, a new processing pipeline is proposed to fully automate the opacity estimation process. In this pipeline the opacity estimation algorithm can be interchanged following a loosely coupled approach, so that the different opacity estimation methods can be easily compared.

Fig. 2 shows a diagram of the processing pipeline. In this pipeline, the image is fed to a semantic segmentation convolutional neural network which outputs a prediction mask with the location of the emission. This

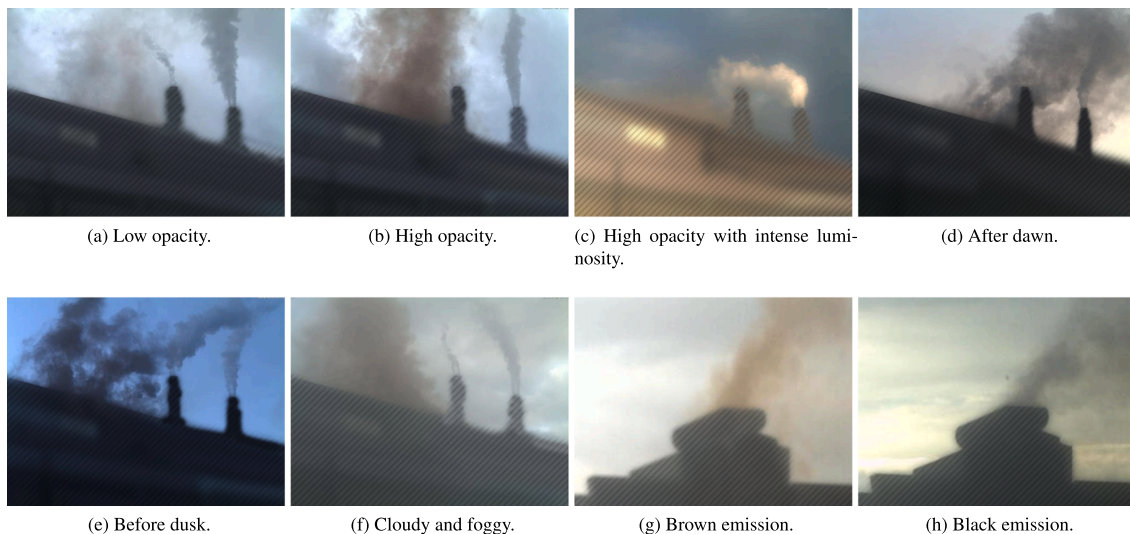


Fig. 1. Examples from the datasets.

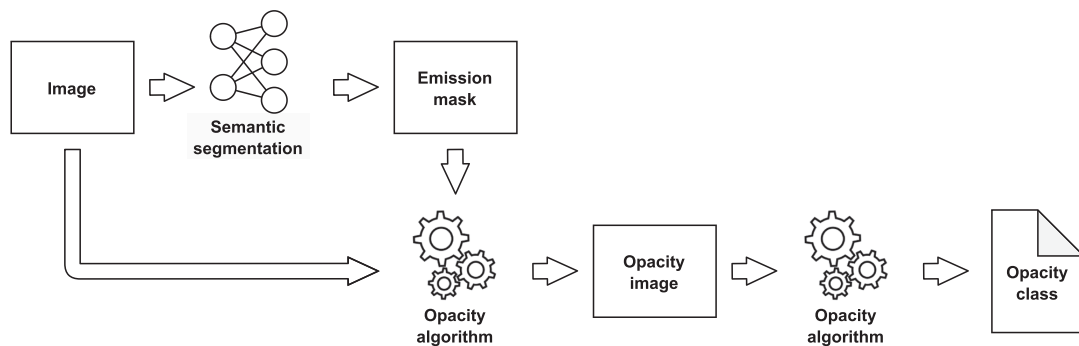


Fig. 2. Processing pipeline.

mask, along with the original image, is fed to the opacity estimation algorithm to calculate the opacity image, in which each pixel shows its opacity from 0 % to 100 %. Finally, this opacity image is used by a simple yet effective classification algorithm to determine the class of the emission opacity in order to calculate its metrics.

### 2.3. Semantic segmentation

Semantic segmentation networks are a type of convolutional neural network used for pixel classification. Pixel classification is useful to detect and localize objects or regions in images, which can be used to isolate the desired pixels. All convolutional neural networks must be trained before they can be used. To train a model, a groundtruth mask and the original image are needed. This groundtruth mask is necessary to calculate the errors made by the network and to correct them. Once the training process is completed, the prediction mask can be obtained when an image is fed to the network.

Convolutional neural networks have hyper-parameters to control their behavior during the training process. For this reason, in order to obtain optimal results, it is necessary to study the performance of the hyper-parameters. Since each modification in the hyper-parameters requires new training, this process can be extremely tedious and time

consuming.

#### 2.3.1. Network architecture

DeepLab (Chen et al., 2014, 2017a; Chen et al) is a convolutional neural network for semantic segmentation developed by Google. The latest version is called DeepLabv3 + (Chen et al., 2018). The architecture of the network combines an Atrous Spatial Pyramid Pooling (ASPP) module with a common encoder-decoder structure. In this paper, the official Tensorflow implementation from Google’s Github<sup>1</sup> is used.

#### 2.3.2. Metrics

All metrics are based on the concepts of true positive (TP), false positive (FP), true negative (TN) and false negative (FN). TP are pixels correctly classified as the target class, TN are pixels correctly classified as belonging to other classes, FP are pixels wrongly classified as the target class, and FN are pixels wrongly classified as belonging to other classes.

Precision is calculated as the percentage of correctly classified pixels from the total predicted pixels, as shown in Equation (1). Recall is

<sup>1</sup> <https://github.com/tensorflow/models/tree/master/research/deeplab>

**Table 1**  
Training parameters for DeepLabV3+.

Training Parameters	
Input size	512 × 384 × 3
Classes	6
Backbone	Xception65
Output stride	16
Padding	Yes
Solver	Adam
Epochs	80
Batch size	6
Learning rate	0.00005
Class weighting	Median Frequency Weighting
Gradient clipping	No
L2 regularization	0.0004
Data augmentation	Scale 0.5–2.0 with 0.25 steps
Shuffle	Yes

calculated as the percentage of pixels classified correctly from the pixels that correspond to that particular class in the ground truth, as shown in Equation (2). If Precision is low and Recall is high, the predictions will overclassify pixels from that particular class. If Recall is low and Precision is high, only the pixels with high confidence will be classified as belonging to that particular class.

$$P = \frac{TP}{TP + FP} \quad (1)$$

$$R = \frac{TP}{TP + FN} \quad (2)$$

F<sub>1</sub>-Score is one of the most common metrics in semantic segmentation. It is calculated as a combination of Precision and Recall as shown in Equation (3), and is equivalent to the Dice Coefficient with two classes.

$$F_1 = \frac{2 \times P \times R}{P + R} \quad (3)$$







Intersection-over-Union or IoU is equivalent to the Jaccard Index. This metric measures the area of similarity of a segmentation and its ground truth. It is calculated as the proportion between True Positives and the sum of True Positives, False Negatives, and False Positives, as shown in Equation (4),

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} = \frac{TP}{TP + FN + FP} \quad (4)$$

### 2.3.3. Training

Before using the network, it must be trained. To obtain the best possible results, its hyper-parameters must be configured. Each time a

**Table 2**  
Metrics of the best model test.

Class	Precision	Recall	IoU	F <sub>1</sub>
 Building	0.995	0.997	0.992	0.996
 Vapour	0.915	0.900	0.831	0.907
 Clouds	0.955	0.834	0.802	0.890
 Fire	0.794	0.837	0.688	0.815
 Emission	0.836	0.832	0.715	0.834
 Sky	0.926	0.951	0.884	0.938

hyper-parameter is modified, the network must be completely retrained to observe its effects on the results of its new model. As a starting point, to accelerate the convergence of the model, the training uses a pre-trained model<sup>2</sup> by Google on the Imagenet dataset (Deng et al., 2009)

Each hyper-parameter was adjusted individually because testing all possible combinations of hyper-parameters with a single computer would have been excessively time consuming. The final configuration used in this paper can be seen in Table 1. The GPU used to train the model was an NVIDIA RTX 2080 TI GPU with 11 GB of VRAM.

The following hyper-parameters were tuned: input sizes (2048 × 1536, 1024 × 768, 512 × 384, and 256 × 192); batch sizes; learning rate; epochs; output strides of 8, 16, and 32; different backbone networks (Resnet50, Xception45, Xception65, Xception71, MobileNetV2, MobileNetV3Small, MobileNetV3Large); L2 regularization; and solver algorithms (Adam or Stochastic Gradient Descent with Momentum (SGDM)).

To prevent overfitting, the dataset was shuffled to each epoch and data augmentation was used to obtain more training data. The augmentation process consisted of zooms of images with varying zoom values ranging from 50 % to 200 % at intervals of 25 % increments.

Pedrayes et al. (Pedrayes et al., 2022) provides additional information regarding the training process and its hyper-parameters for the datasets evaluated in this study.

Results of the test for the best model with data not seen by the network can be seen in Table 2. The F<sub>1</sub>-Score for all classes is over 80 %, which gives high confidence in the prediction of the different regions. In particular, the Building, Vapour, and Sky classes are above 90 %. In Fig. 3 examples from all classes are shown. The first column shows the original image (with the building censored in red to maintain the anonymity of the company that provided the data), the second column shows the groundtruth, the third column shows the predictions, and the fourth and fifth columns show the groundtruth and the predictions overlaid on the original image. Almost all predictions are visually identical to the ground truth.

### 2.3.4. Regions

The sky pixels closest to the emission are used for those methods that require the sky reference to be obtained. For this purpose, the emission region is expanded and those pixels belonging to the sky class are selected. The number of sky pixels selected is equal to the number of emission pixels. Thus, emission and sky regions of the same size are obtained. For the process of selecting sky pixels, the pixels closest to the emission are discarded to avoid errors in the labeling of the regions. To do this, the fugitive emission mask is dilated once using a cross-shaped structure of 3×3 pixels. This value is used for both datasets, however, this gap can be increased depending on camera location, emission type

<sup>2</sup> [https://github.com/tensorflow/models/blob/master/research/deeplab/g3doc/model\\_zoo.md](https://github.com/tensorflow/models/blob/master/research/deeplab/g3doc/model_zoo.md)



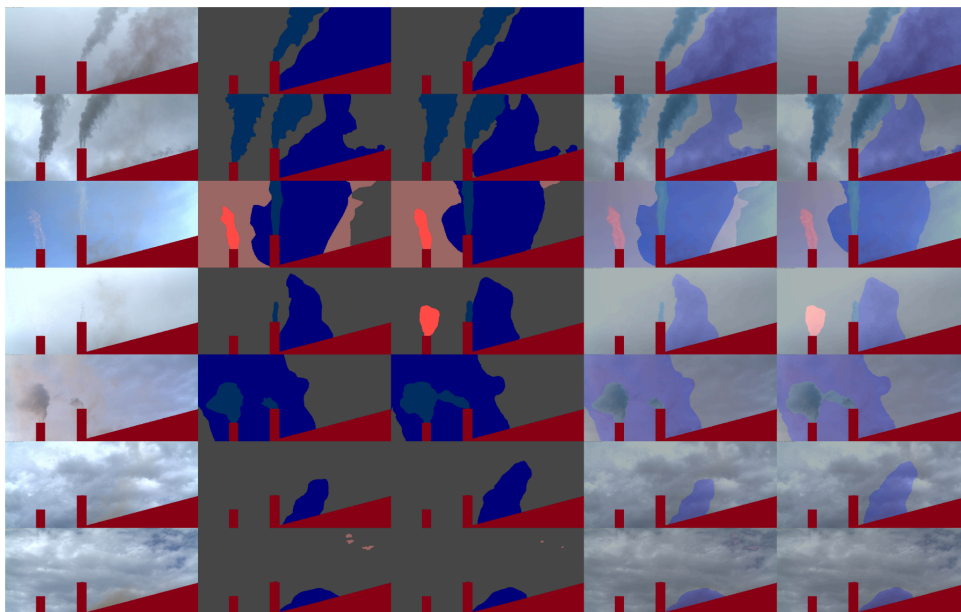


Fig. 3. Predictions (Building censored). Original image (1st column), ground truth (2nd column), prediction (3rd column), ground truth overlay (4th column), and prediction overlay (5th column).



Fig. 4. Selection of the sky region.

Table 3  
Opacity estimation methods.

Method	Citation	Type	Per pixel	Process
Ringelmann	(Ringelmann and Kudlich, 1967)	Stack	Yes	Manual
DOM (Transmission model)	(Du et al., 2007)	Stack	No	Manual
Prakasa et al.	(Prakasa, 2017)	Stack	Yes	Manual/ Assisted
Yuen et al.	(Yuen et al., 2017)	Stack	No	Manual
DOCS	(Rasmussen and Grieco, 2009)	Stack	Yes	Manual
Transmittance	(Randolph and Foster, 1993)	Stack	Yes	Manual
SBPB	Proposed	Stack/ Fugitive	Yes	Automatic

or accuracy of the model. Intuitively, it is clear that selecting the near-sky pixels rather than all sky pixels gives better results because of the comparison with the sky behind the emission, improving on operators that select arbitrary square bounding boxes.

A visual explanation can be seen in Fig. 4. The emission is shown in grey, the sky in dark blue, and the building in red. The white contour is the gap of the ignored sky pixels, and the light blue is the sky pixels used to calculate the opacity estimation.

#### 2.4. Opacity estimation methods

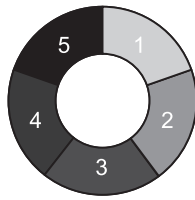
This study evaluates all methods for opacity estimation in emissions published in the literature. The Ringelmann method is well-known in the literature, despite its age. The Prakasa et al. method is of little relevance in the literature but introduces an original approach. The DOM (Transmission Model), Yuen et al., and DOCS methods, on the other hand, are relatively recent and important in the literature. Finally, the Transmittance method is based on a well known equation. From the study of these methods, a new opacity estimation method called SBPB has been developed Table 3.

All methods in the literature are manual or, at most, assisted processes. In this paper an approach which allows the automatic use of any method thanks to the use of semantic segmentation is proposed. In order to make a fair comparison of all the methods, the extraction of regions from all of them is automated by means of semantic segmentation. This is due to the large number of images in the datasets, which makes manual testing impossible.

##### 2.4.1. Method: ringelmann

The Ringelmann chart method is a rough adaptation of the first methodology based on a visual comparison with a chart. In this case, instead of a visual evaluation, the intensity of the emission converted to grayscale using the BT.709 recommendation (Series, 2017) is compared with the intensity of each chart reference (see Fig. 5a).

The pixel value 255 corresponds to 100 % luminous intensity, while 0 corresponds to 0 %. Fig. 5b shows the intensity values of the



(a) Chart (Circle type).

Ringelmann scale	Light transmission (%)	Opacity (%)
0	[100,90)	[0,10]
1	[90,70)	(10,30]
2	[70,50)	(30,50]
3	[50,30)	(50,70]
4	[30,10)	(70,90]
5	[10,0]	(90,100]

(b) Intensity values per scale.

Fig. 5. Ringelmann method.

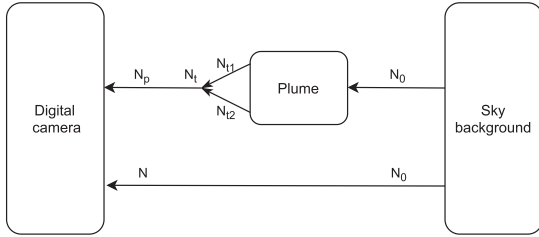


Fig. 6. DOM: Transmission model diagram.

Ringelmann scales. A major disadvantage of this method is that it is image-dependent, so two emissions with the same opacity level may differ depending on the scene lighting and camera calibration. This is because no reference is used to adjust the opacity calculation.

2.4.2. Method: DOM (Transmission model)

In the DOM (Transmission model) method, the opacity of the plume is determined by Equation (5).  $N_p$  refers to the value of the emission pixels. To calculate this value the average of all “emission” pixels is taken.  $N$  is the value of the background pixels. To calculate this value the average of all “sky” pixels is taken.  $K$  is a coefficient that depends on the transmissivity of the particles and the environment. According to (Du et al., 2007) and the DOM patent (Kim et al., 2009), a  $K$  value of 0.16 is recommended for black plumes and 1.4 for white plumes. This method is designed for uniform backgrounds. The dataset used contains dark emissions, thus a value of 0.16 is set for all images Fig. 6.

$$O = \frac{1 - \frac{N_p}{N}}{1 - K} \tag{5}$$

2.4.3. Method: Prakasa et al

The method described by (Prakasa, 2017) first divides the image into several rows containing the same plume, so that each row will use reference values contained in the same row. This is done because it is assumed that the higher the elevation from the plume, the lower the emission density is. For this reason, the sky intensity values for a given row are averaged. Fig. 7 shows an example of a separation in rows.

Opacity is determined by comparing the color difference of the plume with the sky background and the maximum color difference. The maximum value is obtained by assuming that the color of the plume is pure black. Therefore, all values in the RGB channel will be zero to represent an all-black intensity. This maximum value can be considered as a reference to quantify the level of opacity.

Opacity is calculated for each pixel in the region individually. The intensities of neighboring pixels do not influence the calculation of the opacity of an observed point. Eqs. (6–8) are used to obtain the opacity value.

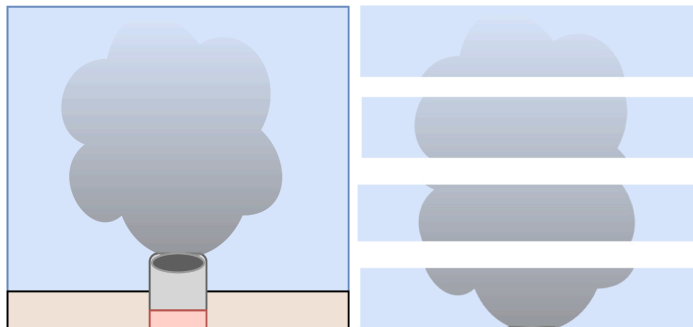
$I_p$  is the value or intensity of the RGB band for the pixels belonging to the plume or emission.  $I_s$  are the pixels belonging to the linear fit representing the sky for the RGB bands.

$$d_{RGB} = \sqrt{(I_p - I_s)_R^2 + (I_p - I_s)_G^2 + (I_p - I_s)_B^2} \tag{6}$$

$$d_{Ref} = \sqrt{I_{p,R}^2 + I_{p,G}^2 + I_{p,B}^2} \tag{7}$$

$$O = \frac{d_{RGB}}{d_{Ref}} * 100\% \tag{8}$$

Equation (8) divides the value of the difference between the intensity of the emission pixel ( $I_p$ ) and the intensity of the sky on the vertical axis of the region ( $I_s$ ) by the intensity of the emission itself ( $I_p$ ). This appears to be incorrect based on the physical equation for calculating opacity ( $Opacity = 1 - I/I_0$ , where  $I$  is the flux of light returning from the emission and  $I_0$  is the flux of incident light without passing through the



(a) Original image.

(b) Division of the plume in several rows.

Fig. 7. Prakasa diagram.

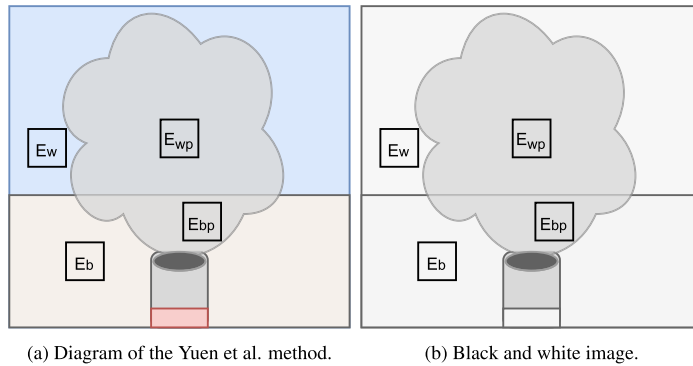


Fig. 8. Yuen et al.

emission), it should be divided by the intensity of the sky ( $I_s$ ). The difference in intensities is normalized for each pixel separately. This normalization causes the opacity resulting from  $I_p$  values close to  $I_s$  to be minimized, while using values with a larger difference between  $I_p$  and  $I_s$ , the opacity value is maximized. It occurs because after dividing by  $I_p$ , when this is a very low intensity value, the value by which it is divided is also very low, and a division by 0 can occur if the  $I_p$  value is completely black. Similarly, when the two values are close, their difference, i.e. the numerator, will have a lower value. Also, in this case, the denominator will have a higher value, resulting in a much reduced opacity value. The major disadvantage of this method is that by visually representing a mask with opacity values, borders can be seen when dividing the image into regions. This method is not designed to create masks, but to obtain opacity plots along the vertical axis. However, masks are generated to provide a comparison with the rest of the methods.

2.4.4. Method: Yuen et al

The method described by (Yuen et al., 2017),(Yuen et al., 2018) splits the DOM (transmission model) method references according to the intensity of their background. The DOM (transmission model) uses only one reference for the sky and another for the emission. The Yuen et al. method uses two references for sky and two for emission. One of each pair of references is in a zone with higher intensity and the other in a zone with lower intensity. This method requires the BT.709 recommendation to convert RGB values to intensity values in a grayscale format.

This is done using Equation (9) where  $O$  is the opacity of the plume.  $E_{wp}$  is the amount of exposure caused by the bright background with plume.  $E_w$  is the amount of exposure caused by the bright background without plume.  $E_{bp}$  is the amount of exposure caused by the dark background with plume, and  $E_b$  is the amount of exposure caused by the dark background without plume. It can be understood visually, as shown in Fig. 8a.

$$O = 1 - \frac{E_{wp} - E_{bp}}{E_w - E_b} = 1 - \frac{E_{wp} - E_{bp}}{E_w - E_b} \tag{9}$$

2.4.5. Method: DOCS

For evaluation purposes, the DOCS method has been implemented in this work following its patent indications (Pfaff and Stretch, 2003). However, this version might not match the official implementation exactly because it leaves certain development aspects unexplained. In (Pfaff and Stretch, 2003; McFarland et al., 2004, 2010) it is explained that RGB is used for the whole process, although HSV can be used.

The first step of the DOCS method is to smooth out the image to eliminate or reduce visual artifacts (see (Pfaff and Stretch, 2003)). The second step is to apply the PCA algorithm to the region of interest of the smoothed RGB image. This reduces the dimensionality of the data from three channels to one channel. The values obtained after using the PCA method represent the color intensity variability of the R, G and B bands. For this process only the first component of the PCA is used. For simplicity it will be called PCA1.

The third step is to calculate the opacity of the plume. The negative values of the PCA1 representation are directly related to the opacity through a linear relationship. In order to avoid images generating completely different results, this linear relationship is established on the complete dataset using a minimum and a maximum value. The minimum value is obtained by taking the 5th percentile of all the minima of each image in the dataset. The minima of each image is obtained using the 1st percentile to eliminate outliers. The maximum is obtained in the same way but using the 95th for the maxima and the 99th percentile to eliminate outliers.

2.4.6. Method: Transmittance

The Transmittance method is based on the transmittance formula (Randolph and Foster, 1993) shown in Eq. (10). As shown in Fig. 9,  $I$  is the intensity of the plume and  $I_0$  the intensity of the sky. It is common for algorithms to follow a variation of this equation. For example, the transmittance method is similar to the DOM (Transmission model) method, but without the  $K$  value in this case. The purpose of the Transmittance method is to follow the physical model equation as simply as possible.

Intensity values are obtained using the BT.709 recommendation to transform the RGB image into intensity values in grayscale format.  $I_0$  is calculated as the median of the sky region. The median is stronger than the average against outliers caused by reflective surfaces or possible artifacts in the image.  $I$  is the value of a particular pixel of the emission.

$$O = 1 - \frac{I}{I_0} \tag{10}$$

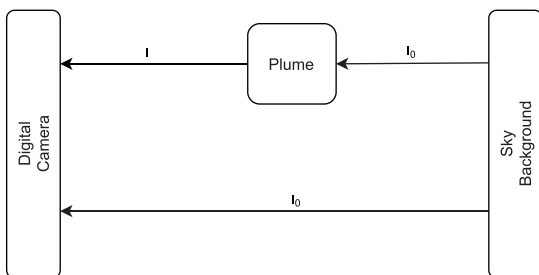


Fig. 9. Transmittance method diagram.

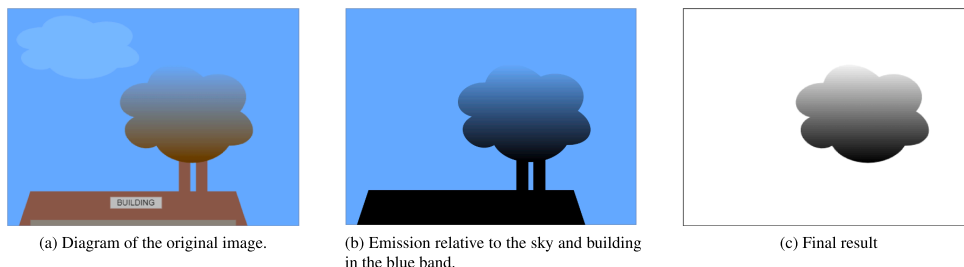


Fig. 10. SBPB method diagrams.

2.4.7. Method: sky and building percentiles in the blue channel (SBPB) - proposed

In the methods described above, the emission must be black. With this method, the emission can be of others colors such as brown or yellowish, except blue (see Fig. 10a). Thus, because the sky/clouds generally have a blue hue, it is assumed that the more blue a pixel value is, the less opaque it is. Conversely, the lower its blue value, the more opaque. After some testing with the RGB bands separately and the grayscale using the BT.709 algorithm, all of them had trouble when sunlight was shining directly on the building. However, as the B-band is robust against this, it was decided to use only the B-band of the RGB images for opacity calculation.

This method establishes that the region segmented as Building has an opacity of 100 % and that the region segmented as Sky has an opacity of 0 %. To use this method, the building used as a reference cannot be blue (see Fig. 10b).

Based on this assumption, the 75th percentile value of the building region, and the 25th of the sky region is obtained. These percentiles provide robust reference values for the building and the sky, preventing outliers which may be caused by artifacts in the image or reflective surfaces (Vinutha et al., 2018). These intensity values can then be used to adjust the opacity calculation to the light conditions of the image. This enables the use of images obtained from dynamically self-adjusted or poorly calibrated cameras.

Given that the building has an opacity of 100 %, emission values equal to or lower than that of the building represent an opacity of 100 %.

Those equal to or higher than the the sky region, have an opacity of 0 %. The remaining pixels, i.e., those between the two limit values, are normalized between 0 % and 100 % using these limit values, so that all emission pixels have values between 0 % and 100 % (see Fig. 10c).

3. Results and discussions

This section presents the most relevant examples for the comparison of the different methods. Figs. 11, 12, and 13 show the original image with the fugitive emission, and the results of the opacity estimation of the different methods in visual form where black is maximum opacity and white no opacity. Images with the opacity estimation show the values in the range [0,1] for each of the pixels.

Figure 11 shows that the proposed SBPB method is the only one which correctly classifies the opacity of the left central part of the emission. The Prakasa et al., DOCS, and Transmittance methods give that part of the emission an excessively low opacity level.

In situations with low intensity opacity (see Fig. 12), a larger difference between the methods is observed. This is caused by the higher complexity of the image due to lower contrast and higher confusion with the background clouds. The Ringelmann method values the pixels of the region by their intensity without taking into account the context of the image, causing these cases to estimate a higher opacity than expected. The DOM (Transmission model) method is not able to characterize the opacity of the emission. The Yuen et al. method overestimates the opacity in transition areas between emission and sky. The DOCS and Transmittance methods significantly improve the transition areas of the

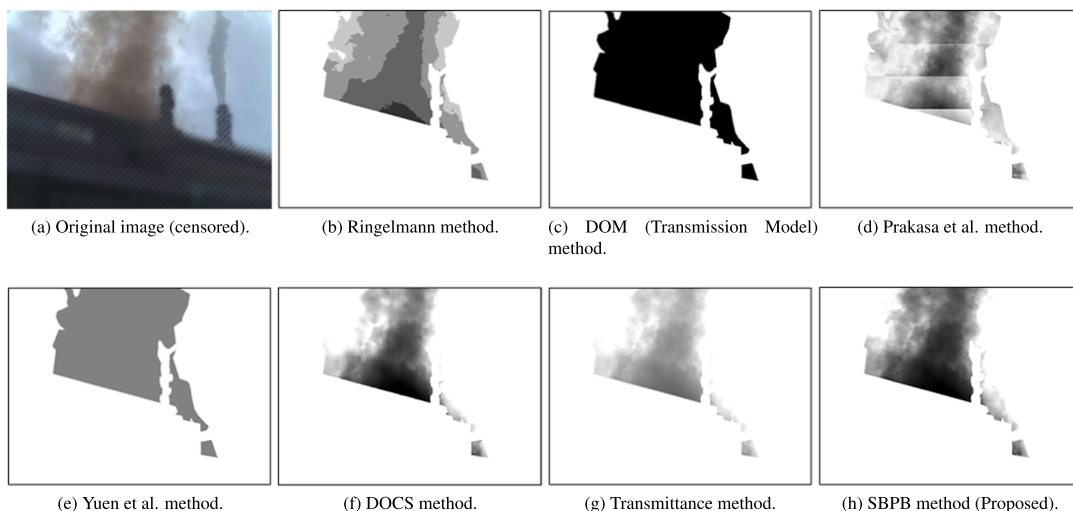


Fig. 11. High opacity emission from the first dataset.

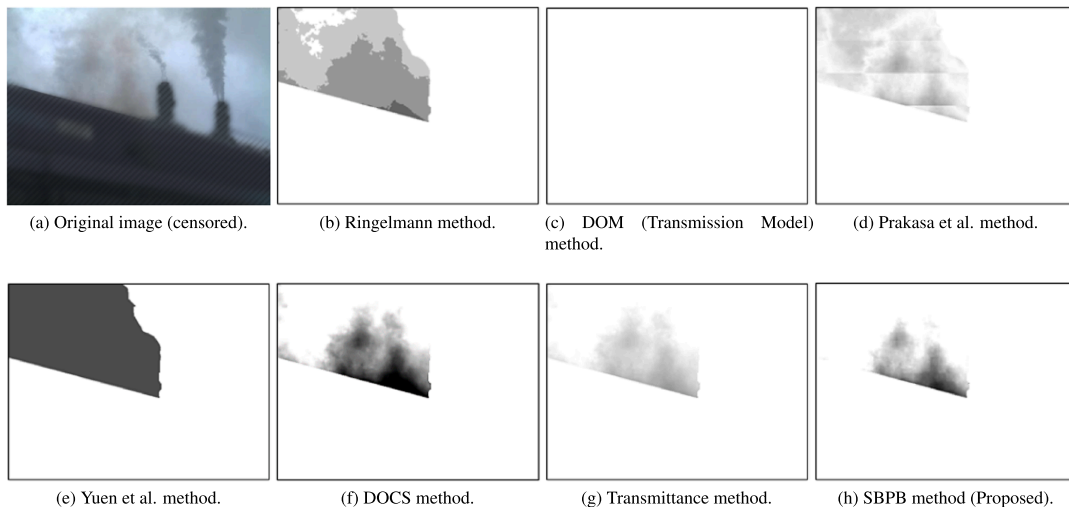


Fig. 12. Low opacity emission from the first dataset.

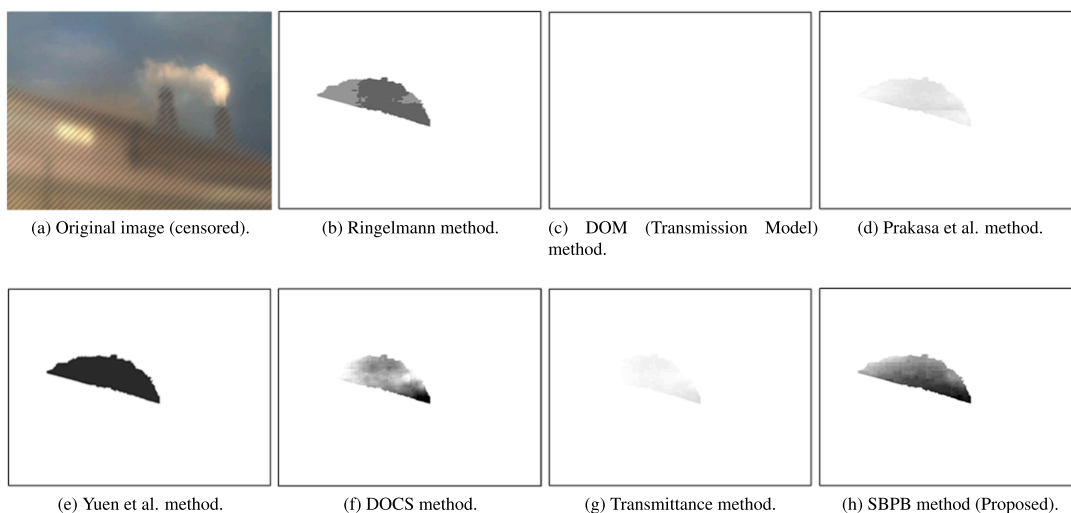


Fig. 13. Intense luminosity opacity emission from the first dataset.

emission region, however, DOCS overestimates emission opacity. Finally, the proposed SBPB method further adjusts the emission region and provides opacity values closer to those expected.

Figure 13 shows that the DOM (Transmission model), Prakasa et al., and Transmittance methods are not able to determine the opacity of the emission correctly. In all these methods the emission is characterized at a very low opacity level. This is due to the fact that this image has a high light intensity, giving the building a high intensity. For this reason, the auto-tuning of the camera causes the sky to darken so that the sky is even darker than the building. However, the proposed SBPB method is able to characterize the emission opacity correctly. This is because, even though the luminous intensity is incorrect due to the camera autotuning, the sky still has a higher blue hue than the building. In this case, the Ringelmann method is not affected by the sky and the building, so it obtains its usual results. As the DOCS method is based on the variance of the image itself, it has fewer problems in characterizing the emission. Finally, the Yuen

et al. method values the complete emission at the same time but is able to characterize this emission in a more reasonable way, which may be due to the use of multiple building and sky references.

Figure 14 shows the result of the opacity estimations of all methods for one of the images of the second dataset of another industrial plant. Here it can be seen that it performs similarly to the rest of the images of the first dataset. The main difference is that in this particular case the DOCS method seems to obtain similar results to the proposed SBPB method. These two methods are the ones that best estimate the opacity of the emission.

Methods that obtain a value for the whole emission, as is the case for DOM (Transmission model), and Yuen et al., are notably affected when the emission has very differentiated parts. That is, if half of the emission has a low opacity and the other half has a high opacity, the result is affected by obtaining an average value. If an operator selected a bounding box, the result would be completely dependent on where the

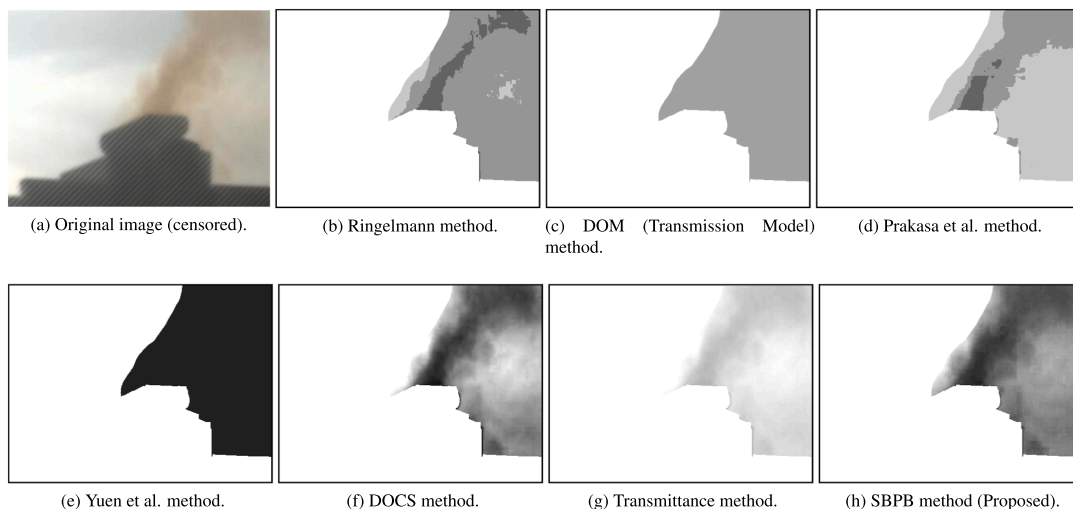


Fig. 14. High opacity emission from the second dataset.

Table 4  
F<sub>1</sub>-Score of the methods for the first dataset.

Method	Class			Average
	Low opacity	Medium opacity	High opacity	
Ringelmann	0.254	0.402	0.322	0.326
DOM (Transmission model)	0.663	0.336	0.415	0.471
Prakasa et al.	<b>0.804</b>	0.071	0.208	0.361
Yuen et al.	0.391	0.000	0.068	0.236
DOCS	0.586	0.019	0.125	0.243
Transmittance	0.747	0.413	0.478	0.546
<b>SBPB (Proposed)</b>	0.775	<b>0.456</b>	<b>0.540</b>	<b>0.590</b>

box were placed. For this reason, the automatic selection proposed in this work yields more reliable results despite the area weight of the different opacity levels. This is much more important for those methods in which the opacity is calculated for each pixel.

If the building has a higher light intensity than the sky, the methods that use the sky and the building as references may behave erroneously, as is the case for the DOM (Transmission model) and Yuen et al. methods. This is usually due to luminance situations which cause the camera autotuning to become very aggressive. In these cases, the proposed SBPB method achieves satisfactory results because the sky still has higher luminous intensity than the building in the B-band. Thus, the proposed method is much more robust in situations where the illumination is not perfect.

Table 5  
F<sub>1</sub>-Score of the methods for the second dataset.

Method	Class			Average
	Low opacity	Medium opacity	High opacity	
Ringelmann	0.609	0.000	0.185	0.264
DOM (Transmission model)	0.769	0.333	0.361	0.488
Prakasa et al.	<b>0.792</b>	0.224	0.000	0.339
Yuen et al.	0.554	0.000	0.107	0.230
DOCS	0.592	0.053	0.128	0.258
Transmittance	0.720	0.274	0.309	0.434
<b>SBPB (Proposed)</b>	0.753	<b>0.342</b>	<b>0.407</b>	<b>0.500</b>

In addition to the visual examples, Table 4 presents the information of the results of each method for the first dataset. Opacity images from the different opacity estimation algorithms are processed by a classification algorithm to obtain their categorical opacity level to be compared with the groundtruth generated by human operators. This classification algorithm discards opacity values lower than 5 % because they are the noisiest (Pfaff and Stretch, 2003), and calculates the 80th percentile of the remaining emission pixels to calculate a single opacity value for the whole emission. This is necessary because human visual perception of brightness is non-linear (McNamara et al., 2000). This single opacity value is used to determine its class (Low, Medium, or High) by using thresholds. These thresholds are calculated separately for each method in order to maximize the separation between classes for a fairer and more generalizable comparison. To calculate them, the midpoint between the median values of the adjacent classes is calculated. In other words, the threshold between the Low class and the Medium class is calculated as the sum of the median of the Low class with half the difference with respect to the Medium class. The threshold between the Medium and High class is calculated in the same way. The resulting classes are compared against a groundtruth to generate the F<sub>1</sub>-Score metric seen in Table 4.

In view of the results shown in Table 4, it can be seen that the proposed SBPB method outperforms the other methods. This method has an average F<sub>1</sub>-Score about 5 % better than the second best. Medium and High classes of the SBPB method have a much higher F<sub>1</sub>-Score than the rest, however, the Prakasa et al. method outperforms the SBPB method in the Low opacity class. DOM (Transmission model) and Transmittance methods also produce a high Low opacity F<sub>1</sub>-Score. The Prakasa et al., Yuen et al., and DOCS method have difficulty distinguishing between Medium and High opacity.

Table 5 shows the complete results of the second dataset. Here it can be seen that the results obtained are similar to those of the first dataset, maintaining the same conclusions. In this particular dataset, the Ringelmann method also struggles with the Medium and High opacity levels. This second analysis helps to validate the methodology and the robustness of the method.

### 3.1. Limitations

The SBPB opacity estimation method has several limitations that should be considered when using it. For example, the method is unable

to work at night, as it relies on the visibility of the emissions and the reference background. Other methods, such as DOM (Transmission model), can work at night, but they require the use of two cameras and two lights pointing towards the emission (Du et al., 2009). Additionally, the SBPB method is unable to work if the color of the building is blue or too bright, as this does not meet the requirement that the sky must have higher intensity in the blue band than the building. For this same reason, the emission intensity in the blue band must be darker than the sky and lighter than the building.

While these limitations may seem strict, most methods in the literature are based on the standard EPA Method 9, which has even stricter requirements. EPA Method 9 requires the observer to have a line of sight of about 18° when looking up to the emission, and the sun must be behind the observer, oriented in a 140° sector to the observer's back. This means that observations using EPA Method 9 can only be made at a particular time of the day, when the sun is in the correct position.

In contrast, the SBPB opacity estimation method allows for more flexibility in terms of the time of day and the conditions under which measurements can be taken. However, the method still requires a building and the sky as a reference in order to accurately compare the degree of opacity in the emissions. The accuracy of the method also depends on the accuracy of the semantic segmentation model used to identify and isolate the emissions. However, in the tested images, this accuracy was high, reaching about 90 % F1-Score. The method could be further improved with the use of better cameras, better lighting and weather conditions, and more spectral bands, such as infrared, to enhance the visibility and accuracy of the measurement. Overall, while the limitations of the SBPB opacity estimation method should be considered and more testing in different settings may be needed to fully validate the method, they are similar to those of other methods in the literature.

#### 4. Conclusion

In this work, a method to estimate opacity of fugitive emissions capable of operating automatically without operator intervention is proposed. Existing methods for emission opacity estimation in images are not designed for fugitive emissions, and therefore have major limitations: these algorithms are always focused on black or white emissions, and they require an operator to select the regions to be used for the calculations.

While it is true that a large dataset is needed to train the network, the proposed method, SBPB, is the most stable of the methods presented due to its closeness to human operator assessments and its performance in different weather and lighting conditions such as a sunny or cloudy day at different times. This method can estimate opacity individually for each pixel, providing more information. One of its great advantages is that, thanks to the reference system, it does not require a properly calibrated camera without autotuning of exposure. A common camera such as a surveillance camera can be used. The use of semantic segmentation makes the SBPB method fully automatic i.e., it does not require the intervention of an operator. Furthermore, because this approach classifies all pixels in the image, more pixels can be employed as needed rather than a small section chosen by an operator. However, the greatest advantage of the SBPB method is its usefulness in characterizing fugitive emissions, since this task would be very difficult and time-consuming for a human operator. SBPB is capable of estimating the opacity of emissions of any color with low intensity in the B-band. The evaluated datasets contain brown and black emissions. Most methods are designed for pure black or pure white plumes. However, SBPB does not rely on a physical model that can be affected by a myriad of factors such as reflectivity indices and other variables, which may be different for each camera. Instead, it is based on a simpler idea: the assumption that the sky is blue, and that the building and emission are not blue. This simplicity makes it possible to extrapolate the method to various situations.

SBPB has an F1-Score 4–7 % better than the Transmittance method, the second best performing method for Medium opacity. Furthermore, it also performs 6–10 % better than the Transmittance method, the second best performing method for High opacity. For Low opacity emissions, the SBPB method is outperformed by Prakasa et al. by about 3 %. However, the Prakasa et al. method tends to underestimate opacity which can be seen in its Medium and High opacity scores. The DOM (Transmission model) and Transmittance methods also produce similar Low opacity scores. The SBPB method is the most robust of all the discussed methods because it outperforms every other method for the Medium and High classes. This is confirmed when looking at the average of all three classes which surpass the second best method, Transmittance, by over 5–7 % for all classes.

Metrics and visual results indicate that SBPB can help monitor stack and fugitive emissions from industrial plants in real time even when using uncalibrated, self-adjusting exposure time monitoring cameras such as the ones used in this study. In this way, emissions can be monitored in realtime as they are detected, as well as recording the severity of emissions during the day. This would make it possible to penalize those industrial plants that do not respect current regulations, thus endangering the environment.

#### Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Acknowledgments

This work has been partially funded by projects RTI2018-094849-B-I00 and PID2021-124383OB-I00 of the Spanish National Plan for Research, Development and Innovation.

#### Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at [doi:10.1016/j.psep.2022.12.023](https://doi.org/10.1016/j.psep.2022.12.023).

#### References

- Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2017a. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 834–848.
- L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, arXiv preprint arXiv:1706.05587 (2017b).
- L.-C. Chen, G. Papandreou, I. Kokkinos, M.R. Murphy, A.L. Yuille, Semantic image segmentation with deep convolutional nets and fully connected crfs, arXiv preprint arXiv:1412.7062 (2014).
- Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. *ECCV* 833–851 (pp.).
- J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.
- Dolan, S.D., 2017. After considerable struggle, epa method 082 modernized visible emissions monitoring. *Nat. Gas. Electr.* 34, 21–25.
- Du, K., Rood, M.J., Kim, B.J., Kemme, M.R., Franek, B., Mattison, K., 2007. Quantification of plume opacity by digital photography. *Environ. Sci. Technol.* 41, 928–935.
- Du, K., Rood, M.J., Kim, B.J., Kemme, M.R., Franek, B., Mattison, K., 2009. Evaluation of digital optical method to determine plume opacity during nighttime. *Environ. Sci. Technol.* 43, 783–789.
- B.J. Kim, M.J. Rood, K. Du, Digital optical method (dom™) and system for determining opacity, 2009.US Patent 7,495,767.
- Laconde, T., 2018. Fugitive emissions: a blind spot in the fight against climate change. *fugitives emissions-sector profile*. INIS 51.
- J.S. Lighty, K.E. Kelly, R.T. Whitaker, D.M. Weinstein, J. Desha, Enhancement of Digital Methods for Determination of Opacity, Technical Report, UTAH UNIV SALT LAKE CITY DEPT OF CHEMICAL ENGINEERING, 2007.
- McFarland, M.J., Terry, S.H., Calidonna, M.J., Stone, D.A., Kerch, P.E., Rasmussen, S.L., 2004. Measuring visual opacity using digital imaging technology. *J. Air Waste Manag. Assoc.* 54, 296–306.

- McFarland, M.J., Olivas, A.C., Atkins, S.G., Kennedy, R.L., Patel, K., 2007. Fugitive emissions opacity determination using the digital opacity compliance system (docs). *J. Air Waste Manag. Assoc.* 57, 1317–1325.
- McFarland, M.J., Palmer, G.R., Olivas, A.C., 2010. Life cycle cost evaluation of the digital opacity compliance system. *J. Environ. Manag.* 91, 927–931.
- McNamara, A., Chalmers, A., Troschianko, T., Gilchrist, I., 2000. Comparing real & synthetic scenes using human judgements of lightness. In: *Eurographics Workshop on Rendering Techniques*. Springer, pp. 207–218 (pp).
- Pedrayes, O.D., Lema, D.G., Usamentiaga, R., García, D.F., 2022. Detection and localization of fugitive emissions in industrial plants using surveillance cameras. *Comput. Ind.* 142, 103731.
- W.P. Pfaff, J. Stretch, Optical digital environment compliance system, 2003. US Patent 6,597,799.
- E. Prakasa, et al., Development of imaging based method for plume opacity measurement, in: 2017 5th International Conference on Instrumentation, Control, and Automation (ICA), IEEE, 2017, pp. 212–216.
- K. Randolph, K. Foster, Visible emissions field manual epa methods 9 and 22, U.S. Environmental Protection Agency (1993).
- S. Rasmussen, P. Grieco, DOCS II as “ACE” in Lieu of an ASTM Standard for Digital Cameras, Technical Report, OGDEN AIR LOGISTICS CENTER HILL AFB UT AIR BASE WING (75TH), 2009.
- Ringelmann, M., Kudlich, R., 1967. Ringelmann Smoke Chart. vol. 8333. US Bureau of Mines.
- B. Series, Colour gamut conversion from recommendation itu-r bt. 2020 to recommendation itu-r bt. 709, International Telecommunication Union (2017).
- Solomon, S., Manning, M., Marquis, M., Qin, D., et al., 2007. Climate change 2007-the physical science basis: Working group I contribution to the fourth assessment report of the IPCC. volume 4. Cambridge university press.
- Vinutha, H., Poornima, B., Sagar, B., 2018. Detection of outliers using interquartile range technique from intrusion dataset. In: *Information and decision sciences*. Springer, pp. 511–518 (pp).
- Yuen, W., Gu, Y., Mao, Y., Koloutsou-Vakakis, S., Rood, M.J., Son, H.-K., Mattison, K., Franek, B., Du, K., et al., 2017. Performance and uncertainty in measuring atmospheric plume opacity using compact and smartphone digital still cameras. *Aerosol Air Qual. Res.* 17, 1281–1293.
- Yuen, W., Gu, Y., Mao, Y., Kozak, P.M., Koloutsou-Vakakis, S., Son, H.-K., Mattison, K., Franek, B., Rood, M.J., 2018. Daytime atmospheric plume opacity measurement using a camcorder. *Environ. Technol. Innov.* 12, 43–54.



### 5.1.5. Remote sensing for detecting freshly manured fields

- Pedrayes, O. D., Usamentiaga, R. (2023), Trichakis Y, & Bouraoui F. *Remote sensing for detecting freshly manured fields*. *Ecological Informatics*, 75, 102006.
- DOI: [10.1016/j.ecoinf.2023.102006](https://doi.org/10.1016/j.ecoinf.2023.102006)
- El índice de impacto de la revista *Ecological Informatics* en 2021 fue 4.498 (Q2, 74.86%) y el índice de impacto a 5 años, 4.180.



Contents lists available at ScienceDirect

## Ecological Informatics

journal homepage: [www.elsevier.com/locate/ecolinf](http://www.elsevier.com/locate/ecolinf)

## Remote sensing for detecting freshly manured fields

Oscar D. Pedrayes<sup>a</sup>, Rubén Usamentiaga<sup>a,\*</sup>, Yanni Trichakis<sup>b</sup>, Faycal Bouraoui<sup>b</sup><sup>a</sup> Department of Computer Science and Engineering, University of Oviedo, Campus de Viesques, Gijón 33204, Asturias, Spain<sup>b</sup> European Commission – Joint Research Centre, Via Enrico Fermi 2749, Ispra, 21027 Varese, Italy

## ARTICLE INFO

## Keywords:

Computer vision  
Agriculture  
Crops  
Machine learning  
Leaching  
Manure  
Spreading

## ABSTRACT

The application of manure on fields during the rainy season can pollute the environment by leaching nitrates into nearby bodies of water. To reduce and prevent this problem, closed periods are declared to stop the use of manure as a fertilizer during these seasons. However, compliance is difficult to verify due to the impracticality of monitoring all the fields in a region or country through on-site or satellite surveys. To address this problem, a method to automatically monitor freshly manured fields using machine learning and satellite imagery is proposed. This paper evaluates the Sentinel-2 (S2) satellite bands as well as 51 of the most common vegetation indices (VI) found in the literature for precision agriculture through experiments based on several feature selection methods. To train the models, a new dataset of freshly manured plots verified by on-site investigations is generated and made publicly available for future research. The proposed method is able to fully detect all freshly manured plots in the test dataset obtaining an F<sub>1</sub>-Score close to 90% of the tested area.

## 1. Introduction

It is a well known fact that the fertilizers used in agriculture can pollute the immediate environment. There are several studies that analyze the environmental impact of the use of fertilizers such as manure (Kleinman et al., 2020; Liu et al., 2018; Tzilivakis et al., 2021). The most widely discussed issue is the impact of nitrogen and phosphorus leaching on nearby bodies of water including groundwaters (Kim et al., 2016). This problem is accentuated by rain, which carries the chemicals in the manure to groundwaters and causes serious pollution, harming the environment and the humans that use that water. Recently there have been reports of large numbers of fish dying due to lack of oxygen in the water because of contamination caused by leaching of fertilizers from nearby crops (Fahmy, 2022; Gillespie, 2022; Kirchman, 2022). Over the past century, climate change has caused the annual precipitation in the United States to rise by 10% (Karl and Knight, 1998) and according to projections from the Intergovernmental Panel on Climate Change (IPCC), this trend is expected to continue in the Midwest region throughout the 21st century (Daloglu et al., 2012; Culbertson et al., 2016). The literature covers other problems such as the supplements given to cattle to help them grow. When these supplements are overdosed, the animals excrete the excess, contaminating the manure with metals. In time, this pollutes the soil and its environment (Brugger

and Windisch, 2015).

According to the Food and Agriculture of the United Nations (2020), the total manure deposited on agricultural land worldwide is 116 million tons. 50% of this quantity represents excess nitrogen, which amounts to 58 million tons. 23 million tonnes of excess nitrogen volatilized in the atmosphere, primarily as ammonia gas, which had a significant impact on the air quality of the region. Leaching caused roughly 35 million tonnes of excess nitrogen to be lost in soil water and runoff, which subsequently contaminated waterways and eventually coastal seas.

Nitrogen leaching is hazardous to the environment and human population. For this reason, the European Commission has created two directives to reduce water pollution from nitrates used for agricultural purposes: Council of the European Union (1991) and Council of the European Union (2000). One proposed solution is closed periods, during which farmers in nitrate vulnerable zones are not authorized to spread organic fertilizers with a high nitrogen content, or manufactured nitrogen fertilizers. Each Member State has its own closed periods depending on its terrain, seasonal characteristics, and fertilizers. This solution was proven effective in Tzilivakis et al. (2021) as a preventive measure to minimize environmental pollution. However, enforcement is problematic since monitoring every plot of land in an extensive area is infeasible. On-site investigations are too costly and time consuming, and

\* Corresponding author.

E-mail addresses: [251056@uniovi.es](mailto:251056@uniovi.es) (O.D. Pedrayes), [rusamentiaga@uniovi.es](mailto:rusamentiaga@uniovi.es) (R. Usamentiaga), [Ioannis.TRICHAKIS@ec.europa.eu](mailto:Ioannis.TRICHAKIS@ec.europa.eu) (Y. Trichakis), [Faycal.BOURAOUI@ec.europa.eu](mailto:Faycal.BOURAOUI@ec.europa.eu) (F. Bouraoui).

<https://doi.org/10.1016/j.ecoinf.2023.102006>

Received 26 October 2022; Received in revised form 18 January 2023; Accepted 18 January 2023

Available online 2 February 2023

1574-9541/© 2023 Published by Elsevier B.V.

the manpower needed would be enormous. A monitoring system based on satellite imagery is the most feasible solution.

There is a vast amount of literature on the classification of different types of crops (Orynbaikyzy et al., 2019; Pedrayes et al., 2021) as well as their health or condition (Mutanga et al., 2017; Shanmugapriya et al., 2019) using satellite imagery. However, to the best of the authors' knowledge, there is no literature that addresses detection of manure spreading in different types of fields using remote sensing satellite technologies. The closest study to the problem is that of Yang et al. (2002), in which decision trees are used to distinguish between organic manure and chemical fertilizers in two different types of crops. In this case, the images were acquired from airplanes and have a spatial resolution of 2 m<sup>2</sup>. Aircraft imagery has a greater resolution than satellite imagery but its revisit time is too long for a monitoring system. Dodin et al. (2021) evaluates the relationship of the presence of manure in bare soil with different vegetation indices (VI) using the satellite Sentinel-2 (S2), but does not study any classification methods.

Wang (2009) is centered around manure visualization but uses much older satellites, such as Landsat 5 and 7, which have fewer bands than more modern satellites such as Landsat8 or Sentinel-2, which are the most frequently used (Dodin et al., 2021; Ma et al., 2010; Romanko, 2017; Zhu et al., 2021a). On rare occasions, commercial satellites are used.

To classify nitrogen, phosphorus, and fertilizers in general, the most common methods are: decision trees (Yang et al., 2002), random forest (Zhu et al., 2021a), or customized artificial neural networks (Jaihuni et al., 2021; Fu et al., 2021). All of the papers create datasets although no public dataset was found in the literature. This could be because the topic is very specific and also because machine learning methods require large datasets. Generating a dataset of adequate size can be time consuming, costly, and tedious. Moreover, in situ samples are needed to confirm the use of fertilizer.

In the field of precision agriculture, the use of more complex deep learning architectures such as Convolutional Neural Networks (CNN), Recurrent Neural Network (RNN), or Long Short Term Memory Network (LSTM) have been gaining popularity (Ferchichi et al., 2022; Ye et al., 2022; Alzu'bi and Alsmadi, 2022; Bragagnolo et al., 2021). However, these architectures have the disadvantage of requiring a larger amount of data than traditional machine learning techniques, which can be a limitation in situations where dataset creation is a complex and costly task.

Most of the papers studied (Dodin et al., 2021; Fu et al., 2021; Romanko, 2017; Yang et al., 2002) agree that short-wave infrared (SWIR) wavelengths are the most suitable for the detection of manure/nitrogen. This includes those vegetation indices that are composed of SWIR bands. The SWIR wavelength is also one of the most important wavelengths for land cover/land use (Prasad et al., 2022). However, there is little research of vegetation indices specifically for the detection of manure application. Dodin et al. (2021) and Wang (2009) followed this approach by creating new vegetation indices. Most studies are focused on a single crop type i.e., wheat, corn, etc, which suggests that each crop type has different features and that a method for all crop types may not be feasible. Some of the papers (Ma et al., 2010; Zhu et al., 2021a) also mention the problem of the low spatial resolution of the satellites. For this reason, works such as Zhu et al. (2021b) use their spectral bands as supporting data for other systems based on unmanned aerial vehicles (UAV) for in situ solutions. The importance of spatial resolution is further reinforced by studies like (Xia and Zhang, 2022), which found that high-definition imagery from Gaofen-2 can lead to slightly better results for soil pH mapping compared to multispectral images from Sentinel-2 and Landsat8. However, it is worth noting that this study did not make use of vegetation indices, which can be an important factor in detecting freshly manured fields. Therefore, it is important to consider the trade-offs between spatial resolution, spectral resolution and other factors.

Dodin et al. (2021) mentions the possibility of using time series to

improve the results. The use of images taken on different dates could solve the spatial resolution issue. However, obtaining images close in time can be complicated when the areas of interest are covered by clouds, which is especially frequent during winter.

In this work, an automated monitoring system for detecting freshly manured fields using machine learning methods and satellite imagery is proposed. The goal is to reduce the time and cost of detecting freshly manured fields and enable the enforcement of closed period laws. This work evaluates the most common vegetation indices related to precision agriculture found in the literature through a series of experiments using various feature selection methods. The aim of these experiments is to provide a robust solution and gain a deeper understanding of the importance of different wavelengths and vegetation indices. To further improve the accuracy of the models, an image prior to manure application is also used in addition to the image after. This provides the model with more information about the changes in the soil. A new dataset, consisting of freshly manured crop fields in northern Spain using Sentinel-2 imagery in a range of 30 days before and after manure application, is generated and made publicly available for further research. On-site investigations were conducted to find suitable fields and confirm recent manure application. The proposed approach has the potential to improve compliance monitoring and reduce the environmental impact of leaching from fertilizers. The results of this study contribute to the development of more effective methods for monitoring fertilizer use and protecting the environment.

## 2. Proposed approach

### 2.1. Methodology

This section presents a workflow for the study, as depicted in Fig. 1. The workflow encompasses several crucial stages, including plot investigation, image acquisition and processing, feature selection, model selection, model training, evaluation, and results and discussion.

The plot investigation stage involves researching and confirming the potential existence of manured plots through on-site investigations. Once the plots have been identified, Sentinel-2 imagery is obtained and utilized to generate ground truth masks. In the feature selection stage,

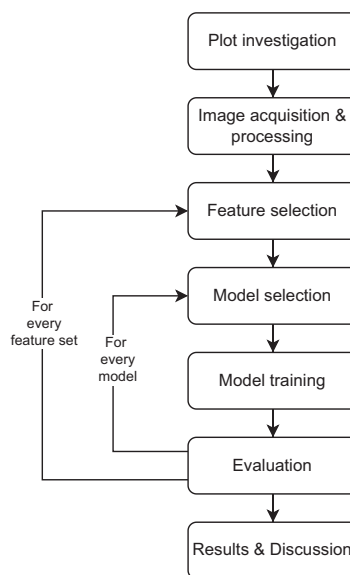


Fig. 1. Workflow.

the images are processed to extract features relevant to the study, and multiple feature sets are selected using various methods. Subsequently, one model is chosen and trained on 70% of the dataset. The remaining 30% of the dataset is used for evaluating the results, which are compared among different models and feature sets, providing an understanding of the significance of feature selection and model selection processes. Lastly, the results are presented and discussed in terms of their implications for identifying manured plots using remote sensing techniques.

The following sections provide an in-depth examination of the training and detection pipelines.

### 2.1.1. Training pipeline

Fig. 2 shows the flow of processes needed to train the model. First, the images required to train the model were obtained from Sentinel-2 satellite data. To determine the regions of interest to download the images, on-site inspections were carried out. From these regions of interest, pixels for the ground truth mask necessary for model training were selected by a manual pixel labeling process. In this process, regions of the images in which the plots had not been fertilized were selected as counterexamples, trying to obtain the greatest possible variability by selecting tall grass, mowed grass, plowed land, etc. The regions that were of no interest for this study were removed from these images, i.e., regions that cannot be fertilized, such as buildings, roads or forests. To determine the regions of no interest, land use information was acquired from an updated regional database. These pixels were removed automatically in function of their georeference.

To prepare the necessary data for training, features from each pixel were calculated and selected. In addition to the different bands, these features can be vegetation indices calculated from them. In this study, multiple approaches based on feature selection were evaluated. To increase the amount of information, the last image before manure application was also considered. This doubles the number of features and adds important information about possible changes over time. The models were trained using these features. The dataset was separated by plots instead of pixels to avoid training and test sets having pixels close to each other, which minimizes the possibility of overfitting. In order to train the models, 70% of the available plots were utilized, which included 21 plots or 18079 pixels (including counter examples). To evaluate the performance of the models, the remaining 30% of the plots were used, comprising 9 plots or 4615 pixels (including counter examples). To select the best approach for the model training, common machine learning methods were evaluated: Decision Tree, Discriminant Analysis, Logistic Regression Classifiers, Naïve Bayes Classifiers, Support Vector Machines, Nearest Neighbor Classifiers, Kernels Approximation Classifiers, Ensemble Classifiers, and Neural Network Classifiers.

### 2.1.2. Detection pipeline

Fig. 3 shows the process flow required to execute the manure detection model. First, as in the training part, it was necessary to obtain

the source images. Both the current image and the previous image are processed for each detection. The model is intended for continuous monitoring of an area, thus it is not necessary to know the date of fertilization. Regions that were not of interest, i.e., regions with land uses not related to crops and fields, were eliminated. This information was obtained from an updated regional database of land use. Using the remaining pixels, features were calculated and selected. The model was fed with the resulting feature image, obtaining a binary mask that indicates whether each corresponding pixel belongs to a plot where manure had been spread. To visualize the detected manure masks, morphological techniques were used for noise reduction/elimination. Because the detection is performed pixel by pixel, it is common for these masks to be noisy. This step is only performed on the final visualization and does not affect the results of the metrics. To eliminate mask noise, two techniques are used: an erosion using a 2x2 square structure to erase all floating pixels, which causes the regions to become smaller; and a dilation using a 3x3 cross structure to restore the size of the regions back to their original and fill gaps. The final masks are improved by eliminating loose pixels and filling in the regions.

## 2.2. Dataset

No public data was found in the literature that could be used for this study. For this reason, a dataset was generated manually. The satellite chosen for the images is Sentinel-2, due to its spectral range, spatial resolution, and above all its short revisit time of 5 days.

To take advantage of Sentinel-2 temporal information, it was decided to use, in addition to the image immediately after the application of the manure, the image immediately before. In this way, the change between the two images can add significant information about the soil and improve the results. This doubles the number of features per pixel.

## 2.3. Ground truth generation

To create a proper ground truth mask it is necessary to locate the target plot, create its mask and remove unnecessary pixels such as roads or buildings, and finally obtain counterexamples to correctly train the models.

### 2.3.1. Plot localization

To locate the plot, an on-site investigation was performed to confirm that the plot had been recently manured and to observe the real dimensions of the fertilization in the plot (see Fig. 4a). Then, using Google Earth Engine, the plot was annotated according to the observed dimensions, as shown in Fig. 4b. The annotation was then used to generate the raster ground truth mask using the Sentinel-2 georeferenced image, as shown in Fig. 5.

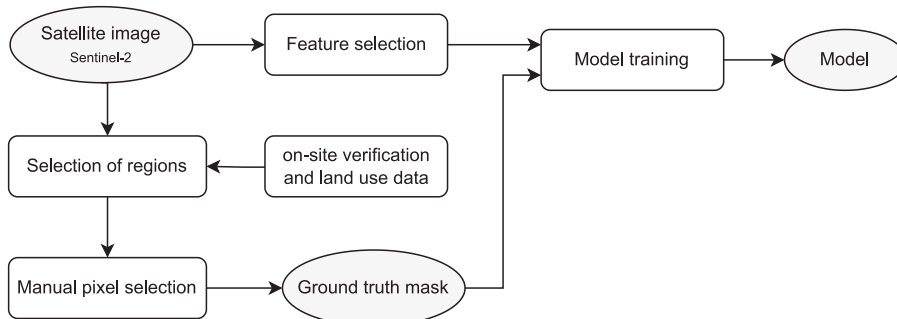


Fig. 2. Training pipeline.

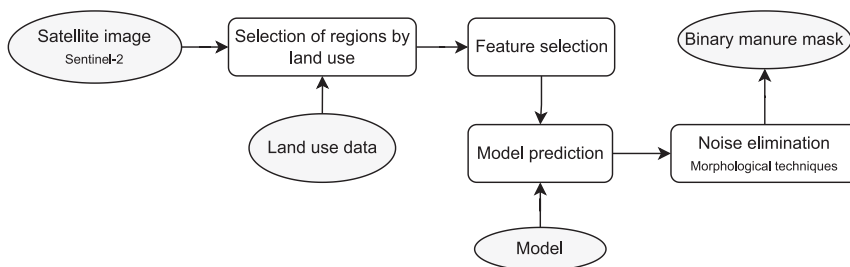


Fig. 3. Detection pipeline.



(a) On-site investigation

(b) Plot annotated in Google Earth Engine (650 x 500 m)

Fig. 4. P-VG1 annotating example.



(a) Original Sentinel-2 image (1980 x 1980 m)

(b) Ground truth mask (1980 x 1980 m)

Fig. 5. P-VG1 ground truth mask example.

### 2.3.2. Regions of no interest

To reduce the complexity of the problem it is helpful to remove pixels of no interest such as roads, buildings, or bodies of water. Since these areas are not fertilized, they need not be taken into consideration. To accomplish this in a reliable and automated way, a regional land use database is used. Sentinel-2 offers a layer called “Scene classification

layer”, however, it does not have enough classes for this study, as it cannot distinguish classes such as Forest from Grassland.

The database of the Geographic Information System of Agricultural Land (SIGPAC) (Spanish Ministry of Agriculture Food and Environment, 2007) is selected to eliminate pixels of no interest according to land use. This database is free to access and is frequently updated. Land use data

from SIGPAC is only available for the Spanish region but this does not limit the usefulness of this approach as other countries in the European Union have similar databases.

SIGPAC has 30 classes in total (see Table 1). From these classes, 18 are considered of interest and 12 of no interest. This selection can be easily seen in Table 2, where classes of interest are in black text and classes of no interest in red text.

### 2.3.3. Counterexamples

In addition to classifying the data relevant to the target class, to train a classification model it is necessary to have a counterexample class in which to classify those pixels that do not belong to the target class. This means that for a classification, at least two classes must exist, i.e., binary classification. This new class is defined as “Others”. The target class is called “Manure application” class. The pixels for the counterexamples are manually chosen from the full images. Regions of the image are selected where the fields are known not to have been fertilized including reflectivity changes caused by other agricultural processes, e.g. plowed land, which is visually similar to recently manured land. In addition, some regions belonging to forests, roads, and buildings are selected to add more context to the counterexamples. Even if these regions are previously eliminated by land use masks from regional databases, it is useful to add a few examples of these to avoid border regions of the masks or mislabelled regions from being incorrectly classified as “Manure application” class.

## 2.4. Features

In this section the features used to generate the dataset are detailed. For this study, the 13 raw bands of the Sentinel-2 satellite are used as features along with 51 of the most common vegetation indices found in the literature for precision agriculture. In total, each pixel has 64 features. As an example, a zoom on the region of plot P-VG1 shown in Fig. 6 is used to visualize all the features from the first image after manure application in Fig. 7. Features corresponding to the Sentinel-2 bands are detailed in Section 2.4.1, while features corresponding to the vegetation indices are detailed in Section 2.4.2.

### 2.4.1. Bands

The 13 Sentinel-2 bands are used as the first 13 features for a given pixel. Since not all bands have the same resolution, bands with lower resolution are interpolated using the Nearest Neighbor algorithm. The rest of the features are calculated as combinations of these 13 bands. The Sentinel-2 mission has two satellites, called Sentinel-2A and Sentinel-2B, each with an orbit of around 10 days. Because both satellites are at the

**Table 1**  
SIGPAC classes.

Class	Description	Class	Description
CF	Citrus-Fruit	IM	Unproductive
CS	Citrus-Fruit Peel	IV	Greenhouses and crops under plastic
CV	Citrus-Vineyard	OV	Olive grove
FF	Association Fruit Trees-Fruit Trees Of Peel	OF	Olive grove - Fruit trees
OC	Olive-Citrus	PS	Pasture land
CI	Citrus	PR	Shrub Grass
AG	Watercourses and Water Surfaces	PA	Grassland with Trees
ED	Buildings	TA	Arable Land
EP	Landscape element	CA	Roads
FO	Forest	VI	Vineyard
FY	Fruit Trees	VF	Vineyard - Fruit Tree
FS	Dried Fruits	VO	Vineyard - Olive grove
FL	Nuts and Olives	ZV	Censored Area
FV	Nuts and Vineyard	ZC	Concentrated Zone not included in Orthophoto
TH	Orchard	ZU	Urban Zone

maximum possible distance from each other, the revisit time is halved. Thus, new images of the same region are obtained every 5 days. Table 3 shows the wavelength, bandwidth, and spatial resolution of every band for the Sentinel-2A and Sentinel-2B satellites.

### 2.4.2. Vegetation indices

A vegetation index generates a feature for each pixel. To generate as many relevant features as possible, the most common vegetation indices in the literature for precision agriculture are studied. This section describes all the vegetation indices used in this study. Table 4 shows the order, abbreviation, full name, and equation used to calculate their value.

### 2.4.3. Feature selection methods

Sometimes having too many features can be counterproductive, as it decreases the accuracy of the classification model. For this reason, and because multiple features could be redundant, different approaches for feature selection are evaluated: the Boruta method (Kursa and Rudnicki, 2010), the Principal Component Analysis (PCA) method (Wold et al., 1987) using 95% variability; the Recursive Feature Elimination (RFE) method (Guyon et al., 2002); and the Ward clustering algorithm (W) using Eq. (1).

$$d(A, B) = \frac{\sqrt{2|A||B|}}{|A| + |B|} |cA - cB| \quad (1)$$

In addition, this paper evaluates the raw Sentinel-2 bands, the RGB bands, the infrared (IR) bands, and the vegetation indices separately.

## 2.5. Metrics

In order to understand the performance of the models and to be able to compare the results of the different methods, different metrics are calculated (Fernandez-Moral et al., 2018). All metrics are based on the concepts of true positive (TP), false positive (FP), true negative (TN) and false negative (FN).

- TP are the correctly classified pixels.
- FP are the pixels that are classified as the target class which do not belong to that class.
- TN are the pixels that are not classified as the target class and do not belong to that class.
- FN are the incorrectly classified pixels.

Accuracy is one of the most commonly used metrics to summarize the performance of a model in a single value. It is calculated as the total number of correctly classified pixels divided by the total number of pixels, as shown in Eq. (2). This metric can be misleading if the classes are too unbalanced, i.e., if there is too great a difference in the number of pixels in each class.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (2)$$

Precision is calculated as the correctly classified pixels divided by the total detected pixels, as shown in Eq. (3).

$$Precision = TP / (TP + FP) \quad (3)$$

Recall is calculated as the pixels classified correctly divided by the pixels that correspond to that target class in the ground truth, as shown in Eq. (4).

$$Recall = TP / (TP + FN) \quad (4)$$

If Precision is low and Recall is high, the detections will over-classify pixels from the target class. If Recall is low and Precision is high, only the pixels with high confidence will be classified as the target class. F<sub>1</sub>-Score is a metric used to evaluate both Precision and Recall in a single value. Because of this, the F<sub>1</sub>-Score is one of the most common metrics in image

**Table 2**  
SIGPAC class selection. Classes of interest in black and classes of no interest in red.

Group	Class
Farmland	TA, TH, IV
Permanent crops	CF, CS, CV, FF, OC, CI, FY, FS, FL, FV, OV, OF, VI, VF, VO
Pastures	PS, PR, PA
Forest	FO
Non-agricultural area	AG, ED, EP, IM, CA, ZU
Others	ZV, ZC



**Fig. 6.** Image P-VG1 (left) and zoom on the plot (right).

segmentation. It is calculated as a combination of Precision and Recall as shown in Eq. (5) and is equivalent to the Dice Coefficient with two classes.

$$F_1\text{-Score} = (2 \times \text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (5)$$

Mean Precision and Mean Recall are calculated as the average of the Precision and Recall of both classes. Mean F<sub>1</sub>-Score is calculated using the Mean Precision and Mean Recall.

### 3. Results and discussions

#### 3.1. Dataset generation

Dataset generation involved visiting 38 different plots of land, of which 8 were discarded because they were not suitable for the study, either because of their small size, the lack of a specific manure application date, or the images being too cloudy in that period. The complete dataset with the analysis of each of the plots visited, the code needed to obtain and process the images, and their ground truth generation, is publicly available to ensure reproducibility at: <https://doi.org/10.17632/fbvvf55kp.1>.

All the plots obtained for this dataset are pasture land as this is the predominant type of activity in this area of northern Spain. The general climate is oceanic, with abundant rainfall spread throughout the year and mild temperatures in both winter and summer. The images of the

plots have an area of about 1,700x1,700 m, although the plots inside the images are much smaller. All inspected plots fit in this area, as well as some adjacent plots without manure. Fields in this region generally belong to one of the following types:

- Tall grass. This type of land has a strong green color.
- Freshly mown. This type of land tends to have a yellowish green color
- Sowed/Plowed lands. This type of land is brown in color.
- Maize fields: This type of land is a dark green with dark shadow lines.
- Grazing lands. This pasture land has a brown color similar to other types. This color is caused by the cattle.
- Woods. This type of land is dark green. It usually has a high density of trees.
- Remnant habitat/Bushland. This type of land has different colors from a darker green to brown. It is composed of different wild bushes and grass.
- Manure application: Manure can be spread over freshly mown or sowed/plowed lands. The color of the land turns darker. It is common to find circular marks due to the method used to spread the manure.

A total of 30 plots of land are studied. Table 5 shows for every plot: its identifier; its manure application date; its area; and its geographical coordinates.

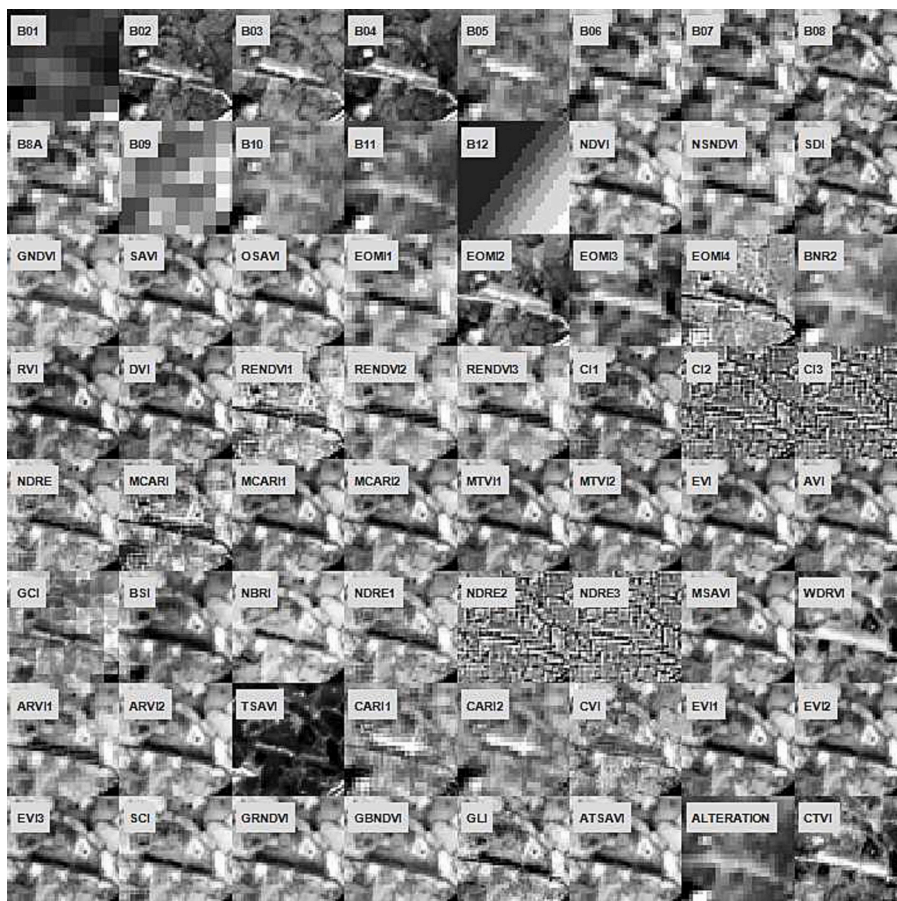


Fig. 7. Visualization of every feature of P-VG1.

Table 3  
Sentinel-2 bands.

#	Band	Central Wavelength ( $\mu\text{m}$ )		Bandwidth ( $\mu\text{m}$ )		Spatial resolution (m)
		S-2A	S-2B	S-2A	S-2B	
0	B01 Coastal aerosol	0.4427	0.4422	0.021	0.021	60
1	B02 Blue	0.4924	0.4921	0.066	0.066	10
2	B03 Green	0.5598	0.5590	0.036	0.036	10
3	B04 Red	0.6646	0.6649	0.031	0.031	10
4	B05 Vegetation red edge	0.7041	0.7038	0.015	0.016	20
5	B06 Vegetation red edge	0.7405	0.7391	0.015	0.015	20
6	B07 Vegetation red edge	0.7828	0.7797	0.020	0.020	20
7	B08 Near-infrared	0.8328	0.8329	0.106	0.106	10
8	B8A Narrow near-infrared	0.8647	0.8640	0.021	0.022	20
9	B09 Water vapour	0.9451	0.9432	0.020	0.021	60
10	B10 Short-wave infrared - Cirrus	1.3735	1.3769	0.031	0.030	60
11	B11 Short-wave infrared	1.6137	1.6104	0.091	0.094	20
12	B12 Short-wave infrared	2.2024	2.1857	0.175	0.185	20

The complete dataset consists of 225.94 hectares, 31.48 belong to the “Manure application” class, and 195.46 to the “Others”, which serves as the counterexample class. Each pixel is 0.01 hectares. The dataset is divided into plots rather than pixels to prevent training and test sets having pixels close to each other. This reduces the possibility of overfitting. For training, 70% of the available plots are used (21 of the 30 plots), with a total of 180.79 hectares (22.28 hectares of the target “Manure application” class and 158.51 hectares for the “Others” class). To evaluate the performance of the models, the remaining 30% of the plots (9 of the 30 plots), are used, with a total of 46.15 hectares (9.20 hectares of the target class and 36.95 hectares of the “Others class”). Table 6 summarizes this information. Fig. 8 shows the ground truth for some test plots in the dataset.

After obtaining the ground truth masks, those pixels belonging to regions of no interest are eliminated, as they are regions that cannot be fertilized. The SIGPAC database is used to perform this task. Fig. 9 shows two examples where regions of no interest are shown in blue.

The selection of counterexamples for the class “Others” is done manually. As not all possible areas of each image could be verified, only pixels that can be reliably confirmed as not being fertilized by on-site investigations are selected. In addition, to avoid a very large class imbalance, only part of the possible pixels in the image are selected. The dataset has a total of 195.46 hectares of counterexamples of the class “Other”. Of these, 158.51 hectares are for training and 36.95 hectares



**Table 4**  
Vegetation indices.

#	Abb.	Name	Equation
13	NDVI	Normalized Difference Vegetation Index (Sishodia et al., 2020; Shou et al., 2007)	$\frac{B08 - B04}{B08 + B04}$
14	NSNDVI	NIR-SWIR Normalized Difference Vegetation Index (Sentinel-Hub, 2022)	$\frac{B11 - B07}{B11 + B07}$
15	SDI	SWIR Difference Index (Ma et al., 2010)	$\frac{B08 - B12}{B08 + B03}$
16	GNDVI	Green Normalized Difference Vegetation Index (Sishodia et al., 2020; Zhu et al., 2021a)	$\frac{B08 - B03}{B08 + B03}$
17	SAVI	Soil Adjusted Vegetation Index (Sishodia et al., 2020; Ma et al., 2010; Sentinel-Hub, 2022)	$\frac{B08 - B04}{B08 + B04 + 0.428} * 1.428$
18	OSAVI	Optimized Soil Adjusted Vegetation Index (Sishodia et al., 2020; Fu et al., 2021; Bagheri et al., 2013; Zhu et al., 2021a)	$(1 + 0.16) * \frac{B08 - B04}{B08 + B04 + 0.16}$
19	EOMI1	Exogenous Organic Matter Index 1 (Dodin et al., 2021)	$\frac{B11 - B8A}{B11 + B8A}$
20	EOMI2	Exogenous Organic Matter Index 2 (Dodin et al., 2021)	$\frac{B12 - B04}{B12 + B04}$
21	EOMI3	Exogenous Organic Matter Index 3 (Dodin et al., 2021)	$\frac{(B11 - B8A) + (B12 + B04)}{B11 + B8A + B12 + B04}$
22	EOMI4	Exogenous Organic Matter Index 4 (Dodin et al., 2021)	$\frac{B11 - B04}{B11 + B04}$
23	BNR2	Normalized Burn Ratio 2 (Dodin et al., 2021)	$\frac{B11 - B12}{B11 + B12}$
24	RVI	Ratio Vegetation Index (Sishodia et al., 2020; Shou et al., 2007)	$\frac{B08}{B04}$
25	DVI	Difference Vegetation Index (Sishodia et al., 2020; Shou et al., 2007)	$\frac{B08 - B04}{B05 - B04}$
26	RENDVI1	Red Edge Normalized Difference Vegetation Index (Sishodia et al., 2020)	$\frac{B05 + B04}{B06 - B04}$
27	RENDVI2	Red Edge Normalized Difference Vegetation Index (Sishodia et al., 2020)	$\frac{B06 + B04}{B07 - B04}$
28	RENDVI3	Red Edge Normalized Difference Vegetation Index (Sishodia et al., 2020)	$\frac{B07 + B04}{B08 - 1}$
29	CI1	Chlorophyll Index (Sentinel-Hub, 2022)	$\frac{B08}{B05} - 1$
30	CI2	Chlorophyll Index (Sentinel-Hub, 2022)	$\frac{B08}{B06} - 1$
31	CI3	Chlorophyll Index (Sentinel-Hub, 2022)	$\frac{B08}{B07} - 1$
32	NDRE	Normalized Difference Red Edge (Sishodia et al., 2020; Zhu et al., 2021a)	$\frac{B07 - B05}{B08 + B05}$
33	MCARI	Modified Chlorophyll Absorption in Reflectance Index (Fu et al., 2021; Zhu et al., 2021a)	$\frac{((B05 - B04) - 0.2 * (B05 - B03)) * \frac{B05}{B04}}{1.2 * (2.5 * (B08 - B04) - 1.3 * (B08 - B03))}$
34	MCARI1	Modified Chlorophyll Absorption in Reflectance Index 1 (Sentinel-Hub, 2022)	$\frac{2.5 * (B08 - B04) - 1.3 * (B08 - B03)}{1.5 * \sqrt{(2 * B08 + 1)^2 - (6 * B08 - 5 * \sqrt{B04})} - 0.5}$
35	MCARI2	Modified Chlorophyll Absorption in Reflectance Index 2 (Bagheri et al., 2013)	$\frac{1.2 * (1.2 * (B08 - B03) - 2.5 * (B04 - B03))}{1.5 * \sqrt{(2 * B08 + 1)^2 - (6 * B08 - 5 * \sqrt{B04})} - 0.5}$
36	MTVI1	Modified Triangular Vegetation Index 1 (Bagheri et al., 2013)	$\frac{(B08 + 6 * B04 - 7.5 * B02) + 1}{(B08 * (1 - B04) * (B08 - B04))^{1/3}}$
37	MTVI2	Modified Triangular Vegetation Index 2 (Bagheri et al., 2013)	$\frac{B09 - 1}{B03}$
38	EVI	Enhanced Vegetation Index (Sentinel-Hub, 2022)	$\frac{B11 + B04 + \frac{B08 + B02}{B11 + B04} + B08 + B02}{B08 - B12}$
39	AVI	Advanced Vegetation Index (Sentinel-Hub, 2022)	$\frac{B08 + B12}{B08 + B05}$
40	GCI	Green Coverage Index (Sentinel-Hub, 2022)	$\frac{B08 + B05}{B08 - B06}$
41	BSI	Bare Soil Index (Sentinel-Hub, 2022)	$\frac{B08 + B06}{B08 - B07}$
42	NBRI	Normalized Burned Ratio Index (Sentinel-Hub, 2022)	$\frac{B08 + B07}{2.0 * B08 + 1 - \sqrt{((2.0 * B08 + 1.0)^2 - 8 * (B08 - B04))}}$
43	NDRE1	Normalized Difference Red Edge (Sishodia et al., 2020; Zhu et al., 2021a)	$\frac{0.1 * B08 - B04}{0.1 * B08 + B04}$
44	NDRE2	Normalized Difference Red Edge (Sishodia et al., 2020; Zhu et al., 2021a)	$\frac{B8A - B04 - 0.069 * (B04 - B02)}{B8A + B04 - 0.069 * (B04 - B02)}$
45	NDRE3	Normalized Difference Red Edge (Sishodia et al., 2020; Zhu et al., 2021a)	$\frac{B08 + B05}{B08 - B06}$
46	MSAVI	Modified Soil Adjusted Vegetation Index (Sishodia et al., 2020; Ma et al., 2010)	$\frac{B08 + B06}{B08 - B07}$
47	WDRVI	Wide Dynamic Range Vegetation Index (Sishodia et al., 2020)	$\frac{0.1 * B08 - B04}{0.1 * B08 + B04}$
48	ARVI1	Atmospherically Resistant Vegetation Index 1 (Sishodia et al., 2020; Sentinel-Hub, 2022)	$\frac{B8A + B04 - 0.069 * (B04 - B02)}{-0.18 + 1.17 * \frac{B8 - B4}{B8 + B4}}$
49	ARVI2	Atmospherically Resistant Vegetation Index 2 (Sishodia et al., 2020; Sentinel-Hub, 2022)	$\frac{(0.421 * (B08 - 0.421 * B04 - 0.824))}{(B04 + 0.421 * (B08 - 0.824) + 0.114 * (1 + 0.421)^2)}$
50	TSAVI	Transformed Soil Adjusted Vegetation Index (Sishodia et al., 2020)	$\frac{B05 * \left( \frac{(B05 - B03)}{150} * 670.0 + B04 + (B03 - \frac{(B05 - B03) * 550}{150}) \right)}{B04}$
51	CARI1	Chlorophyll Absorption Ratio Index 1 (Sentinel-Hub, 2022)	$\frac{\sqrt{(B05 - B03) / 150^2 + 1}}{(B05 - B03) / 150 * B04 + B04 + B03 - 0.496 * B03}$
52	CARI2	Chlorophyll Absorption Ratio Index 2 (Sentinel-Hub, 2022)	$\frac{\sqrt{(0.496^2 + 1)} * (B05 / B04)}{}$

(continued on next page)

Table 4 (continued)

#	Abb.	Name	Equation
53	CVI	Chlorophyll Vegetation Index (Sentinel-Hub, 2022)	$\frac{B08 * B04}{B03^2}$
54	EV11	Enhanced Vegetation Index 1 (Sentinel-Hub, 2022)	$\frac{2.5 * (B08 - B04)}{(B08 + 6 * B04 - 7.5 * B02) + 1}$
55	EV12	Enhanced Vegetation Index 2 (Sentinel-Hub, 2022)	$\frac{2.4 * \frac{B08 - B04}{B08 + B04 + 1}}{B08 - B04}$
56	EV13	Enhanced Vegetation Index 3 (Sentinel-Hub, 2022)	$\frac{2.5 * \frac{B08 + 2.4 * B04 + 1}{B08 - B04}}{B11 - B08}$
57	SCI	Soil Composition Index (Sentinel-Hub, 2022)	$\frac{B11 + B08}{B08 - (B03 + B04)}$
58	GRNDVI	Green-Red Normalized Difference Vegetation Index (Sentinel-Hub, 2022)	$\frac{B08 + (B03 + B04)}{B08 - (B03 + B02)}$
59	GBNDVI	Green-Blue Normalized Difference Vegetation Index (Sentinel-Hub, 2022)	$\frac{B08 + (B03 + B02)}{2 * B03 - B04 - B02}$
60	GLI	Green Leaf Index (Sentinel-Hub, 2022)	$\frac{2 * B03 - B04 - B02}{2 * B03 + B04 + B02}$
61	ATSAVI	Adjusted Transformed Soil-Adjusted Vegetation Index (Sentinel-Hub, 2022)	$\frac{1.22 * (B08 - 1.22 * B04 - 0.03)}{1.22 * B08 + B04 - 1.22 * 0.03 + 0.08 * (1 + 1.22^2)}$
62	ALTERATION	Alteration Index (Sentinel-Hub, 2022)	$\frac{B11}{B12}$
63	CTVI	Corrected Transformed Vegetation Index (Sentinel-Hub, 2022)	$\frac{((B04 - B03) / (B04 + B03)) + 0.5}{\frac{ B04 - B03 }{ B04 + B03 } + 0.5 * \sqrt{\frac{B04 - B03}{B04 + B03} + 0.5}}$

Table 5  
Plots in the dataset.

Plot	Date (YYYY/MM/dd)	Area (m <sup>2</sup> )	Coordinates (Long./Lat.)
P-BLD	2022/05/26	8900	-4.2018, 43.3973
P-BLLT1	2022/05/16	21200	-4.0840, 43.4309
P-BLLT2	2022/05/26	3300	-4.0840, 43.4310
P-Cardana	2022/02/24	6500	8.6580, 45.8592
P-CBRCS1	2022/05/26	6700	-4.2005, 43.3897
P-CBRCS2	2022/05/26	6400	-4.2048, 43.3875
P-CLGT	2022/05/16	17200	-4.1096, 43.3987
P-CLMBRS	2022/05/26	4300	-4.5447, 43.3804
P-CMNTR	2022/05/16	2600	-4.1470, 43.4001
P-DR	2022/03/21	2500	-4.1424, 43.3967
P-FNFR	2022/05/16	10100	-4.2657, 43.3880
P-LLT	2022/05/03	9600	-4.1515, 43.4001
P-LNDRS1	2022/05/16	3200	-4.2510, 43.3880
P-LNDRS2	2022/05/16	5400	-4.2503, 43.3880
P-LNDRS3	2022/05/16	8500	-4.2497, 43.3872
P-LNDRS4	2022/05/16	9100	-4.2467, 43.3877
P-MT	2022/05/04	19900	-4.1536, 43.3980
P-NMS	2022/02/10	5500	-4.1490, 43.4003
P-QNTLS2	2022/05/16	8500	-5.5840, 43.5458
P-SNTLLN	2022/03/17	14200	-4.1170, 43.3935
P-SNVCNT1	2022/05/16	6700	-4.4048, 43.3939
P-SNVCNT2	2022/05/16	29200	-4.4001, 43.3945
P-STBN	2022/05/04	11300	-4.1366, 43.3960
P-TGL2	2022/05/16	12300	-4.0701, 43.4276
P-TNNS1	2022/05/26	19500	-4.1871, 43.3996
P-TNNS2	2022/05/26	15800	-4.1918, 43.3987
P-VG1	2022/04/09	12200	-5.4866, 43.4699
P-VG2	2022/04/13	4900	-5.4801, 43.4693
P-VLDMR	2022/02/07	17500	-4.1561, 43.4056
P-VNS	2022/04/23	16600	-4.1504, 43.4042

Table 6  
Hectares per class.

Class	Train	Test
Others	158.51	36.95
Manure application	22.28	9.20
Total	180.79	46.15

are for testing.

### 3.2. Vegetation indices and manure application

This section analyzes the findings from the 51 generated vegetation indices taken at different times before and after the manure application. The purpose of this analysis is to determine whether or not the choice of vegetation indices is correct based on their correlation with manure application. Fig. 10 displays a heatmap with all 51 vegetation indices for six of the plots in order to depict the intensity of their values. The values of each row are scaled using the min-max normalization method. The first image after applying manure is labeled image 0 on the X axis. Images are categorized as negative or positive depending on whether manure application came before or after them. To aid in visualization, a vertical dotted red line is placed at the beginning of the first image after the manure application date. On the day of manure application, the intensity of the majority of the vegetation indices in Fig. 10 is almost zero, and as time goes on, the values rise. The remaining indices also seem to be inversely associated, with values near 0 before manure application and near 1 following manure application. A small number of the vegetation indices appear to be unrelated to the application of manure. As a result, it appears that the indices used offer useful information regarding the presence of manure in the plots.

### 3.3. Summary of experiments

To find the best model, a series of experiments are performed by varying the set of features. For each set of features all the different classification methods mentioned above are trained and evaluated. Table 7 describes all feature sets and associates them with an identifier. It is important to note that the Boruta feature selection does not discard any feature, which means that the experiment that uses all the features (BA-128) is the same and does not need to be repeated. The Recursive Feature Elimination has found that the optimal number of features is 90. For this reason, only the BA-90-RFE feature set is shown for this selection method.

Table 8 shows the best results for every feature set. All models are trained using the train set and all the metrics are calculated using the test set. The best feature set is BA-102-VI. BA-90-RFE shows very similar results, however, Mean Precision and Mean Recall are slightly more balanced in the BA-102-VI feature set.

This study proves that manure application in fields can be detected by satellite remote sensing with great success. The best model has a

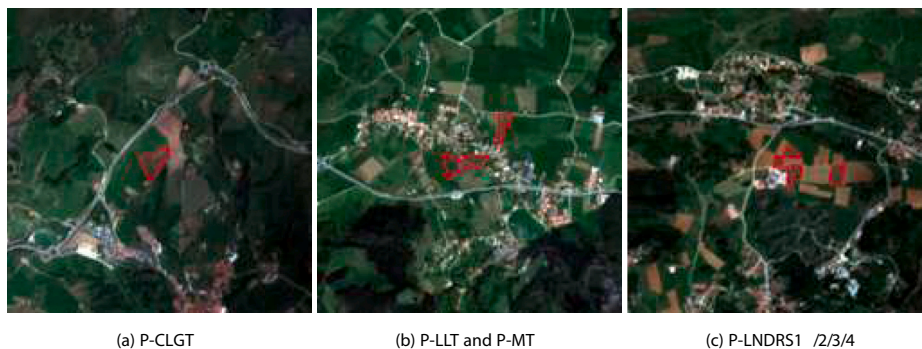


Fig. 8. Examples of ground truth.

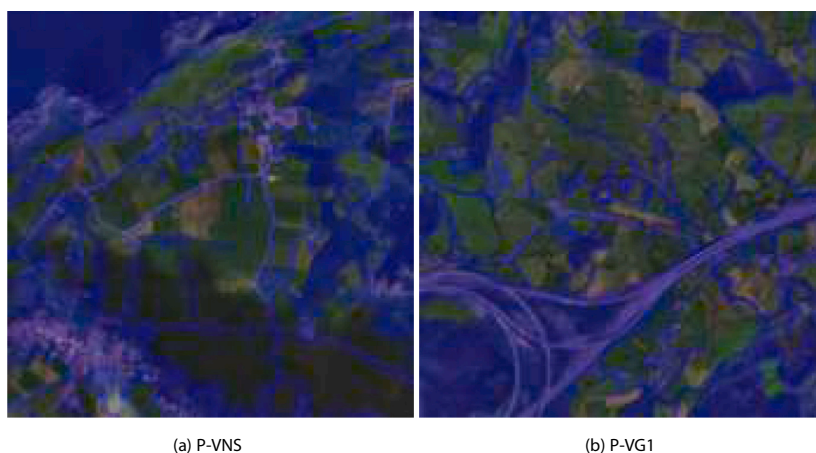


Fig. 9. SIGPAC mask examples overlaid in blue on the Sentinel-2 image.

Mean  $F_1$ -Score of 93.4%. However, these results were obtained using a dataset of 31.48 ha of newly manured fields. With a larger and more varied dataset, conclusions could change slightly.

It was found that combining the features from before and after manure application images provided better results. For example, the Mean Recall of BA-128 is about 8% better than that of A-64 while maintaining its Mean Precision. This improvement is consistent for every experiment of type “BA” against “A”.

Models with a reduced number of features have inferior results. The Ward clustering feature selection method (A-7-W and BA-14-W) shows results about 5–7% lower than when using all features.

The Recursive Feature Elimination method (BA-90-RFE) achieves better results than using all features (BA-128). Using more features than necessary can make the models harder to train and produce less accurate detections. This problem is known as the “curse of dimensionality”. Most features are highly correlated, making the use of all features unnecessary.

Acceptable results are achieved with BA-RGB, using only the RGB bands of both images (after and before manure application). This may be due to the spatial resolution offered by these bands of  $10\text{ m} \times 10\text{ m}$  per pixel. However, BA-16-IR still outperforms it by about 4%. Even when the IR bands (bands with a wavelength greater than B04 Red band) have a spatial resolution of  $20\text{ m} \times 20\text{ m}$  per pixel, better results are still achieved. This is further supported by the response of the IR bands when manure is applied, as depicted in Fig. 11. The figure illustrates that

bands B06, B07, B08, B8A, and B09 are highly correlated, while B10 and B11 are inversely correlated. This supports the findings of the study and shows that raw data in IR wavelengths is essential for detecting manure application. If the IR bands had a similar spatial resolution to the RGB bands, it is expected that results would be even more accurate. Similarly, the wavelength of the IR bands ranges from 700 nm to 2200 nm, so it is possible that if this range were to be extended, results would improve.

Discriminant Analysis is the classification method that offers the most accurate results, especially when a high number of features is used. Support Vector Machines usually obtains better results when the number of features is lower.

### 3.4. Best experiment

This section shows in detail the results of the best experiment: the BA-102-VI. Table 9 shows the results for each classification method. In this case, the most accurate method is Discriminant Analysis, which surpasses the second most accurate method, Ensemble Classifiers, by almost 2% Mean  $F_1$ -Score. Table 10 shows the details of the most accurate method. In this table, Precision, Recall, and  $F_1$ -Score are shown for the “Others” class and the “Manure application” class. The classes are slightly imbalanced because there are more samples for the “Others” class. For this reason, it is more reasonable to focus on the results of the “Manure application” class, specifically, the  $F_1$ -Score, which considers both Precision and Recall. The  $F_1$ -Score for the “Manure application”

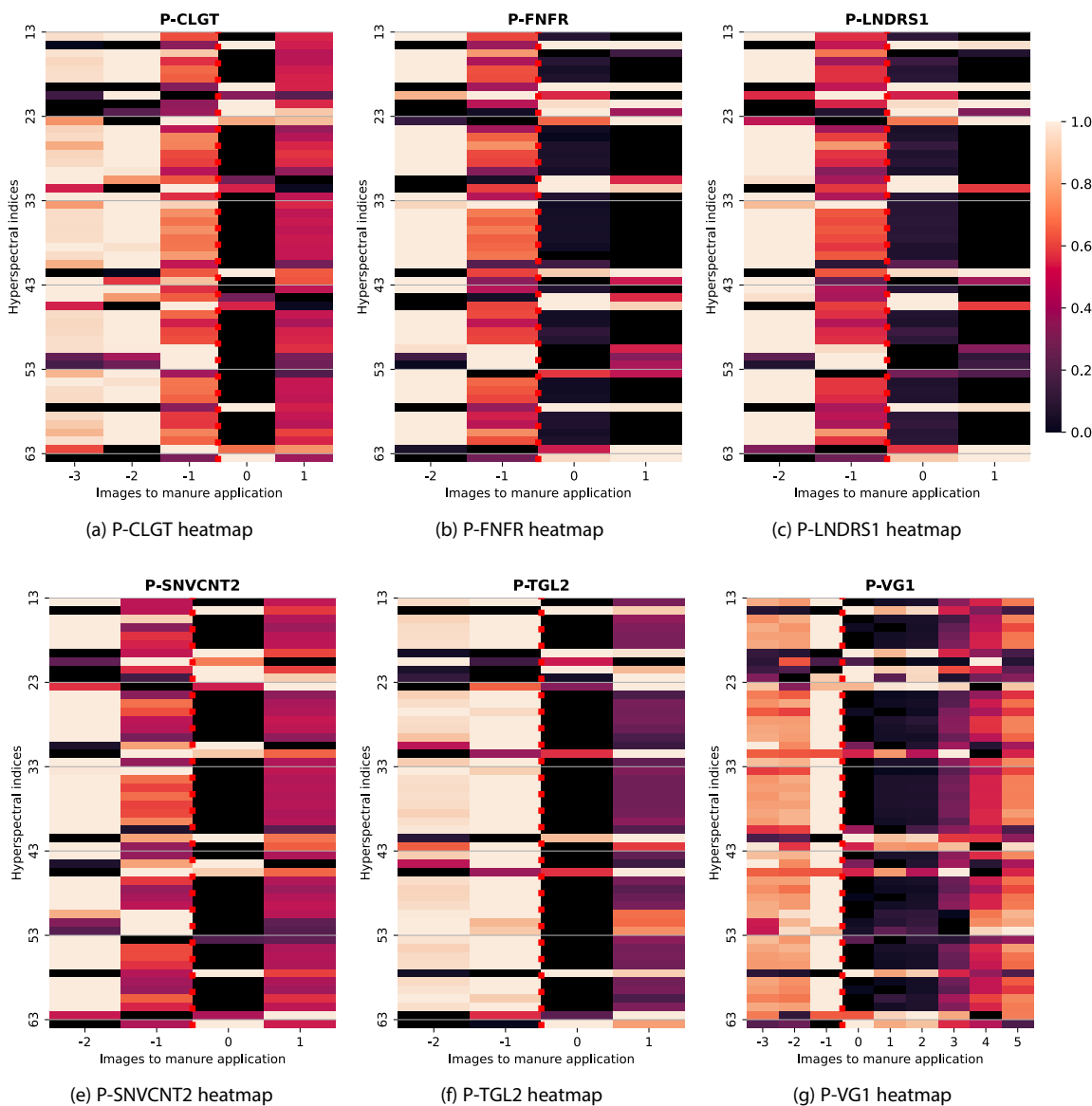


Fig. 10. Before and After manure application heatmaps for all vegetation indices.

class is 89.1%, indicating that this model is capable of successfully detecting manured fields on a per pixel basis.

Fig. 12 shows the visual representation of the manure detection in the test set. Some plots appear in the same image and have the same date of manure application. This is the case for P-MT and P-LLT, and for P-LNDRS1, P-LNDRS2, P-LNDRS3, and P-LNDRS4. The freshly manured plots are almost perfectly detected with few or no incorrectly classified pixels. Only a small region outside the ground truth is classified incorrectly as “Manure application” class. This occurs in all three images. However, some regions outside the groundtruth are unknown and cannot be used to judge the quality of the detection. This is only a visualization of the complete images of plots from the test set. These unknown regions do not affect the numerical metrics obtained above

since the unknown pixels are not used in the calculation of metrics. Nevertheless, the visual results appear to align with a human visual perception.

### 3.5. Discussion

The proposed method of using machine learning and remote sensing technology to detect freshly manured fields has the potential to be a useful tool for environmental conservation and management. The ability to monitor manure application and compliance with regulations can help to prevent nitrate leaching and protect nearby bodies of water from pollution. In addition, the proposed method can be used to improve the efficiency of manure management by identifying areas where manure is

**Table 7**  
Description of all feature sets.

Feature set	Description
A-13-S2	All the Sentinel-2 raw bands using the first image after manure application.
BA-26-S2	All the 13 Sentinel-2 raw bands from the first image after manure application and the image immediately before it, totalling 26 features.
BA-6-RGB	RGB raw bands from the first image after manure application and the image immediately before it, totalling 6 features.
BA-16-IR	All the infrared raw bands from the first image after manure application and the image immediately before it, totalling 16 features.
A-51-VI	All the calculated vegetation indices as features using the first image after manure application, totalling 51 features.
BA-102-VI	All the calculated vegetation indices as features using the first image after manure application and the image preceding it, totalling 102 features.
A-64	All 64 features (bands and vegetation indices) from the image immediately after the application of manure.
BA-128	All 64 features (bands and vegetation indices) from the image immediately after the application of manure and the image before it, totalling 128 features.
A-7-W	7 of the 64 features (bands and vegetation indices) from the image immediately after the application of manure. The 7 features were selected manually using the Ward clustering method.
BA-14-W	7 of the 64 features generated (vegetation indices and bands) from the image immediately following manure application and the image immediately preceding it. Each image has the same 7 features manually selected using the Ward clustering method, totalling 14 features per pixel.
BA-95%-PCA	Components of the PCA that extract 95% variability out of the 128 features (bands and vegetation indices) from the image immediately after the application of manure and the image immediately preceding it.
BA-90-RFE	90 of the 128 features (bands and vegetation indices) from the image immediately after the application of manure and the image immediately preceding it. The 90 features were selected by the Recursive Feature Elimination method.

**Table 8**  
Results of all the feature set experiments.

Feature set	Best classification method	Mean Precision	Mean Recall	Mean F1-Score	Accuracy
A-13-S2	Support Vector Machines	0.929	0.799	0.859	0.915
BA-26-S2	Discriminant Analysis	0.951	0.844	0.894	0.935
BA-6-RGB	Neural Network Classifiers	0.869	0.823	0.846	0.906
BA-16-IR	Support Vector Machines	0.912	0.863	0.887	0.931
A-51-VI	Support Vector Machines	0.930	0.839	0.882	0.928
BA-102-VI	Discriminant Analysis	0.952	0.916	0.934	0.959
A-64	Ensemble Classifiers	0.948	0.822	0.881	0.927
BA-128	Discriminant Analysis	0.935	0.907	0.921	0.951
A-7-W	Naïve Bayes Classifiers	0.825	0.824	0.824	0.888
BA-14-W	Support Vector Machines	0.882	0.814	0.847	0.908
BA-95%-PCA	Nearest Neighbor Classifiers	0.901	0.502	0.645	0.802
BA-10-PCA	Neural Network Classifiers	0.863	0.851	0.857	0.910
BA-90-RFE	Discriminant Analysis	0.960	0.910	0.934	0.959

being applied.

To train and test the detection models, 30 freshly manured plots were collected, totaling 31.48 hectares. All the plots in this study are pasture lands due to the nature of fields in northern Spain and were manually

validated by on-site investigations. This study was conducted in a specific region with a specific type of crops, but the proposed method could be applied in other regions with different types of crops and different environments. In the literature, it is common to see studies focused on a single crop type (Jaihuni et al., 2021; Romanko, 2017; Shou et al., 2007; Ye et al., 2022). This is because different crop types can have varying responses and may therefore require specific models in some cases. However, this can also be the result of the complexity of creating diverse datasets.

The results showed that the proposed method is able to fully detect all freshly manured plots in the test dataset obtaining an F1-Score superior to 90% for the tested area. It's important to note that the method relies on obtaining images shortly before and after manure application to achieve good results. Using time series from Sentinel-2 to improve results is a common practice in other precision agriculture fields such as soil mapping (Guo et al., 2021).

The study found that using a larger number of features improves the performance of classification models. The optimal number of features was found to be around 90 using the Recursive Feature Elimination method, which produces similar results to using all vegetation indices without raw bands.

The study also found that models trained using infrared bands (B5-B12) performed better than those trained using RGB bands (B2, B3, and B4) despite having less than half the spatial resolution. Studies such as Zhu et al. (2021b) have highlighted the importance of spatial resolution in the results and how the use of high-definition imagery from Gaofen-2 over multispectral images from Sentinel-2 and Landsat8 can provide better results for soil mapping. However, Gaofen-2 RGB imagery has a spatial resolution 3 times superior to that of Sentinel-2. It's worth noting that the use of vegetation indices can also make a difference in the results. Using only raw bands was not sufficient for accurate detection, vegetation indices are crucial for training a good model improving its accuracy over 5%.

This work demonstrates that there is no one perfect vegetation index to detect freshly manured fields, but rather a large number of them are needed to collect enough data for a machine learning model to generalize and detect manure in never-before-seen images. However, arbitrarily increasing the number of vegetation indices does not seem to be a feasible solution. A good combination is more effective than an exaggerated number of indices. To improve the results, the development of new and improved vegetation indices dedicated solely to manure detection could be investigated. In Dodin et al. (2021) several vegetation indices for organic matter (EOMI1, EOMI2, EOMI3, and EOMI4) were introduced. However, it was found that these indices alone were not sufficient to achieve the same level of performance as the combination of all 51 vegetation indices used in this study.

Other technologies such as SAR data from Sentinel-1 could also be used to add new information and further improve the results obtained. For example, Prasad et al. (2022) indicates that using both Sentinel-1 and Sentinel-2 yields the highest accuracy for land cover/land use.

It's important to note that the proposed method has limitations such as the reliance on high-resolution images with little cloud coverage and the need for obtaining images shortly before and after manure application. Additionally, the results of this study are specific to the region and type of crops studied and may not be generalizable to other regions or crops. Furthermore, the proposed method does not take into account other factors that could affect the accuracy of the results, such as soil conditions.

In order to further improve the proposed method, it would be beneficial to conduct studies in different regions with different types of crops and environments. This would allow for a better understanding of how the proposed method performs under different conditions and provide insights into how the method can be adapted to different contexts. Additionally, more research could be done on developing new and improved vegetation indices specifically designed for manure detection, as well as investigating the use of other technologies such as SAR to

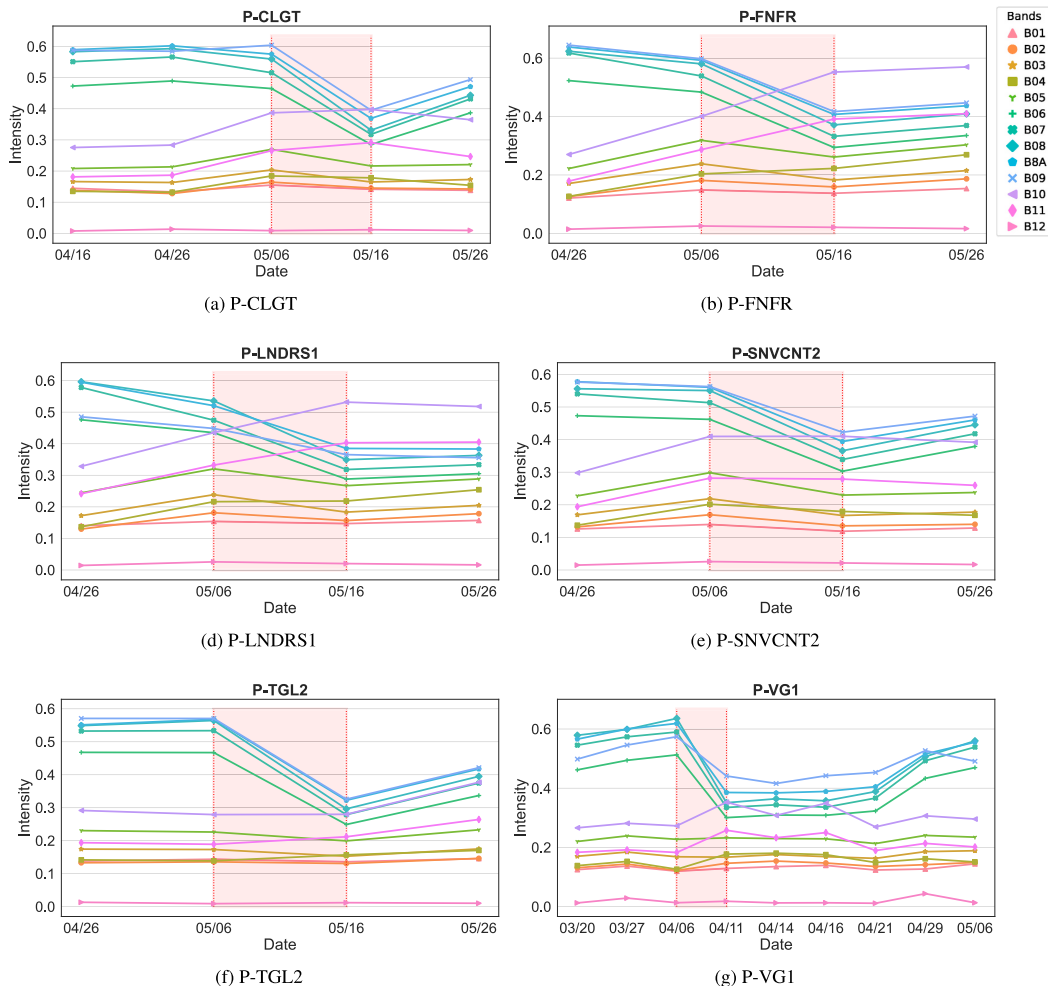


Fig. 11. Sentinel-2 bands intensity values for the time series of the plots P-CLGT, P-FNFR, P-LNDRS1, P-SNVCNT2, P-TGL2, and P-VG1.

Table 9  
Results per classification method for the BA-102-VI experiment.

Classification Method	Mean Precision	Mean Recall	Mean F <sub>1</sub> -Score	Accuracy
Decision Tree	0.833	0.769	0.800	0.881
<b>Discriminant Analysis</b>	<b>0.952</b>	<b>0.916</b>	<b>0.934</b>	<b>0.959</b>
Logistic Regression	0.838	0.911	0.873	0.904
Classifiers				
Naïve Bayes Classifiers	0.779	0.879	0.826	0.851
Support Vector Machines	<b>0.955</b>	0.836	0.891	0.933
Nearest Neighbor Classifiers	0.805	0.771	0.787	0.871
Kernels Approximation Classifiers	0.814	0.671	0.736	0.852
Ensemble Classifiers	0.953	0.884	0.917	0.949
Neural Network Classifiers	0.923	0.906	0.914	0.946

Table 10  
Results per class for Discriminant Analysis classification method of the BA-102-VI experiment.

Class	Precision	Recall	F <sub>1</sub> -Score
Others	0.962	0.987	0.974
Manure application	0.942	0.845	0.891

supplement the results obtained with remote sensing imagery.

In conclusion, this study has demonstrated that the use of machine learning and remote sensing technology can be a powerful tool for monitoring manure application and preventing nitrate leaching. However, the proposed method has limitations and further research is needed to improve the accuracy and generalizability of the results. The proposed method provides a valuable tool for precision agriculture and sustainable management of resources but it is important to consider its limitations and uncertainties.

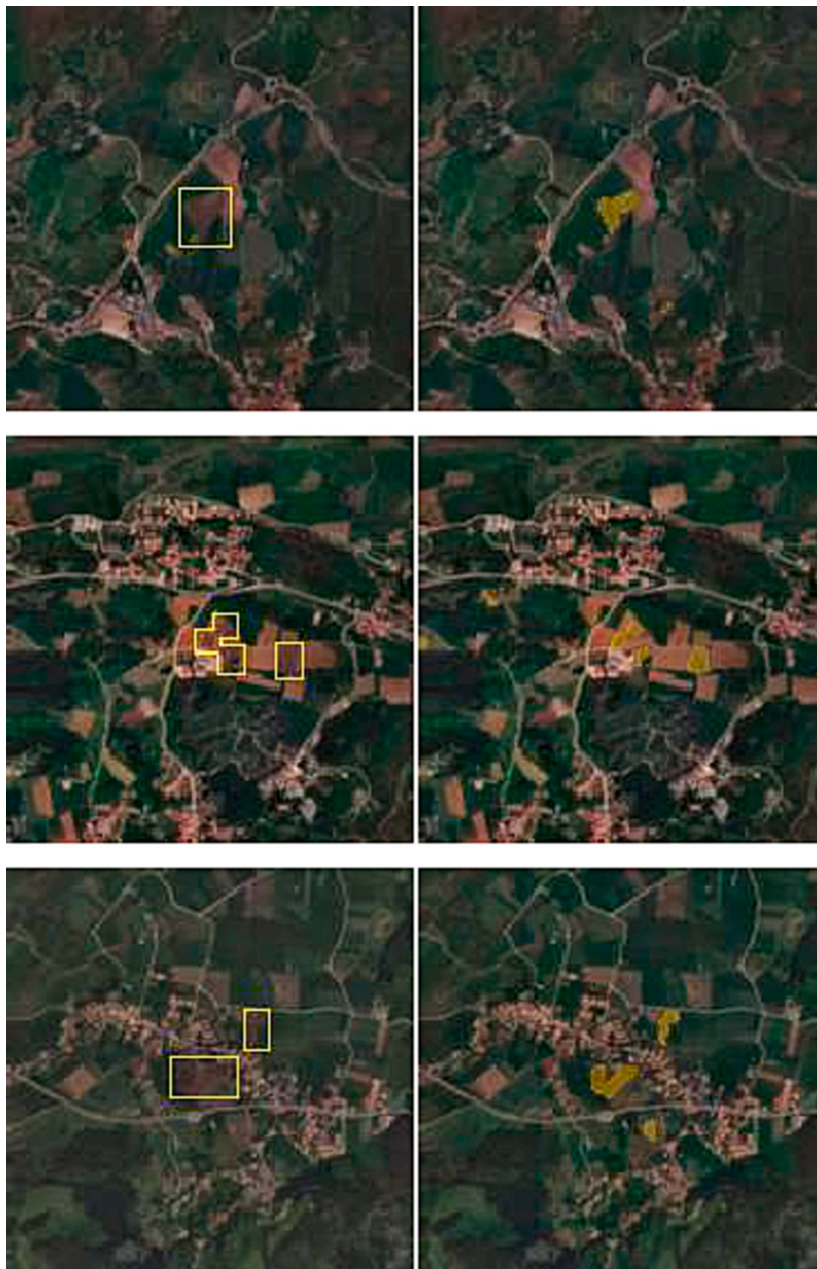


Fig. 12. Test set detections. Left column is the ground truth location and right column is the detection mask. P-CLGT in 1st row, P-LNDRS1/2/3/4 in 2nd row, and P-LLT and P-MT in 3rd row.

#### 4. Conclusion

This research analyzes machine learning based detection of freshly manured fields using vegetation indices from satellite images. This novel research can detect possible legal violations (Council of the European Union, 1991; Council of the European Union, 2000) in order to prevent leaching contamination caused by manure application in periods with

heavy rainfall, which accounts for about 60% of the pollution caused by excess nitrogen (Food and Agriculture of the United Nations, 2020). This study found that:

- It is possible to detect freshly manured fields using satellite imagery from Sentinel-2 with high accuracy.

- Obtaining imagery shortly before and after manure application yields the best results.
- A combination of several vegetation indices is necessary for optimal detection.

While the proposed method shows promising results, it is important to note that there are limitations to the study that should be considered when interpreting the results:

- The study is specific to the region and type of crops studied, and the results may not be generalizable to other regions or types of crops.
- This approach relies on the availability of clear imagery and this may limit the ability to obtain imagery in certain seasons or weather conditions.
- The study uses only one type of sensor and further research is needed to investigate the performance of the proposed method using other sensors such as SAR.

Overall, this study is an important step forward in the field of ecological informatics, providing a valuable tool for monitoring and managing manure application, and contributing to the preservation of the environment and water quality.

#### Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Data availability

The dataset is available at <https://doi.org/10.17632/fbvfv55kp.1>.

#### Acknowledgments

This work has been partially funded by the project PID2021-124383OB-I00 of the Spanish National Plan for Research, Development and Innovation.

#### References

- Alzu'bi, A., Alsmadi, L., 2022. Monitoring deforestation in Jordan using deep semantic segmentation with satellite imagery. *Ecol. Inform.* 70, 101745.
- Bagheri, N., Ahmadi, H., Alavipanah, S.K., Omid, M., 2013. Multispectral remote sensing for site-specific nitrogen fertilizer management. *Pesqui. Agropecu. Bras.* 48, 1394–1401.
- Bragagnolo, L., da Silva, R.V., Grzybowski, J.M.V., 2021. Towards the automatic monitoring of deforestation in Brazilian rainforest. *Ecol. Inform.* 66, 101454.
- Brugger, D., Windisch, W.M., 2015. Environmental responsibilities of livestock feeding using trace mineral supplements. *Anim. Nutr.* 1 (3), 113–118.
- Council of the European Union, 1991. Council directive 91/676/EEC of 12 December 1991 concerning the protection of waters against pollution caused by nitrates from agricultural sources. *Offic. J.* 375.
- Council of the European Union, 2000. Directive 2000/60/EC of the European Parliament and of the Council of 23 October 2000 establishing a framework for community action in the field of water policy. *Offic. J.*
- Culbertson, A.M., Martin, J.F., Aloysius, N., Ludsin, S.A., 2016. Anticipated impacts of climate change on 21st century Maumee river discharge and nutrient loads. *J. Great Lakes Res.* 42 (6), 1332–1342.
- Daloglu, I., Cho, K.H., Scavia, D., 2012. Evaluating causes of trends in long-term dissolved reactive phosphorus loads to Lake Erie. *Environ. Sci. Technol.* 46 (19), 10660–10666.
- Dodin, M., Smith, H.D., Levassasseur, F., Hadjar, D., Houot, S., Vaudour, E., 2021. Potential of Sentinel-2 satellite images for monitoring green waste compost and manure amendments in temperate cropland. *Remote Sens.* 13 (9), 1616.
- Fahmy, F., 2022. Pollution erodes fish stocks and livelihoods in Egyptian lake. <https://www.reuters.com/world/africa/pollution-erodes-fish-stocks-livelihoods-egyptian-lake-2022-09-01/>, accessed: 2022-09-07.
- Ferchichi, A., Abbas, A.B., Barra, V., Farah, I.R., 2022. Forecasting vegetation indices from spatio-temporal remotely sensed data using deep learning-based approaches: A systematic literature review. *Ecol. Inform.*, 101552.
- Fernandez-Moral, E., Martins, R., Wolf, D., Rives, P., 2018. A new metric for evaluating semantic segmentation: leveraging global and contour accuracy. In: 2018 IEEE intelligent vehicles symposium (iv). IEEE, pp. 1051–1056.
- Food and Agriculture Organization of the United Nations, 2020. Livestock and environment statistics: manure and greenhouse gas emissions. global, regional and country trends 1990–2018. FAOSTAT analytical briefs 14.
- Fu, Y., Yang, G., Pu, R., Li, Z., Li, H., Xu, X., Song, X., Yang, X., Zhao, C., 2021. An overview of crop nitrogen status assessment using hyperspectral remote sensing: Current status and perspectives. *Eur. J. Agron.* 124, 126241.
- Gillespie, A., 2022. NOAA forecasts summer 'dead zone' of nearly 5.4k square miles in Gulf of Mexico. <https://www.noaa.gov/news-release/noaa-forecasts-summer-dead-zone-of-nearly-54k-square-miles-in-gulf-of-mexico>, accessed: 2022-09-07.
- Guo, L., Sun, X., Fu, P., Shi, T., Dang, L., Chen, Y., Linderman, M., Zhang, G., Zhang, Y., Jiang, Q., et al., 2021. Mapping soil organic carbon stock by hyperspectral and time-series multispectral remote sensing images in low-relief agricultural areas. *Geoderma* 398, 115118.
- Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. *Mach. Learn.* 46 (1), 389–422.
- Jaihuini, M., Khan, F., Lee, D., Basak, J.K., Bhujel, A., Moon, B.E., Park, J., Kim, H.T., 2021. Determining spatiotemporal distribution of macronutrients in a cornfield using remote sensing and a deep learning model. *IEEE Access* 9, 30256–30266.
- Karl, T.R., Knight, R.W., 1998. Secular trends of precipitation amount, frequency, and intensity in the United States. *Bull. Am. Meteor. Soc.* 79 (2), 231–242.
- Kim, D.-K., Kaluskar, S., Mugalingam, S., Arhonditsis, G.B., 2016. Evaluating the relationships between watershed physiography, land use patterns, and phosphorus loading in the bay of Quinte basin, Ontario, Canada. *J. Great Lakes Res.* 42 (5), 972–984.
- Kirchman, D., 2022. Dead zones: growing areas of aquatic hypoxia are threatening our oceans and rivers. <https://blog.oup.com/2021/02/dead-zones-growing-areas-of-aquatic-hypoxia-are-threatening-our-oceans-and-rivers/>, accessed: 2022-09-07.
- Kleinman, P.J., Spiegel, S., Liu, J., Holly, M., Church, C., Ramirez-Avila, J., 2020. Managing animal manure to minimize phosphorus losses from land to water. In: *Animal Manure: Production, Characteristics, Environmental Concerns, and Management*, vol. 67, pp. 201–228.
- Kursa, M.B., Rudnicki, W.R., 2010. Feature selection with the Boruta package. *J. Stat. Softw.* 36, 1–13.
- Liu, J., Kleinman, P.J., Aronsson, H., Flaten, D., McDowell, R.W., Bechmann, M., Beegle, D.B., Robinson, T.P., Bryant, R.B., Liu, H., et al., 2018. A review of regulations and guidelines related to winter manure application. *Ambio* 47 (6), 657–670.
- Ma, Q., Yu, W., Zhou, H., 2010. The relationship between soil nutrient properties and remote sensing indices in the phaeozem region of northeast China. In: 2010 Second International Conference on Computational Intelligence and Natural Computing, vol. 2. IEEE, pp. 109–112.
- Mutanga, O., Dube, T., Galal, O., 2017. Remote sensing of crop health for food security in Africa: Potentials and constraints. *Remote Sens. Appl.: Soc. Environ.* 8, 231–239.
- Orynbaikyzy, A., Gessner, U., Conrad, C., 2019. Crop type classification using a combination of optical and radar remote sensing data: A review. *Int. J. Remote Sens.* 40 (17), 6553–6595.
- Pedrayes, O.D., Lema, D.G., García, D.F., Usamentiaga, R., Alonso, Á., 2021. Evaluation of semantic segmentation methods for land use with spectral imaging using Sentinel-2 and PNOA imagery. *Remote Sens.* 13 (12), 2292.
- Prasad, P., Lovesson, V.J., Chandra, P., Kotha, M., 2022. Evaluation and comparison of the earth observing sensors in land cover/land use studies using machine learning algorithms. *Ecol. Inform.* 68, 101522.
- Romanko, M., 2017. Remote sensing in precision agriculture: Monitoring plant chlorophyll, and soil ammonia, nitrate, and phosphate in corn and soybean fields. Ph.D. thesis. Bowling Green State University.
- Sentinel-Hub, 2022. Sentinel-2 rs indices – sentinel-hub custom scripts. <https://custom-scripts.sentinel-hub.com/custom-scripts/sentinel-2/indexdb/>, accessed: 2022-12-06.
- Shanmugapriya, P., Rathika, S., Ramesh, T., Janaki, P., 2019. Applications of remote sensing in agriculture—a review. *Int. J. Curr. Microbiol. Appl. Sci.* 8 (1), 2270–2283.
- Shou, L., Jia, L., Cui, Z., Chen, X., Zhang, F., 2007. Using high-resolution satellite imaging to evaluate nitrogen status of winter wheat. *J. Plant Nutr.* 30 (10), 1669–1680.
- Sishodia, R.P., Ray, R.L., Singh, S.K., 2020. Applications of remote sensing in precision agriculture: A review. *Remote Sens.* 12 (19), 3136.
- Spanish Ministry of Agriculture Food and Environment, 2007. Geographic information system for agricultural land plots (SIGPAC). <https://www.mapa.gob.es/es/agricultura/temas/sistema-de-informacion-geografica-de-parcelas-agricolas-sigpac/-default.aspx>, accessed: 2022-09-07.
- Tzilivakakis, J., Warner, D., Green, A., Lewis, K., 2021. A broad-scale spatial analysis of the environmental benefits of fertilizer closed periods implemented under the nitrates directive in Europe. *J. Environ. Manage.* 299, 113674.
- Wang, J., 2009. Satellite mapping of past biosolids (sewage sludge) and animal manure application to agriculture fields in Wood County, Ohio. Ph.D. thesis. Bowling Green State University.
- Wold, S., Esbensen, K., Geladi, P., 1987. Principal component analysis. *Chemom. Intell. Lab. Syst.* 2 (1–3), 37–52.
- Xia, C., Zhang, Y., 2022. Comparison of the use of Landsat 8, Sentinel-2, and Gaofen-2 images for mapping soil pH in Dehui, northeastern China. *Ecol. Inform.*, 101705.
- Yang, C.-C., Prasher, S.O., Whalen, J., Goel, P.K., 2002. Pa-precision agriculture: use of hyperspectral imagery for identification of different fertilisation methods with decision-tree technology. *Biosyst. Eng.* 83 (3), 291–298.



Ye, W., Lao, J., Liu, Y., Chang, C.-C., Zhang, Z., Li, H., Zhou, H., 2022. Pine pest detection using remote sensing satellite images combined with a multi-scale attention-unet model. *Ecol. Inform.* 72, 101906.

Zhu, W., Rezaei, E.E., Nouri, H., Yang, T., Li, B., Gong, H., Lyu, Y., Peng, J., Sun, Z., 2021a. Quick detection of field-scale soil comprehensive attributes via the integration of uav and sentinel-2b remote sensing data. *Remote Sens.* 13 (22), 4716.

Zhu, W., Rezaei, E.E., Nouri, H., Yang, T., Li, B., Gong, H., Lyu, Y., Peng, J., Sun, Z., 2021b. Quick detection of field-scale soil comprehensive attributes via the integration of uav and sentinel-2b remote sensing data. *Remote Sens.* 13 (22), 4716.

## 5.2. Publicaciones en revisión: JCR (Journal Citation Reports)

### 5.2.1. Evaluation of remote sensing spectral indices for manure application in pasture fields

- El índice de impacto de la revista *IEEE Transactions on Geoscience and Remote Sensing* en 2021 fue 8.125 (Q1, 94.83%) y el índice de impacto a 5 años, 8.137.

## 5.3. Publicaciones: JCI (*Journal Citation Indicator*)

### 5.3.1. Satellite imagery dataset of manure application on pasture fields

- Pedrayes, O. D., & Usamentiaga, R. (2023). *Satellite imagery dataset of manure application on pasture fields*. *Data in Brief*, 46, 108786.
- DOI: [10.1016/j.dib.2022.108786](https://doi.org/10.1016/j.dib.2022.108786)
- El *Journal Citation Indicator* (JCI) de la revista *Data in Brief* en 2021 fue 0.28 (Q3, 48.52%) y su Clasificación Integrada de Revistas Científicas (CIRC) fue C.



ELSEVIER

Contents lists available at ScienceDirect

## Data in Brief

journal homepage: [www.elsevier.com/locate/dib](http://www.elsevier.com/locate/dib)

## Data Article

## Satellite imagery dataset of manure application on pasture fields



Oscar D. Pedrayes, Rubén Usamentiaga\*

*Department of Computer Science and Engineering, University of Oviedo, Campus de Viesques, Gijón, Asturias 33204, Spain*

## ARTICLE INFO

*Article history:*

Received 9 August 2022

Revised 2 November 2022

Accepted 24 November 2022

Available online 29 November 2022

*Keywords:*

Fertilizer

Land

Slurry

Crops

Semantic segmentation

Classification

Precision agriculture

## ABSTRACT

Applying manure to pasture fields is a very common method of fertilization. However, rainfall can cause the manure to leach into water bodies near the field, contaminating the water and damaging the environment and the animals living in it, ultimately affecting human life. This paper presents a dataset consisting of images of 30 plots after manure application, verified by on-site investigations. This involved visiting 38 different plots, of which 8 were discarded because they were not suitable, either because of their small size, the lack of a specific manure application date, or the images being too cloudy in that period. The imagery is collected through Google Earth Engine using the satellite Sentinel-2, which offers 13 hyperspectral bands in the range of ultraviolet and near-infrared wavelengths including the visible spectrum. From these 13 bands, the most common hyperspectral indices in the literature for precision agriculture are calculated and added into the images as channels. 51 hyperspectral indices are calculated, summing up to a total of 64 channels per image when adding the raw bands from Sentinel-2. No normalization has been performed on any of the channels. The data can be used for further research of automatic classification of manure application to control its use and prevent contamination.

\* Corresponding author.

E-mail addresses: [UO251056@uniovi.es](mailto:UO251056@uniovi.es) (O.D. Pedrayes), [rusamentiaga@uniovi.es](mailto:rusamentiaga@uniovi.es) (R. Usamentiaga).

## Specifications Table

Subject	Agronomy and Crop Science
Specific subject area	Remote sensing for precision agriculture to detect and classify recently manured pasture fields.
Type of data	Image
How the data were acquired	All data is acquired through Google Earth Engine. Plots are manually selected in Google Earth Engine after an on-site investigation. Images are downloaded from the satellite Sentinel-2 using Google Earth Engine. Finally, cloudy images are filtered out manually.
Data format	Raw Filtered Processed
Description of data collection	The regions of interest with the considered plots in the images are located after careful on-site inspection and verification. When a manured field is found, photographs are taken as validation and the location is indicated by GPS. Then, from Google Earth Engine, the appropriate region is manually selected, and the corresponding Sentinel-2 images are downloaded from the date on which the plot was manured, or the closest possible later date.
Data source location	City/Town/Region: Northern region of Spain Country: Spain Latitude and longitude: latitude around [43.55, 43.38], and longitude around [-5.50, -4.10].
Data accessibility	All images are obtained from the satellite Sentinel-2. Repository name: Mendeley Data Data identification number: <a href="https://doi.org/10.17632/fbvvf55kp.1">https://doi.org/10.17632/fbvvf55kp.1</a> Direct URL to data: <a href="https://data.mendeley.com/datasets/fbvvf55kp">https://data.mendeley.com/datasets/fbvvf55kp</a>

## Value of the Data

- The development of such a dataset is costly and time-consuming, as on-site investigations are necessary to verify manure application and to accurately select the plot. In addition, clouds and other problems, such as plots that are too small, must be filtered out.
- This dataset can be used to train machine learning models to automatically detect manured fields to analyze illegal fertilization or hot spots. This provides an opportunity for further research on this topic.
- Each plot has multiple images from different dates from before and after manure application. This offers the opportunity to investigate classification methods that benefit from temporal analysis. The differences in terrain depending on its date can be considerable, which adds a substantial amount of information to the status of the plot.
- The imagery contains the most relevant hyperspectral indices in the literature for precision agriculture and provides all 13 Sentinel-2 bands from which more hyperspectral indices can be created if needed.
- There is no other dataset of this type in the literature for this particular problem. Moreover, even if other datasets were created, this data would still be useful, as it belongs to a particular region and crop type which could be used to complete other datasets or to validate results.

## 1. Data Description

The dataset consists of three folders: the “src” folder, where all the code to generate the dataset is stored; the “groundtruth” folder, which contains an image mask for each plot; and the “imagery” folder which contains images with the satellite imagery raw bands and the calculated hyperspectral indices. The ground truth images are in “.png” format and follow a color code:

- White (255, 255, 255): Plot of interest
- Black (0,0,0): Other

The “imagery” folder contains a folder for each plot. Each plot folder contains another two folders, one for the images from before the application of manure and another one for the images from after manure application. Every image is in “.tif” format and has 64 channels. The order of the channels and how to calculate them can be found in the “Experimental design, materials and methods” section.

All the plots obtained for this dataset are pastures. This is because pasture is the predominant type of crop in this area of northern Spain. In most cases the grass is mowed prior to manure application. Although in some of the plots the manure is applied directly on the plowed land. This could prevent the trained models from confusing plowed lands and manure. Images of the plots have an area of about  $1700 \times 1700$  m, although the plots inside the images are smaller. A total of 38 plots are studied.

Table 1 summarizes every plot of interest in the dataset, showing the date of manure application, area in square meters, number of available images for each plot from before and after manure application, its suitability for further studies, and its geographical coordinates. The plot identifier is composed of “P-” plus the abbreviation (using only the consonants) of the locality in which the plot is located. The area of the plots is calculated after generating the ground truth mask, where each Sentinel-2 pixel counts as 100 m<sup>2</sup>. The suitability is assessed after studying the Sentinel-2 images of the plot in question. For example, if the region is extremely small, it is discarded.

**Table 1**  
Dataset summary.

Plot	Date (YYYY/MM/dd)	Area (m <sup>2</sup> )	Available images (Before/After)	Suitable (Yes/No)	Geographical Coordinates (Long/Lat)	
P-BLD	2022/05/26	8900	2/1	Yes	-4.2018	43.3973
P-BLLT1	2022/05/16	21,200	2/2	Yes	-4.0840	43.4309
P-BLLT2	2022/05/26	3300	2/1	Yes	-4.0840	43.4310
P-Cardana	2022/02/24	6500	8/9	Yes	8.6580	45.8592
P-CBRCS1	2022/05/26	6700	2/1	Yes	-4.2005	43.3897
P-CBRCS2	2022/05/26	6400	2/1	Yes	-4.2048	43.3875
P-CLGT	2022/05/16	17,200	3/2	Yes	-4.1096	43.3987
P-CLMBRS	2022/05/26	4300	3/1	Yes	-4.5447	43.3804
P-CMNTR	2022/05/16	2600	1/2	Yes	-4.1470	43.4001
P-DR.	2022/03/21	2500	1/5	Yes	-4.1424	43.3967
P-FNFR	2022/05/16	10,100	2/2	Yes	-4.2657	43.3880
P-GLS	2022/04/30	7800	2/-	No (Clouds)	-4.1452	43.3996
P-LLT	2022/05/03	9600	2/1	Yes	-4.1515	43.4001
P-LNDRS1	2022/05/16	3200	2/2	Yes	-4.2510	43.3880
P-LNDRS2	2022/05/16	5400	2/2	Yes	-4.2503	43.3880
P-LNDRS3	2022/05/16	8500	2/2	Yes	-4.2497	43.3872
P-LNDRS4	2022/05/16	9100	2/2	Yes	-4.2467	43.3877
P-LNDRS5	-	5100	2/2	No (application date unclear)	-4.2435	43.3864
P-MT	2022/05/04	19,900	2/1	Yes	-4.1536	43.3980
P-NMS	2022/02/10	5500	2/1	Yes	-4.1490	43.4003

(continued on next page)

**Table 1** (continued)

Plot	Date (YYYY/MM/dd)	Area (m <sup>2</sup> )	Available images (Before/After)	Suitable (Yes/No)	Geographical Coordinates (Long/Lat)	
P-PQN	2022/02/27	5300	-/2	<b>No.</b> (Clouds)	-4.1495	43.3991
P-PSG	2022/04/06	5400	3/2	<b>No.</b> (Too narrow, partly fertilized)	-4.1411	43.3970
P-QNTLS1	-	13,600	7/3	<b>No</b> (application date unclear)	-5.5830	43.5463
P-QNTLS2	2022/05/16	8500	7/3	<b>Yes</b>	-5.5840	43.5458
P-SNTLLN	2022/03/17	14,200	2/4	<b>Yes</b>	-4.1170	43.3935
P-SNVCNT1	2022/05/16	6700	2/2	<b>Yes</b>	-4.4048	43.3939
P-SNVCNT2	2022/05/16	29,200	2/2	<b>Yes</b>	-4.4001	43.3945
P-STBN	2022/05/04	11,300	3/1	<b>Yes</b>	-4.1366	43.3960
P-TGL1	-	28,000	2/1	<b>No</b> (application date unclear)	-4.0695	43.4216
P-TGL2	2022/05/16	12,300	2/2	<b>Yes</b>	-4.0701	43.4276
P-TMSN	2022/02/10	4700	-	<b>No</b> (Clouds)	-4.1519	43.3996
P-TNNS1	2022/05/26	19,500	2/1	<b>Yes</b>	-4.1871	43.3999
P-TNNS2	2022/05/06	11,100	1/2	<b>Yes</b>	-4.1918	43.3987
P-TPRN	2022/04/06	1800	3/3	<b>No.</b> (Too narrow)	-4.1390	43.3965
P-VG1	2022/04/09	12,200	3/6	<b>Yes</b>	-5.4866	43.4699
P-VG2	2022/04/13	4900	4/5	<b>Yes</b>	-5.4801	43.4693
P-VLDMR	2022/02/07	17,500	2/2	<b>Yes</b>	-4.1561	43.4056
P-VNS	2022/04/23	16,600	3/2	<b>Yes</b>	-4.1504	43.4042

The complete dataset consists of 31.48 ha for the plots of interest. Each pixel is 0.01 ha.

## 2. Experimental Design, Materials and Methods

The first indications of a newly manured plot are given by people living in the area or by Sentinel-2 imagery surveys. To label the plots, first, an on-site investigation is carried out to confirm that the plot has been fertilized with manure and to observe the real dimensions of the fertilization in the plot. Then, using Google Earth Engine, the plot is annotated according to the observed dimensions. An example of this process is shown in Fig. 1.

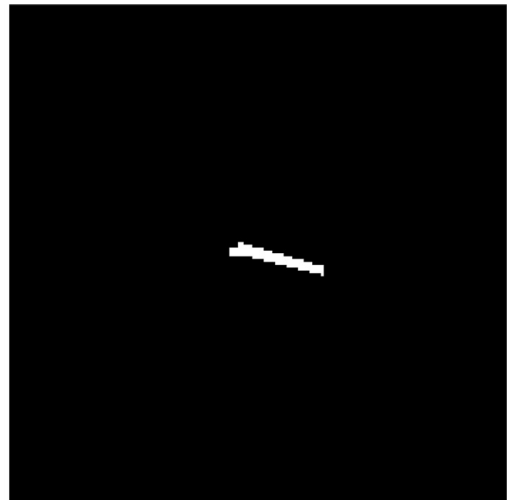
Google Earth Engine is also used to download the imagery with the script “download\_imagery.js”. The plots are then exported as KML files and used to generate ground truth masks by taking advantage of the georeferenced Sentinel-2 imagery, as shown in Fig. 2. The script used to generate the ground truth masks from the KML files is called “generate\_groundtruth.py”

To generate the images of the dataset, the first 13 channels are obtained directly from the 13 bands of the Sentinel-2 images. Sentinel-2 has two satellites in its orbit (Sentinel-2A and Sentinel2B), where each one has an orbit of 10 days. Their orbits are at the greatest distance from each other, which is why the acquisition time of the images for the same region is around 5 days. Table 2 shows the wavelengths and bandwidths for each band in  $\mu\text{m}$  for S-2A and S-2B separately.

The remaining 51 channels of the images from the dataset are hyperspectral indices which are calculated as different combinations of the 13 Sentinel-2 raw bands. To obtain these hyperspectral indices, the general literature of precision agriculture for fertilizers using satellite imagery has been studied [1–9] and the most relevant hyperspectral indices have been obtained. The script necessary to generate the hyperspectral indices is called “calculate\_indices.py”. Table 3 shows how to calculate each hyperspectral index and its channel number in the images.



**Fig. 1.** On-site investigation (left). Plot annotated in Google Earth Engine (right).



**Fig. 2.** Sentinel-2 georeferenced image (left). Generated ground truth (right).

To get an idea of the final images, an example of a visualization of one of the plots is shown. [Fig. 3](#) shows at the left the total area of the image in RGB, and at the right, an enlarged version. [Fig. 4](#) shows each of the 64 channels in a black and white color scale. To better visualize the plot, an enlarged version is shown in [Fig. 5](#).

**Table 2**  
Sentinel-2 bands (Sentinel2A and Sentinel2B).

#	Band	Central Wavelength ( $\mu\text{m}$ )		Bandwidth ( $\mu\text{m}$ )		Spatial resolution (m)
		S-2A	S-2B	S-2A	S-2B	
0	B01 Coastal aerosol	0.4427	0.4422	0.021	0.021	60
1	B02 Blue	0.4924	0.4921	0.066	0.066	10
2	B03 Green	0.5598	0.5590	0.036	0.036	10
3	B04 Red	0.6646	0.6649	0.031	0.031	10
4	B05 VRE	0.7041	0.7038	0.015	0.016	20
5	B06 VRE	0.7405	0.7391	0.015	0.015	20
6	B07 VRE	0.7828	0.7797	0.020	0.020	20
7	B08 NIR	0.8328	0.8329	0.106	0.106	10
8	B8A Narrow Nir	0.8647	0.8640	0.021	0.022	20
9	B09 Water vapor	0.9451	0.9432	0.020	0.021	60
10	B10 SWIR Cirrus	1.3735	1.3769	0.031	0.030	60
11	B11 WIR	1.6137	1.6104	0.091	0.094	20
12	B12 SWIR	2.2024	2.1857	0.175	0.185	20

**Table 3**  
Hyperspectral indices.

#	Abb.	Name	Description
13	<b>NDVI</b>	Normalized Difference Vegetation Index	$\frac{B08 - B04}{B08 + B04}$
14	<b>NSNDVI</b>	NIR-SWIR Normalized Difference Vegetation Index	$\frac{B11 - B07}{B11 + B07}$
15	<b>SDI</b>	Swir Difference Index	$B08 - B12$
16	<b>GNDVI</b>	Green Normalized Difference Vegetation Index	$\frac{B08 - B03}{B08 + B03}$
17	<b>SAVI</b>	Soil Adjusted Vegetation Index	$\frac{B08 - B04}{B08 + B04 + 0.428} * 1.428$
18	<b>OSAVI</b>	Optimized Soil Adjusted Vegetation Index	$(1 + 0.16) * \frac{B08 - B04}{B08 + B04 + 0.16}$
19	<b>EOMI1</b>	Exogenous Organic Matter Index 1	$\frac{B11 - B8A}{B11 + B8A}$
20	<b>EOMI2</b>	Exogenous Organic Matter Index 2	$\frac{B12 - B04}{B12 + B04}$
21	<b>EOMI3</b>	Exogenous Organic Matter Index 3	$\frac{(B11 - B8A) + (B12 + B04)}{B11 + B8A + B12 + B04}$
22	<b>EOMI4</b>	Exogenous Organic Matter Index 4	$\frac{B11 - B04}{B11 + B04}$
23	<b>BNR2</b>	Normalized Burn Ratio 2	$\frac{B11 - B12}{B11 + B12}$
24	<b>RFI</b>	Ratio Vegetation Index	$\frac{B08}{B04}$
25	<b>DVI</b>	Difference Vegetation Index	$B08 - B04$
26	<b>RENDVI1</b>	Red Edge Normalized Difference Vegetation Index	$\frac{B05 - B04}{B05 + B04}$
27	<b>RENDVI2</b>	Red Edge Normalized Difference Vegetation Index	Same as RENDVI1, but uses B06 instead of B05
28	<b>RENDVI3</b>	Red Edge Normalized Difference Vegetation Index	Same as RENDVI1, but uses B07 instead of B05
29	<b>CI1</b>	Chlorophyll Index	$\frac{B08}{B05} - 1$
30	<b>CI2</b>	Chlorophyll Index	Same as CI1, but uses B06 instead of B05
31	<b>CI3</b>	Chlorophyll Index	Same as CI1, but uses B07 instead of B05
32	<b>NDRE</b>	Normalized Difference Red Edge	$\frac{B08 - B05}{B08 + B05}$

(continued on next page)



**Table 3** (continued)

#	Abb.	Name	Description
33	<b>MCARI</b>	Modified Chlorophyll Absorption in Reflectance Index	$((B05 - B04) - 0.2 * (B05 - B03)) * \frac{B05}{B04}$
34	<b>MCARI1</b>	Modified Chlorophyll Absorption in Reflectance Index 1	$1.2 * (2.5 * (B08 - B04) - 1.3 * (B08 - B03))$
35	<b>MCARI2</b>	Modified Chlorophyll Absorption in Reflectance Index 2	$1.5 * \frac{2.5 * (B08 - B04) - 1.3 * (B08 - B03)}{\sqrt{(2 * B08 + 1)^2 - (6 * B08 - 5 * \sqrt{B04})} - 0.5}$
36	<b>MTVI1</b>	Modified Triangular Vegetation Index 1	$1.2 * (1.2 * (B08 - B03) - 2.5 * (B04 - B03))$
37	<b>MTVI2</b>	Modified Triangular Vegetation Index 2	$1.5 * \frac{1.2 * (B08 - B03) - 2.5 * (B08 - B03)}{\sqrt{(2 * B08 + 1)^2 - (6 * B08 - 5 * \sqrt{B04})} - 0.5}$
38	<b>EVI</b>	Enhanced Vegetation Index	$\frac{2.5 * (B08 - B04)}{(B08 + 6 * B04 - 7.5 * B02) + 1}$
39	<b>AVI</b>	Advanced Vegetation Index	$(B08 * (1 - B04) * (B08 - B04))^{1/3}$
40	<b>GCI</b>	Green Coverage Index	$\frac{B09}{B03} - 1$
41	<b>BSI</b>	Bare Soil Index	$B11 + B04 + \frac{B08 + B02}{B11 + B04} + B08 + B02$
42	<b>NBRI</b>	Normalized Burned Ratio Index	$\frac{B08 - B12}{B08 + B12}$
43	<b>NDRE1</b>	Normalized Difference Red Edge	$\frac{B08 - B05}{B08 + B05}$
44	<b>NDRE2</b>	Normalized Difference Red Edge	Same as NDRE1, but uses B06 instead of B05
45	<b>NDRE3</b>	Normalized Difference Red Edge	Same as NDRE1, but uses B07 instead of B05
46	<b>MSAVI</b>	Modified Soil Adjusted Vegetation Index	$\frac{(2.0 * B08 + 1 - \sqrt{(2.0 * B08 + 1.0)^2 - 8 * (B08 - B04)})}{2}$
47	<b>WDRVI</b>	Wide Dynamic Range Vegetation Index	$\frac{0.1 * B08 - B04}{0.1 * B08 + B04}$
48	<b>ARVI1</b>	Atmospherically Resistant Vegetation Index 1	$\frac{B8A - B04 - 0.069 * (B04 - B02)}{B8A + B04 - 0.069 * (B04 - B02)}$
49	<b>ARVI2</b>	Atmospherically Resistant Vegetation Index 2	$-0.18 + 1.17 * \frac{B8 - B4}{B8 + B4}$
50	<b>TSAVI</b>	Transformed Soil Adjusted Vegetation Index	$\frac{(0.421 * (B08 - 0.421 * B04 - 0.824))}{(B04 + 0.421 * (B08 - 0.824) + 0.114 * (1 + 0.421)^2)}$
51	<b>CARI1</b>	Chlorophyll Absorption Ratio Index 1	$\frac{B05}{B04} * \frac{ (B05 - B03) / 150 * 670.0 + B04 + (B03 - ((B05 - B03) * 550)) }{\sqrt{(B05 - B03) / 150^2 + 1}}$
52	<b>CARI2</b>	Chlorophyll Absorption Ratio Index 2	$\frac{ (B05 - B03) / 150 * B04 + B04 + B03 - 0.496 * B03 }{\sqrt{(0.496^2 + 1)} * (B05 / B04)}$
53	<b>CVI</b>	Chlorophyll Vegetation Index	$\frac{B08 * B04}{B03^2}$
54	<b>EVI1</b>	Enhanced Vegetation Index 1	$\frac{2.5 * (B08 - B04)}{(B08 + 6 * B04 - 7.5 * B02) + 1}$
55	<b>EVI2</b>	Enhanced Vegetation Index 2	$2.4 * \frac{B08 - B04}{B08 + B04 + 1}$
56	<b>EVI3</b>	Enhanced Vegetation Index 3	$2.5 * \frac{B08 - B04}{B08 + 2.4 * B04 + 1}$
57	<b>SCI</b>	Soil Composition Index	$\frac{B11 - B08}{B11 + B08}$
58	<b>GRNDVI</b>	Green-Red Normalized Difference Vegetation Index	$\frac{B08 - (B03 + B04)}{B08 + (B03 + B04)}$
59	<b>GBNDVI</b>	Green-Blue Normalized Difference Vegetation Index	$\frac{B08 - (B03 + B02)}{B08 + (B03 + B02)}$
60	<b>GLI</b>	Green Leaf Index	$\frac{2 * B03 - B04 - B02}{2 * B03 + B04 + B02}$
61	<b>ATSAVI</b>	Adjusted Transformed Soil-Adjusted Vegetation Index	$\frac{1.22 * (B08 - 1.22 * B04 - 0.03)}{1.22 * B08 + B04 - 1.22 * 0.03 + 0.08 * (1 + 1.22^2)}$
62	<b>ALTERATION</b>	Alteration Index	$\frac{B11}{B12}$
63	<b>CTVI</b>	Corrected Transformed Vegetation Index	$\frac{(B04 - B03) / (B04 + B03) + 0.5}{  \frac{B04 - B03}{B04 + B03}   + 0.5 * \sqrt{  \frac{B04 - B03}{B04 + B03} + 0.5  }}$



Fig. 3. Sentinel-2 image of a manured plot. Total area of the image plot (left). Enlarged plot (right).

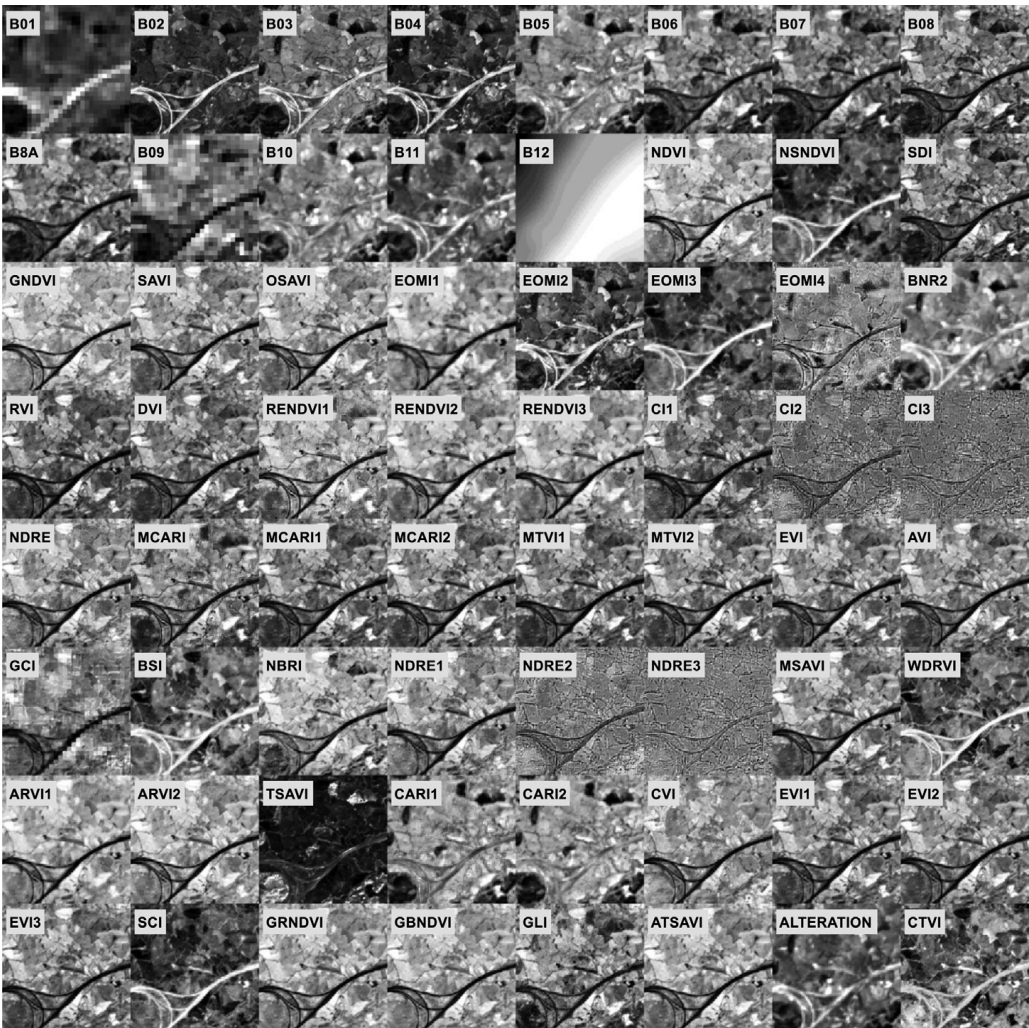


Fig. 4. Example of the 64 channels, including raw band and computed indices.

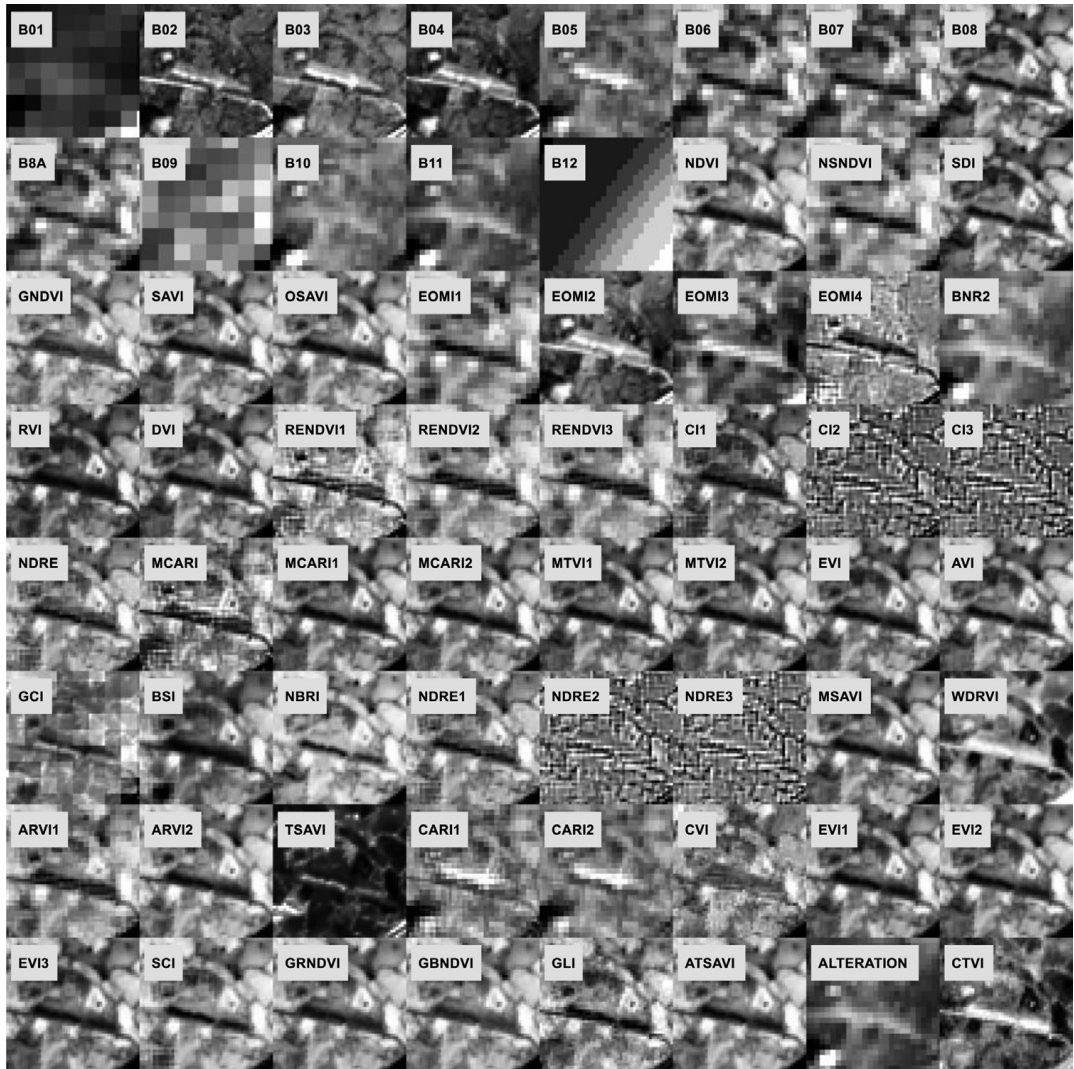


Fig. 5. Example of the 64 channels, including raw bands and computed indices. (Enlarged plot).

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data Availability

[Satellite imagery dataset of manure application on pasture fields \(Original data\)](#) (Mendeley Data).

## CRediT Author Statement

**Oscar D. Pedrayes:** Conceptualization, Methodology, Software, Visualization, Investigation, Writing – original draft; **Rubén Usamentiaga:** Data curation, Supervision, Validation, Writing – review & editing.

## Acknowledgments

This work has been partially funded by the project PID2021-124383OB-I00 of the Spanish National Plan for Research, Development and Innovation.

## References

- [1] Y. Fu, G. Yang, R. Pu, Z. Li, H. Li, X. Xu, C. Zhao, An overview of crop nitrogen status assessment using hyperspectral remote sensing: current status and perspectives, *Eur. J. Agron.* 124 (2021) 126241.
- [2] M. Dodin, H.D. Smith, F. Levvasseur, D. Hadjar, S. Houot, E. Vaudour, Potential of Sentinel-2 satellite images for monitoring green waste compost and manure amendments in temperate cropland, *Remote Sens.* 13 (9) (2021) 1616.
- [3] N. Bagheri, H. Ahmadi, S.K. Alavipanah, M. Omid, Multispectral remote sensing for site-specific nitrogen fertilizer management, *Pesqui. Agropecu. Bras.* 48 (2013) 1394–1401.
- [4] D.G. Lema, O.D. Pedrayes, R. Usamentiaga, D.F. García, Á. Alonso, Cost-performance evaluation of a recognition service of livestock activity using aerial images, *Remote Sens.* 13 (12) (2021) 2318.
- [5] Q. Ma, W. Yu, H. Zhou, The relationship between soil nutrient properties and remote sensing indices in the Phaeozem region of Northeast China, in: *Proceedings of the Second International Conference on Computational Intelligence and Natural Computing*, 2, IEEE, 2010, pp. 109–112.
- [6] Romanko, M. (2017). *Remote sensing in precision agriculture: monitoring plant chlorophyll, and soil ammonia, nitrate, and phosphate in corn and soybean fields* (Doctoral dissertation, Bowling Green State University).
- [7] L. Shou, L. Jia, Z. Cui, X. Chen, F. Zhang, Using high-resolution satellite imaging to evaluate nitrogen status of winter wheat, *J. Plant Nutr.* 30 (10) (2007) 1669–1680.
- [8] R.P. Sishodia, R.L. Ray, S.K. Singh, Applications of remote sensing in precision agriculture: a review, *Remote Sens.* 12 (19) (2020) 3136.
- [9] W. Zhu, E.E. Rezaei, H. Nouri, T. Yang, B. Li, H. Gong, Z. Sun, Quick detection of field-scale soil comprehensive attributes via the integration of UAV and sentinel-2B remote sensing data, *Remote Sens.* 13 (22) (2021) 4716.

## 5.4. Otras contribuciones

### 5.4.1. UOPNOA and UOS2 datasets for aerial crop classification [Conjuntos de datos]

- Pedrayes, O. D., & Usamentiaga, R. (2021). *UOPNOA and UOS2 datasets for aerial crop classification*. Zenodo.
- DOI: [10.5281/zenodo.4648002](https://doi.org/10.5281/zenodo.4648002)

### 5.4.2. Dataset for semantic segmentation in NDT with step-heating thermography for CFRP laminates [Conjuntos de datos]

- Pedrayes, O. D., Usamentiaga, R., & Venegas P. (2021). *Dataset for semantic segmentation in NDT with step-heating thermography for CFRP laminates*. Zenodo.
- DOI: [10.5281/zenodo.5426793](https://doi.org/10.5281/zenodo.5426793)

### 5.4.3. Satellite imagery dataset of manure application on pasture fields [Conjuntos de datos]

- Pedrayes, O. D., Usamentiaga, R., & Venegas P. (2022). *Satellite imagery dataset of manure application on pasture fields*. Zenodo.
- DOI: [10.17632/fbvfvf55kp.1](https://doi.org/10.17632/fbvfvf55kp.1)



# Apéndice A

## Índices multispectrales de vegetación

Tabla A.1: Índices de vegetación mas usados en la agricultura de precisión y su orden en las imágenes multispectrales generadas.

#	Abr.	Nombre	Ecuación
13	NDVI	Normalized Difference Vegetation Index [114, 115]	$\frac{B08 - B04}{B08 + B04}$
14	NSNDVI	NIR-SWIR Normalized Difference Vegetation Index [111]	$\frac{B11 - B07}{B11 + B07}$
15	SDI	SWIR Difference Index [73]	$B08 - B12$
16	GNDVI	Green Normalized Difference Vegetation Index [115, 141]	$\frac{B08 - B03}{B08 + B03}$
17	SAVI	Soil Adjusted Vegetation Index [73, 111, 115]	$\frac{B08 - B04}{B08 + B04 + 0.428} * 1.428$

Apéndice A Índices multiespectrales de vegetación

---

18	OSAVI	Optimized Soil Adjusted Vegetation Index [8, 40, 115, 141]	$(1 + 0.16) * \frac{B08 - B04}{B08 + B04 + 0.16}$
19	EOMI1	Exogenous Organic Matter Index 1 [27]	$\frac{B11 - B8A}{B11 + B8A}$
20	EOMI2	Exogenous Organic Matter Index 2 [27]	$\frac{B12 - B04}{B12 + B04}$
21	EOMI3	Exogenous Organic Matter Index 3 [27]	$\frac{(B11 - B8A) + (B12 + B04)}{B11 + B8A + B12 + B04}$
22	EOMI4	Exogenous Organic Matter Index 4 [27]	$\frac{B11 - B04}{B11 + B04}$
23	BNR2	Normalized Burn Ratio 2 [27]	$\frac{B11 - B12}{B11 + B12}$
24	RVI	Ratio Vegetation Index [114, 115]	$\frac{B08}{B04}$
25	DVI	Difference Vegetation Index [114, 115]	$B08 - B04$
26	RENDVI1	Red Edge Normalized Difference Vegetation Index [115]	$\frac{B05 - B04}{B05 + B04}$
27	RENDVI2	Red Edge Normalized Difference Vegetation Index [115]	$\frac{B06 - B04}{B06 + B04}$



---

28	RENDVI3	Red Edge Normalized Difference Vegetation Index [115]	$\frac{B07 - B04}{B07 + B04}$
<hr/>			
29	CI1	Chlorophyll Index [111]	$\frac{B08}{B05} - 1$
<hr/>			
30	CI2	Chlorophyll Index [111]	$\frac{B08}{B06} - 1$
<hr/>			
31	CI3	Chlorophyll Index [111]	$\frac{B08}{B07} - 1$
<hr/>			
32	NDRE	Normalized Difference Red Edge [115, 141]	$\frac{B08 - B05}{B08 + B05}$
<hr/>			
33	MCARI	Modified Chlorophyll Absorption in Reflectance Index [40, 141]	$((B05 - B04) - 0.2 * (B05 - B03)) * \frac{B05}{B04}$
<hr/>			
34	MCARI1	Modified Chlorophyll Absorption in Reflectance Index 1 [111]	$1.2 * (2.5 * (B08 - B04) - 1.3 * (B08 - B03))$
<hr/>			
35	MCARI2	Modified Chlorophyll Absorption in Reflectance Index 2 [8]	$1.5 * \frac{2.5 * (B08 - B04) - 1.3 * (B08 - B03)}{\sqrt{(2 * B08 + 1)^2 - (6 * B08 - 5 * \sqrt{B04}) - 0.5}}$
<hr/>			
36	MTVI1	Modified Triangular Vegetation Index 1 [8]	$1.2 * (1.2 * (B08 - B03) - 2.5 * (B04 - B03))$
<hr/>			
37	MTVI2	Modified Triangular Vegetation Index 2 [8]	$1.5 * \frac{1.2 * (B08 - B03) - 2.5 * (B08 - B03)}{\sqrt{(2 * B08 + 1)^2 - (6 * B08 - 5 * \sqrt{B04}) - 0.5}}$

Apéndice A Índices multiespectrales de vegetación

---

38	EVI	Enhanced Vegetation Index [111]	$\frac{2.5 * (B08 - B04)}{(B08 + 6 * B04 - 7.5 * B02) + 1}$
39	AVI	Advanced Vegetation Index [111]	$(B08 * (1 - B04) * (B08 - B04))^{1/3}$
40	GCI	Green Coverage Index [111]	$\frac{B09}{B03} - 1$
41	BSI	Bare Soil Index [111]	$B11 + B04 + \frac{B08 + B02}{B11 + B04} + B08 + B02$
42	NBRI	Normalized Burned Ratio Index [111]	$\frac{B08 - B12}{B08 + B12}$
43	NDRE1	Normalized Difference Red Edge [115, 141]	$\frac{B08 - B05}{B08 + B05}$
44	NDRE2	Normalized Difference Red Edge [115, 141]	$\frac{B08 - B06}{B08 + B06}$
45	NDRE3	Normalized Difference Red Edge [115, 141]	$\frac{B08 - B07}{B08 + B07}$
46	MSAVI	Modified Soil Adjusted Vegetation Index [73, 115]	$\frac{(2.0 * B08 + 1 - \sqrt{(2.0 * B08 + 1.0)^2 - 8 * (B08 - B04)})}{2}$
47	WDRVI	Wide Dynamic Range Vegetation Index [115]	$\frac{0.1 * B08 - B04}{0.1 * B08 + B04}$

---

48	ARVI1	Atmospherically Resistant Vegetation Index 1 [111, 115]	$\frac{B8A - B04 - 0.069 * (B04 - B02)}{B8A + B04 - 0.069 * (B04 - B02)}$
49	ARVI2	Atmospherically Resistant Vegetation Index 2 [111, 115]	$-0.18 + 1.17 * \frac{B8-B4}{B8+B4}$
50	TSAVI	Transformed Soil Adjusted Vegetation Index [115]	$\frac{(0.421 * (B08 - 0.421 * B04 - 0.824))}{(B04 + 0.421 * (B08 - 0.824) + 0.114 * (1 + 0.421)^2)}$
51	CARI1	Chlorophyll Absorption Ratio Index 1 [111]	$\frac{B05}{B04} * \frac{\left  \frac{(B05 - B03)}{150} * 670.0 + B04 + (B03 - \left( \frac{(B05 - B03)}{150} * 550 \right)) \right }{\sqrt{(B05 - B03) / 150^2 + 1}}$
52	CARI2	Chlorophyll Absorption Ratio Index 2 [111]	$\frac{ (B05 - B03) / 150 * B04 + B04 + B03 - 0.496 * B03 }{\sqrt{(0.496^2 + 1)} * (B05 / B04)}$
53	CVI	Chlorophyll Vegetation Index [111]	$\frac{B08 * B04}{B03^2}$
54	EVI1	Enhanced Vegetation Index 1 [111]	$\frac{2.5 * (B08 - B04)}{(B08 + 6 * B04 - 7.5 * B02) + 1}$
55	EVI2	Enhanced Vegetation Index 2 [111]	$2.4 * \frac{B08 - B04}{B08 + B04 + 1}$
56	EVI3	Enhanced Vegetation Index 3 [111]	$2.5 * \frac{B08 - B04}{B08 + 2.4 * B04 + 1}$

Apéndice A Índices multiespectrales de vegetación

---

57	SCI	Soil Composition Index [111]	$\frac{B_{11} - B_{08}}{B_{11} + B_{08}}$
58	GRNDVI	Green-Red Normalized Difference Vegetation Index [111]	$\frac{B_{08} - (B_{03} + B_{04})}{B_{08} + (B_{03} + B_{04})}$
59	GBNDVI	Green-Blue Normalized Difference Vegetation Index [111]	$\frac{B_{08} - (B_{03} + B_{02})}{B_{08} + (B_{03} + B_{02})}$
60	GLI	Green Leaf Index [111]	$\frac{2 * B_{03} - B_{04} - B_{02}}{2 * B_{03} + B_{04} + B_{02}}$
61	ATSAVI	Adjusted Transformed Soil-Adjusted Vegetation Index [111]	$\frac{1.22 * (B_{08} - 1.22 * B_{04} - 0.03)}{1.22 * B_{08} + B_{04} - 1.22 * 0.03 + 0.08 * (1 + 1.22^2)}$
62	ALTERATION	Alteration Index [111]	$\frac{B_{11}}{B_{12}}$
63	CTVI	Corrected Transformed Vegetation Index [111]	$\frac{((B_{04} - B_{03}) / (B_{04} + B_{03})) + 0.5}{\left  \frac{B_{04} - B_{03}}{B_{04} + B_{03}} \right  + 0.5 * \sqrt{\left  \frac{B_{04} - B_{03}}{B_{04} + B_{03}} + 0.5 \right }}$

# Bibliografía

- [1] Hosameldin Ahmed and Asoke K Nandi. *Decision trees and random forests*. Wiley-IEEE Press, 2019.
- [2] Craig D Allen, Alison K Macalady, Haroun Chenchouni, Dominique Ba-chelet, Nate McDowell, Michel Vennetier, Thomas Kitzberger, Andreas Rigling, David D Breshears, EH Ted Hogg, et al. A global overview of drought and heat-induced tree mortality reveals emerging climate change risks for forests. *Forest ecology and management*, 259(4):660–684, 2010.
- [3] Md Zahangir Alom, Tarek M Taha, Christopher Yakopcic, Stefan West-berg, Paheding Sidike, Mst Shamima Nasrin, Brian C Van Esesn, Abdul A S Awwal, and Vijayan K Asari. The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint ar-Xiv:1803.01164*, 2018.
- [4] José Manuel Amigo and Carolina Santos. Preprocessing of hyperspectral and multispectral images. In *Data handling in science and technology*, volume 32, pages 37–53. Elsevier, 2019.
- [5] EOS DATA ANALYTICS. El satélite landsat 8: Imágenes, descrip-ción y características. Available online: <https://eos.com/es/find-satellite/landsat-8/>. (accedido el 13-02-2023).
- [6] EOS DATA ANALYTICS. Sentinel-2: Satellite imagery, overview, and characteristics. Available online: <https://eos.com/sentinel-2/>. (acce-dido el 13-02-2023).
- [7] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, dec 2017.
- [8] Nikrooz Bagheri, Hojjat Ahmadi, Seyed Kazem Alavipanah, and Mah-moud Omid. Multispectral remote sensing for site-specific nitrogen fertil-izer management. *Pesquisa Agropecuária Brasileira*, 48:1394–1401, 2013.
- [9] Daniel Balageas, Bastien Chapuis, Geoffrey Deban, and Françoise Passilly. Improvement of the detection of defects by pulse thermography thanks to

- the tsr approach in the case of a smart composite repair patch. *Quantitative InfraRed Thermography Journal*, 7(2):167–187, 2010.
- [10] Daniel L Balageas, Jean-Michel Roche, François-Henri Leroy, Wei-Min Liu, and Alexander M Gorbach. The thermographic signal reconstruction method: a powerful tool for the enhancement of transient thermographic images. *Biocybernetics and biomedical engineering*, 35(1):1–9, 2015.
- [11] László Bertalan, Imre Holb, Angelika Pataki, Gergely Szabó, Annamária Kupásné Szalóki, and Szilárd Szabó. Uav-based multispectral and thermal cameras to predict soil water content—a machine learning approach. *Computers and Electronics in Agriculture*, 200:107262, 2022.
- [12] Massimo Bertolini, Davide Mezzogori, Mattia Neroni, and Francesco Zamori. Machine learning for industrial applications: A comprehensive literature review. *Expert Systems with Applications*, 175:114820, 2021.
- [13] Jerzy Bodzenta, A Kazmierczak, and Tadeusz Kruczek. Analysis of thermograms based on fft algorithm. In *Journal de Physique IV (Proceedings)*, volume 129, pages 201–205. EDP sciences, 2005.
- [14] L. Bragagnolo, L.R. Rezende, R.V. da Silva, and J.M.V. Grzybowski. Convolutional neural networks applied to semantic segmentation of landslide scars. *CATENA*, 201:105189, 2021.
- [15] Jessica Briffa, Emmanuel Sinagra, and Renald Blundell. Heavy metal pollution in the environment and their toxicological effects on humans. *Helvion*, 6(9):e04691, 2020.
- [16] E Oran Brigham. *The fast Fourier transform and its applications*. Prentice-Hall, Inc., 1988.
- [17] Daniel Brugger and Wilhelm M Windisch. Environmental responsibilities of livestock feeding using trace mineral supplements. *Animal Nutrition*, 1(3):113–118, 2015.
- [18] Stuart HM Butchart, Matt Walpole, Ben Collen, Arco Van Strien, Jörn PW Scharlemann, Rosamunde EA Almond, Jonathan EM Baillie, Bastian Bomhard, Claire Brown, John Bruno, et al. Global biodiversity: indicators of recent declines. *Science*, 328(5982):1164–1168, 2010.
- [19] Genevieve Carruthers and Gavin Tinning. Where, and how, do monitoring and sustainability indicators fit into environmental management systems? *Australian journal of experimental agriculture*, 43(3):307–323, 2003.

- 
- [20] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation, 2017.
- [21] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.
- [22] Bowen Cheng, Maxwell D. Collins, Yukun Zhu, Ting Liu, Thomas S. Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020.
- [23] Francois Chollet. Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017.
- [24] Nello Cristianini, John Shawe-Taylor, et al. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.
- [25] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019.
- [26] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [27] Maxence Dodin, Hunter D Smith, Florent Levasseur, Dalila Hadjar, Sabine Houot, and Emmanuelle Vaudour. Potential of sentinel-2 satellite images for monitoring green waste compost and manure amendments in temperate cropland. *Remote Sensing*, 13(9):1616, 2021.
- [28] E. M. Dogo, O. J. Afolabi, N. I. Nwulu, B. Twala, and C. O. Aigbavboa. A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks. In *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*. IEEE, dec 2018.

- [29] Piyush Doke, Dhiraj Shrivastava, Chichun Pan, Qinghua Zhou, and Yu-Dong Zhang. Using CNN with bayesian optimization to identify cerebral micro-bleeds. *Machine Vision and Applications*, 31(5), may 2020.
- [30] Stephan Dreiseitl and Lucila Ohno-Machado. Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics*, 35(5-6):352–359, 2002.
- [31] Juan Du. Understanding of object detection based on cnn family and yolo. In *Journal of Physics: Conference Series*, volume 1004, page 012029. IOP Publishing, 2018.
- [32] Ke Du, Mark J Rood, Byung J Kim, Michael R Kemme, Bill Franek, and Kevin Mattison. Quantification of plume opacity by digital photography. *Environmental science & technology*, 41(3):928–935, 2007.
- [33] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.
- [34] Omar Elharrouss, Somaya Al-Maadeed, Nandhini Subramanian, Najmath Ottakath, Noor Almaadeed, and Yassine Himeur. Panoptic segmentation: a review. *arXiv preprint arXiv:2111.10250*, 2021.
- [35] Fatma Fahmy. Pollution erodes fish stocks and livelihoods in egyptian lake. <https://www.reuters.com/world/africa/pollution-erodes-fish-stocks-livelihoods-egyptian-lake-2022-09-01/>, 2022. Accessed: 2022-09-07.
- [36] Qiang Fang and Xavier Maldague. A method of defect depth estimation for simulated infrared thermography data with deep learning. *Applied Sciences*, 10(19):6819, 2020.
- [37] Eduardo Fernandez-Moral, Renato Martins, Denis Wolf, and Patrick Rives. A new metric for evaluating semantic segmentation: leveraging global and contour accuracy. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1051–1056. IEEE, 2018.
- [38] Food and Agriculture Organization of the United Nations. Livestock and environment statistics: manure and greenhouse gas emissions. global, regional and country trends 1990–2018. *FAOSTAT analytical briefs*, 14, 2020.
- [39] International Organization for Standardization. *ISO 20473:2007-04: Optics and photonics, spectral bands*. Beuth, 2007.



- 
- [40] Yuanyuan Fu, Guijun Yang, Ruiliang Pu, Zhenhai Li, Heli Li, Xingang Xu, Xiaoyu Song, Xiaodong Yang, and Chunjiang Zhao. An overview of crop nitrogen status assessment using hyperspectral remote sensing: Current status and perspectives. *European Journal of Agronomy*, 124:126241, 2021.
- [41] S Gholizadeh. A review of non-destructive testing methods of composite materials. *Procedia Structural Integrity*, 1:50–57, 2016.
- [42] Alison Gillespie. Pnoaa forecasts summer 'dead zone' of nearly 5.4k square miles in gulf of mexico. <https://www.noaa.gov/news-release/noaa-forecasts-summer-dead-zone-of-nearly-54k-square-miles-in-gulf-of-mexico>, 2022. Accessed: 2022-09-07.
- [43] YanFeng Gu, XuDong Jin, RunZi Xiang, QingWang Wang, Chen Wang, and ShengXiong Yang. Uav-based integrated multispectral-lidar imaging system and data processing. *Science China Technological Sciences*, 63(7):1293–1301, 2020.
- [44] Dayan Guan, Yanpeng Cao, Jiangxin Yang, Yanlong Cao, and Michael Ying Yang. Fusion of multispectral data through illumination-aware deep neural networks for pedestrian detection. *Information Fusion*, 50:148–157, oct 2019.
- [45] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. Knn model-based approach in classification. In *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*, pages 986–996. Springer, 2003.
- [46] Yanming Guo, Yu Liu, Theodoros Georgiou, and Michael S Lew. A review of semantic segmentation using deep neural networks. *International journal of multimedia information retrieval*, 7:87–93, 2018.
- [47] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1):389–422, 2002.
- [48] Abdul Mueed Hafiz and Ghulam Mohiuddin Bhat. A survey on instance segmentation: state of the art. *International journal of multimedia information retrieval*, 9(3):171–189, 2020.
- [49] Nathan A. Hagen and Michael W. Kudenov. Review of snapshot spectral imaging technologies. *Optical Engineering*, 52(9):090901, 2013.

- [50] Basna Mohammed Salih Hasan and Adnan Mohsin Abdulazeez. A review of principal component analysis algorithm for dimensionality reduction. *Journal of Soft Computing and Data Mining*, 2(1):20–30, 2021.
- [51] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2017.
- [52] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016.
- [53] Xin He, Kaiyong Zhao, and Xiaowen Chu. AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622, jan 2021.
- [54] Yunze He, Baoyuan Deng, Hongjin Wang, Liang Cheng, Ke Zhou, Siyuan Cai, and Francesco Ciampa. Infrared machine vision and infrared thermography with deep learning: a review. *Infrared Physics & Technology*, page 103754, 2021.
- [55] Petr Hurtik, Vojtech Molek, Jan Hula, Marek Vajgl, Pavel Vlasanek, and Tomas Nejezchleba. Poly-yolo: higher speed, more precise detection and instance segmentation for yolov3. *Neural Computing and Applications*, 34(10):8275–8290, 2022.
- [56] Bernard Kamsu-Foguem. Knowledge-based support in non-destructive testing for health monitoring of aircraft structures. *Advanced engineering informatics*, 26(4):859–869, 2012.
- [57] Yu Kang, Zerui Li, Yunbo Zhao, Jiahu Qin, and Weiguo Song. A novel location strategy for minimizing monitors in vehicle emission remote sensing system. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(4):500–510, apr 2018.
- [58] Paramjit Kaur, Kewal Krishan, Suresh K Sharma, and Tanuj Kanchan. Facial-recognition algorithms: A literature review. *Medicine, Science and the Law*, 60(2):131–139, 2020.
- [59] Shihab Hamad Khaleefah, Salama A Mostafa, Aida Mustapha, and Mohammad Faizul Nasrudin. Review of local binary pattern operators in image feature extraction. *Indonesian Journal of Electrical Engineering and Computer Science*, 19(1):23–31, 2020.
- [60] Madhu Khanna and David Zilberman. Incentives, precision technology and environmental protection. *Ecological Economics*, 23(1):25–43, 1997.

- 
- [61] Zolo Kiala, Onesimo Mutanga, John Odindi, and Kabir Peerbhay. Feature selection on sentinel-2 multispectral imagery for mapping a landscape infested by parthenium weed. *Remote Sensing*, 11(16):1892, 2019.
- [62] Byung J Kim, Mark J Rood, and Ke Du. Digital optical method (dom<sup>TM</sup>) and system for determining opacity, February 24 2009. US Patent 7,495,767.
- [63] David Kirchman. Dead zones: growing areas of aquatic hypoxia are threatening our oceans and rivers. <https://blog.oup.com/2021/02/dead-zones-growing-areas-of-aquatic-hypoxia-are-threatening-our-oceans-and-rivers/>, 2022. Accessed: 2022-09-07.
- [64] Peter JA Kleinman, Sheri Spiegel, Jian Liu, Mike Holly, Clint Church, and John Ramirez-Avila. Managing animal manure to minimize phosphorus losses from land to water. *Animal manure: Production, characteristics, environmental concerns, and management*, 67:201–228, 2020.
- [65] Miron B Kursa and Witold R Rudnicki. Feature selection with the boruta package. *Journal of statistical software*, 36:1–13, 2010.
- [66] Nataliia Kussul, Mykola Lavreniuk, Sergii Skakun, and Andrii Shelestov. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5):778–782, may 2017.
- [67] Thibault Laconde. Fugitive emissions: a blind spot in the fight against climate change. fugitives emissions-sector profile. *INIS*, 51, 2018.
- [68] Andrea S. Laliberte, Mark A. Goforth, Caitriana M. Steele, and Albert Rango. Multispectral remote sensing from unmanned aircraft: Image processing workflows and applications for rangeland environments. *Remote Sensing*, 3(11):2529–2551, nov 2011.
- [69] Petro Liashchynskiy and Pavlo Liashchynskiy. Grid search, random search, genetic algorithm: A big comparison for NAS. *CoRR*, abs/1912.06059, 2019.
- [70] Jian Liu, Peter JA Kleinman, Helena Aronsson, Don Flaten, Richard W McDowell, Marianne Bechmann, Douglas B Beegle, Timothy P Robinson, Ray B Bryant, Hongbin Liu, et al. A review of regulations and guidelines related to winter manure application. *Ambio*, 47(6):657–670, 2018.
- [71] Yang Lu. Artificial intelligence: a survey on evolution, models, applications and future trends. *Journal of Management Analytics*, 6(1):1–29, 2019.

- [72] Qin Luo, Bin Gao, Wai Lok Woo, and Yang Yang. Temporal and spatial deep learning network for infrared thermal defect detection. *NDT & E International*, 108:102164, 2019.
- [73] Qiang Ma, Wantai Yu, and Hua Zhou. The relationship between soil nutrient properties and remote sensing indices in the phaeozem region of northeast china. In *2010 Second International Conference on Computational Intelligence and Natural Computing*, volume 2, pages 109–112. IEEE, 2010.
- [74] FJ Madruga, C Ibarra-Castanedo, O Conde, JM Lopez-Higuera, and X Maldague. Automatic data processing based on the skewness statistic parameter for subsurface defect detection by active infrared thermography. In *Proc. QIRT*, volume 9, page 6. Citeseer, 2008.
- [75] Francisco J Madruga, Clemente Ibarra-Castanedo, Olga M Conde, Xavier P Maldague, and José M López-Higuera. Enhanced contrast detection of subsurface defects by pulsed infrared thermography based on the fourth order statistic moment, kurtosis. In *Thermosense XXXI*, volume 7299, page 72990U. International Society for Optics and Photonics, 2009.
- [76] Michael J McFarland, Glenn R Palmer, and Arthur C Olivas. Life cycle cost evaluation of the digital opacity compliance system. *Journal of environmental management*, 91(4):927–931, 2010.
- [77] Michael J McFarland, Spencer H Terry, Michael J Calidonna, Daniel A Stone, Paul E Kerch, and Steven L Rasmussen. Measuring visual opacity using digital imaging technology. *Journal of the Air & Waste Management Association*, 54(3):296–306, 2004.
- [78] Ann McNamara, Alan Chalmers, Tom Troscianko, and Iain Gilchrist. Comparing real & synthetic scenes using human judgements of lightness. In *Eurographics Workshop on Rendering Techniques*, pages 207–218. Springer, 2000.
- [79] Bojan Milovanović, Mergim Gaši, and Sanjin Gumbarević. Principal component thermography for defect detection in concrete. *Sensors*, 20(14):3891, 2020.
- [80] Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

- 
- [81] Eizan Miyamoto and Thomas Merryman. Fast calculation of haralick texture features. *Human computer interaction institute, Carnegie Mellon University, Pittsburgh, USA. Japanese restaurant office*, 2005.
- [82] Yujian Mo, Yan Wu, Xinneng Yang, Feilin Liu, and Yujun Liao. Review the state-of-the-art technologies of semantic segmentation based on deep learning. *Neurocomputing*, 493:626–646, 2022.
- [83] Rohit Mohan and Abhinav Valada. EfficientPS: Efficient panoptic segmentation. *International Journal of Computer Vision*, 129(5):1551–1579, feb 2021.
- [84] David Moravec, Jan Komárek, Serafin López-Cuervo Medina, and Iñigo Molina. Effect of atmospheric corrections on ndvi: Intercomparability of landsat 8, sentinel-2, and uav sensors. *Remote Sensing*, 13(18):3550, 2021.
- [85] Sajjad Mozaffari, Omar Y Al-Jarrah, Mehrdad Dianati, Paul Jennings, and Alexandros Mouzakitis. Deep learning-based vehicle behavior prediction for autonomous driving applications: A review. *IEEE Transactions on Intelligent Transportation Systems*, 23(1):33–47, 2020.
- [86] Hafiz Suliman Munawar, Ji Zhang, Hongzhou Li, Deqing Mo, and Liang Chang. Mining multispectral aerial images for automatic detection of strategic bridge locations for disaster relief missions. In *Lecture Notes in Computer Science*, pages 189–200. Springer International Publishing, 2019.
- [87] M Sam Navin and Loganathan Agilandeewari. Multispectral and hyperspectral images based land use/land cover change prediction analysis: an extensive review. *Multimedia Tools and Applications*, 79(39-40):29751–29774, 2020.
- [88] K. Niemi, S. Reuter, L. M. Graham, J. Waskoenig, and T. Gans. Diagnostic based modeling for determining absolute atomic oxygen densities in atmospheric pressure helium-oxygen plasmas. *Applied Physics Letters*, 95(15):151504, oct 2009.
- [89] Phan Thanh Noi and Martin Kappas. Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using sentinel-2 imagery. *Sensors*, 18(2):18, dec 2017.
- [90] Henri J Nussbaumer. The fast fourier transform. In *Fast Fourier Transform and Convolution Algorithms*, pages 80–111. Springer, 1981.

- [91] Dickson L Omucheni, Kenneth A Kaduki, Wallace D Bulimo, and Hudson K Angeyo. Application of principal component analysis to multispectral-multimodal optical image analysis for malaria diagnostics. *Malaria Journal*, 13(1), dec 2014.
- [92] Yukihiro Ozaki. Infrared spectroscopy—mid-infrared, near-infrared, and far-infrared/terahertz spectroscopy. *Analytical Sciences*, 37(9):1193–1212, 2021.
- [93] Sharmila Padmanabhan, Todd C Gaier, Alan B Tanner, Shannon T Brown, Boon H Lim, Steven C Reising, Robert Stachnik, Rudi Bendig, and Richard Cofield. Tempest-d radiometer: Instrument description and prelaunch calibration. *IEEE Transactions on Geoscience and Remote Sensing*, 59(12):10213–10226, 2020.
- [94] Mahesh Pal. Random forest classifier for remote sensing classification. *International journal of remote sensing*, 26(1):217–222, 2005.
- [95] Jang-Sik Park, Jong-Kwan Song, et al. Fcn based gas leakage segmentation and improvement using transfer learning. In *2019 IEEE Student Conference on Electric Machines and Systems (SCEMS 2019)*, pages 1–4. IEEE, 2019.
- [96] William P Pfaff and Jay Stretch. Optical digital environment compliance system, July 22 2003. US Patent 6,597,799.
- [97] Darius Phiri, Matamy Simwanda, Serajis Salekin, Vincent R Nyirenda, Yuji Murayama, and Manjula Ranagalage. Sentinel-2 data for land cover/use mapping: A review. *Remote Sensing*, 12(14):2291, 2020.
- [98] Plan Nacional de Ortofotografía Aérea. Plan nacional de ortofotografía aérea. Available online: <https://pnoa.ign.es/>. (accessed on 26-02-2021).
- [99] S. Platnick, M.D. King, S.A. Ackerman, W.P. Menzel, B.A. Baum, J.C. Riedi, and R.A. Frey. The MODIS cloud products: algorithms and examples from terra. *IEEE Transactions on Geoscience and Remote Sensing*, 41(2):459–473, feb 2003.
- [100] Esa Prakasa et al. Development of imaging based method for plume opacity measurement. In *2017 5th International Conference on Instrumentation, Control, and Automation (ICA)*, pages 212–216. IEEE, 2017.
- [101] Nikolas Rajic. Principal component thermography. Technical report, Defence Science and Technology Organisation Victoria (Australia . . . , 2002.

- 
- [102] Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer graphics and image processing*, 1(3):244–256, 1972.
- [103] KKD Ramesh, G Kiran Kumar, K Swapna, Debabrata Datta, and S Suman Rajest. A review of medical image segmentation algorithms. *EAI Endorsed Transactions on Pervasive Health and Technology*, 7(27):e6–e6, 2021.
- [104] Christopher A Ramezan. Transferability of recursive feature elimination (rfe)-derived feature sets for support vector machine land cover classification. *Remote Sensing*, 14(24):6218, 2022.
- [105] Karen Randolph and Kirk Foster. Visible emissions field manual epa methods 9 and 22. *U.S. Environmental Protection Agency*, 1993.
- [106] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [107] Y Rejani and S Thamarai Selvi. Early detection of breast cancer using svm classifier technique. *arXiv preprint arXiv:0912.2314*, 2009.
- [108] Jiangtao Ren, Sau Dan Lee, Xianlu Chen, Ben Kao, Reynold Cheng, and David Cheung. Naive bayes classification of uncertain data. In *2009 Ninth IEEE international conference on data mining*, pages 944–949. IEEE, 2009.
- [109] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.
- [110] Dioline Sara, Ajay Kumar Mandava, Arun Kumar, Shiny Duela, and Anitha Jude. Hyperspectral and multispectral image fusion techniques for high resolution applications: A review. *Earth Science Informatics*, 14(4):1685–1705, 2021.
- [111] Sentinel-Hub. Sentinel-2 rs indices — sentinel-hub custom scripts. <https://custom-scripts.sentinel-hub.com/custom-scripts/sentinel-2/indexdb/>, 2022. Accessed: 2022-12-06.
- [112] BT Series. Colour gamut conversion from recommendation itu-r bt. 2020 to recommendation itu-r bt. 709. *International Telecommunication Union*, 2017.

- [113] Steven Shepard, James Lhota, Bruce Rubadeux, David Wang, and Taymur Ahmed. Reconstruction and enhancement of active thermographic image sequences. *Optical Engineering - OPT ENG*, 42:1337–1342, 05 2003.
- [114] Lina Shou, Liangliang Jia, Zhenling Cui, Xinping Chen, and Fusuo Zhang. Using high-resolution satellite imaging to evaluate nitrogen status of winter wheat. *Journal of plant nutrition*, 30(10):1669–1680, 2007.
- [115] Rajendra P Sishodia, Ram L Ray, and Sudhir K Singh. Applications of remote sensing in precision agriculture: A review. *Remote Sensing*, 12(19):3136, 2020.
- [116] Vivek Singh Sisodiya and Rohit Agrawal. A comprehensive study of image segmentation techniques. In *Recent Innovations in Mechanical Engineering: Select Proceedings of ICRITDME 2020*, pages 247–255. Springer, 2022.
- [117] Susan Solomon, Martin Manning, Melinda Marquis, Dahe Qin, et al. *Climate change 2007-the physical science basis: Working group I contribution to the fourth assessment report of the IPCC*, volume 4. Cambridge university press, 2007.
- [118] Yin Song, Arkaprabha Konar, Riley Sechrist, Ved Prakash Roy, Rong Duan, Jared Dziurgot, Veronica Policht, Yassel Acosta Matutes, Kevin J Kubarych, and Jennifer P Ogilvie. Multispectral multidimensional spectrometer spanning the ultraviolet to the mid-infrared. *Review of Scientific Instruments*, 90(1):013108, 2019.
- [119] Yifan Sun, Nicolas Bohm Agostini, Shi Dong, and David Kaeli. Summarizing cpu and gpu design trends with product data. *arXiv preprint arXiv:1911.11313*, 2019.
- [120] R. Taghizadeh-Mehrjardi, M. Mahdianpari, F. Mohammadimanesh, T. Behrens, N. Toomanian, T. Scholten, and K. Schmidt. Multi-task convolutional neural networks outperformed random forest for mapping soil particle size fractions in central iran. *Geoderma*, 376:114552, 2020.
- [121] Kenichi Tatsumi, Yosuke Yamashiki, Miguel Angel Canales Torres, and Cayo Leonidas Ramos Taipe. Crop classification of upland fields using random forest of time-series landsat 7 etm+ data. *Computers and Electronics in Agriculture*, 115:171–179, 2015.
- [122] R. L. Thompson, L. Lassaletta, P. K. Patra, C. Wilson, K. C. Wells, A. Gressent, E. N. Koffi, M. P. Chipperfield, W. Winiwarter, E. A. Davidson, H. Tian, and J. G. Canadell. Acceleration of global n2o emissions



- seen from two decades of atmospheric inversion. *Nature Climate Change*, 9(12):993–998, nov 2019.
- [123] Eva Tuba, Nebojša Bačanin, Ivana Strumberger, and Milan Tuba. Convolutional neural networks hyperparameters tuning. In *Artificial Intelligence: Theory and Applications*, pages 65–84. Springer International Publishing, 2021.
- [124] John Tzilivakis, DJ Warner, Andrew Green, and KA Lewis. A broad-scale spatial analysis of the environmental benefits of fertiliser closed periods implemented under the nitrates directive in europe. *Journal of Environmental Management*, 299:113674, 2021.
- [125] Rubén Usamentiaga, Clemente Ibarra-Castanedo, Matthieu Klein, Xavier Maldague, Jeroen Peeters, and Alvaro Sanchez-Beato. Nondestructive evaluation of carbon fiber bicycle frames using infrared thermography. *Sensors*, 17(11):2679, 2017.
- [126] HP Vinutha, B Poornima, and BM Sagar. Detection of outliers using interquartile range technique from intrusion dataset. In *Information and decision sciences*, pages 511–518. Springer, 2018.
- [127] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.
- [128] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. UPSNet: A unified panoptic segmentation network. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2019.
- [129] Wenling Xuan, Zongjian Lin, Xiuwan Chen, and Gang Zhao. Separating manual operation from remote sensing image processing procedure for high performance parallel computing. In *Geoinformatics 2007: Remotely Sensed Data and Information*, volume 6752, pages 192–199. SPIE, 2007.
- [130] Chun-Chieh Yang, Shiv O Prasher, Joann Whalen, and Pradeep K Goel. Pa—precision agriculture: use of hyperspectral imagery for identification of different fertilisation methods with decision-tree technology. *Biosystems Engineering*, 83(3):291–298, 2002.
- [131] Qifan Yang, Huijuan Zhang, Jun Xia, and Xiaoliang Zhang. Evaluation of magnetic resonance image segmentation in brain low-grade gliomas using support vector machine and convolutional neural network. *Quantitative Imaging in Medicine and Surgery*, 11(1):300, 2021.

- [132] Su Ye and Dongmei Chen. An unsupervised urban change detection procedure by using luminance and saturation for multispectral remotely sensed images. *Photogrammetric Engineering & Remote Sensing*, 81(8):637–645, aug 2015.
- [133] Zhiping Ye, Jiaqian Yang, Na Zhong, Xin Tu, Jining Jia, and Jiade Wang. Tackling environmental challenges in pollution controls using artificial intelligence: A review. *Science of the Total Environment*, 699:134279, 2020.
- [134] Kan Hua Yu, Yue Zhang, Danni Li, Carlos Enrique Montenegro-Marin, and Priyan Malarvizhi Kumar. Environmental planning based on reduce, reuse, recycle and recover using artificial intelligence. *Environmental Impact Assessment Review*, 86:106492, 2021.
- [135] Xiaohui Yuan, Jianfang Shi, and Lichuan Gu. A review of deep learning methods for semantic segmentation of remote sensing imagery. *Expert Systems with Applications*, 169:114417, 2021.
- [136] Wangki Yuen, Yichao Gu, Yalin Mao, Sotiria Koloutsou-Vakakis, Mark J Rood, Hyun-Keun Son, Kevin Mattison, Bill Franek, Ke Du, et al. Performance and uncertainty in measuring atmospheric plume opacity using compact and smartphone digital still cameras. *Aerosol and Air Quality Research*, 17(5):1281–1293, 2017.
- [137] Wangki Yuen, Yichao Gu, Yalin Mao, Peter M Kozak, Sotiria Koloutsou-Vakakis, Hyun-Keun Son, Kevin Mattison, Bill Franek, and Mark J Rood. Daytime atmospheric plume opacity measurement using a camcorder. *Environmental Technology & Innovation*, 12:43–54, 2018.
- [138] Baobao Zhang and Allan Dafoe. Artificial intelligence: American attitudes and trends. *Available at SSRN 3312874*, 2019.
- [139] Caiming Zhang and Yang Lu. Study on artificial intelligence: The state of the art and future prospects. *Journal of Industrial Information Integration*, 23:100224, 2021.
- [140] Nan Zhang, Mingjie Chen, Fan Yang, Cancan Yang, Penghui Yang, Yushan Gao, Yue Shang, and Daoli Peng. Forest height mapping using feature selection and machine learning by integrating multi-source satellite data in baoding city, north china. *Remote Sensing*, 14(18):4434, 2022.
- [141] Wanxue Zhu, Ehsan Eyshi Rezaei, Hamideh Nouri, Ting Yang, Binbin Li, Huarui Gong, Yun Lyu, Jinbang Peng, and Zhigang Sun. Quick detection of field-scale soil comprehensive attributes via the integration of uav and sentinel-2b remote sensing data. *Remote Sensing*, 13(22):4716, 2021.