



Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo



Escuela de
Ingeniería
Informática
Universidad de Oviedo

Enriquecimiento y Generación de Modelos de Aprendizaje Automático para Detección de Objetos en Imágenes de Satélite

GRADO EN INGENIERÍA INFORMÁTICA DEL SOFTWARE

TRABAJO DE FIN DE GRADO

AUTOR

Sergio Cimadevilla Torga

TUTOR

Darío Álvarez Gutiérrez

Noviembre 2024



Agradecimientos

A mi familia, por su constante apoyo y por brindarme esta oportunidad.

A mis compañeros de clase, en especial a los cyberpapis, por el viaje compartido y el ánimo a lo largo de estos años.

A mis colegas en Imageryst, cuya colaboración y experiencia han sido de gran ayuda durante este proceso.

Y a mi tutor del TFG, por su guía, ayuda y valiosos consejos.

Capítulo 1 RESUMEN





1.1 RESUMEN

La detección de infraestructuras a partir de imágenes satelitales ha ganado relevancia en los últimos años gracias al avance en el Aprendizaje Profundo. En este estudio, nos enfocamos en la identificación de plantas solares utilizando imágenes del satélite Copernicus Sentinel-2¹ y técnicas de aprendizaje automático.

Para ello, procesamos las imágenes obtenidas a través de una STAC API², centrando el análisis en las bandas RGB y generando máscaras que delimitan las áreas de las plantas solares, utilizando datos vectoriales descargados mediante consultas en Overpass Turbo³. Posteriormente, aplicamos técnicas de aumento de datos para enriquecer el conjunto de imágenes y entrenamos dos redes neuronales, DeepLabV3+⁴ e Inception ResNet V2⁵, con el fin de mejorar la precisión en la detección de estas infraestructuras.

Tras varias iteraciones y ajustes, validamos los resultados obtenidos, comprobando que los modelos desarrollados son capaces de detectar eficazmente las plantas solares en las imágenes.

Este trabajo busca contribuir a la monitorización de infraestructuras de energía renovable, demostrando la efectividad de las técnicas de aprendizaje profundo para la detección de objetos en imágenes satelitales de manera precisa y escalable.

1.2 PALABRAS CLAVE

Palabras Clave: Imágenes satelitales, aprendizaje profundo, detección de objetos, DeepLabV3, Inception ResNet V2, plantas solares, Copernicus Sentinel-2.

¹ Satélite de observación de la Tierra, especializado en la captura de imágenes multispectrales para la monitorización ambiental y análisis de áreas específicas.

² Herramienta de acceso a imágenes satelitales que facilita la búsqueda y recuperación de datos geoespaciales mediante una interfaz estandarizada.

³ Plataforma de consulta para datos geográficos de OpenStreetMap, utilizada para extraer información vectorial como ubicaciones de plantas solares.

⁴ Arquitectura de red neuronal profunda diseñada para la segmentación semántica de imágenes, permitiendo identificar áreas específicas dentro de una imagen.

⁵ Red neuronal avanzada que combina módulos de la arquitectura Inception con redes residuales, optimizando el rendimiento en tareas de clasificación y segmentación.

1.3 ABSTRACT

The detection of infrastructures from satellite images has gained relevance in recent years thanks to advancements in Deep Learning. This study focuses on identifying solar plants using images from the Copernicus Sentinel-2 ⁶satellite and machine learning techniques.

The images were obtained via a STAC API⁷, processed to focus on the RGB bands, and used to generate masks delineating solar plant areas, with vector data sourced through specific queries in Overpass Turbo⁸. *Data augmentation* techniques were applied to enhance the dataset, and two neural networks, DeepLabV3+⁹ and Inception ResNet V2¹⁰, were trained to improve detection accuracy.

After several iterations and adjustments, the results were validated, showing that the developed models effectively detect solar plants in satellite images.

This work aims to contribute to the monitoring of renewable energy infrastructures, demonstrating the effectiveness of deep learning techniques for accurate and scalable object detection in satellite imagery.

1.4 KEYWORDS

Keywords: Satellite images, deep learning, object detection, DeepLabV3, Inception ResNet V2, solar plants, Copernicus Sentinel

⁶ A satellite mission developed by the European Space Agency as part of the Copernicus Programme, designed for Earth observation and monitoring.

⁷ The SpatioTemporal Asset Catalog API, an interface for obtaining satellite images through a catalog that allows spatial and temporal searches for the required images.

⁸ A tool for querying and extracting geospatial data from OpenStreetMap, facilitating the retrieval of geographic information such as the locations of solar plants.

⁹ A neural network architecture based on deep learning used for semantic segmentation in images, specifically for identifying and classifying objects within images.

¹⁰ A convolutional neural network architecture that combines residual connections and inception modules, allowing the extraction of high-quality features from images.

Índice de contenido

Capítulo 1	Resumen	4
1.1	Resumen	5
1.2	Palabras clave	5
1.3	Abstract.....	6
1.4	Keywords	6
Capítulo 2	Introducción	13
2.1	Motivación	14
2.2	Explicación del Trabajo	14
2.2.1	Problema a Solucionar	14
2.2.2	Solución Propuesta	15
2.2.3	Resultados Esperados	16
2.3	Objetivos y enfoque	17
2.4	Organización de la memoria	18
Capítulo 3	Planificación y Gestión del TFG	19
3.1	Planificación del proyecto.....	20
3.1.1	Identificación de interesados.....	20
3.1.2	OBS y PBS.....	22
3.1.3	Planificación inicial. WBS	26
3.2	Riesgos	32
3.3	Bitácora de incidencias	35
3.4	Planificación Final	36
3.5	Presupuestos.....	41
3.5.1	Presupuesto Inicial.....	41
3.5.2	Presupuesto de Costes	42
3.5.3	Resumen del presupuesto de costes	47
3.5.4	Presupuesto de Cliente.....	48
3.6	Presupuesto Final de Costes	49
3.7	Presupuesto Final para el Cliente	50
3.8	Informe de Lecciones Aprendidas.....	51

Capítulo 4	Detección de Plantas Solares en Imágenes Satelitales.....	52
4.1	Estado del Arte en la Segmentación de Imágenes Satelitales y Plantas Solares.....	53
4.2	TÉCNICAS EXISTENTES PARA LA SEGMENTACIÓN DE PLANTAS SOLARES EN IMÁGENES DE SATÉLITE.....	53
4.2.1	Métodos Tradicionales de Visión por Computador para Segmentación de Imágenes.....	54
4.2.2	Métodos Basados en Aprendizaje Profundo	55
4.2.3	Abordaje mediante Visión por Computador e Inteligencia Artificial	56
4.2.4	Métodos para Mejorar la Precisión de los Resultados Obtenidos.....	57
4.3	Tipos de Plantas Solares.....	59
4.4	Bases de Datos Utilizadas	59
4.5	Estudios Realizados.....	60
4.5.1	Detección y Segmentación de Módulos Fotovoltaicos en Imágenes RGB Aéreas	60
4.5.2	Procesamiento Inteligente de Imágenes Satelitales: Métodos Avanzados y Aplicaciones 61	
4.5.3	Identificación de Sistemas Fotovoltaicos Residenciales mediante Imágenes Satelitales de Alta Resolución.....	61
4.5.4	Modelo de Segmentación Semántica en Imágenes Satelitales basado en Redes Neuronales Convolucionales.....	62
4.5.5	Identificación de Sistemas Solares Descentralizados en Imágenes Aéreas mediante Deep Learning	62
4.6	Escasez de Estudios sobre la Detección y Segmentación de Plantas Solares.....	63
4.7	Redes Neuronales Convolucionales para Extracción de Características	64
4.7.1	Arquitectura y Funcionamiento	64
4.7.2	Entrenamiento de CNNs	65
4.7.3	Desafíos y Ventajas en la Clasificación de Imágenes Satelitales.....	65
4.8	Selección de Herramientas	66
4.8.1	Modelos de Segmentación: Inception ResNet V2 y DeepLabV3+.....	66
4.8.2	Proveedor de Imágenes Satelitales: Sentinel-2	67
4.8.3	Infraestructura de Procesamiento y Almacenamiento: Google Cloud y Amazon Web Services 67	
4.8.4	Otras Herramientas y Librerías	68
4.8.5	Conclusión	68
Capítulo 5	Diseño y Desarrollo	69

5.1	Esquema General del Algoritmo Propuesto.....	70
5.2	Obtención de Datos	71
5.2.1	Obtención de Vectoriales con Overpass Turbo	72
5.2.2	Edición de Vectoriales en geojson.io	73
5.2.3	Descarga y Procesamiento de Imágenes Satelitales con la STAC API	74
5.2.4	Georreferenciación, Re-proyección y Creación de Rasters	74
5.2.5	Evaluación de la Utilidad del Índice NDVI	75
5.2.6	Generación de Rasters Georreferenciados.....	75
5.2.7	Resultados de la Obtención de Datos.....	75
5.3	Generación de Máscaras y Preprocesamiento del <i>Dataset</i>	76
5.3.1	Normalización de Imágenes RGB.....	78
5.3.2	Generación y Conversión de Máscaras de Segmentación	80
5.3.3	<i>Random Cropping</i> y Generación de Parches.....	82
5.3.4	<i>Data Augmentation</i> : Aumento del <i>Dataset</i> mediante Transformaciones.....	84
5.3.5	División del <i>Dataset</i> en Conjuntos de Entrenamiento, Validación y Prueba	86
5.3.6	Evaluación del <i>Dataset</i> y Eliminación de Imágenes con Poco Contexto.....	88
5.4	Entrenamiento y Evaluación del Modelo de Segmentación	90
5.4.1	Implementación de InceptionResNetV2 para Segmentación.....	91
5.4.2	Modelo DeepLabV3+ para Segmentación de Plantas Solares	94
5.4.3	Selección de DeepLabV3+ como Modelo Final.....	97
Capítulo 6	Evaluación del Modelo y Resultados	98
6.1	Introducción.....	99
6.2	Evaluación durante el Entrenamiento	100
6.2.1	Análisis de Resultados de Entrenamiento	101
6.3	Evaluación en Conjunto de Datos No Visto.....	102
6.3.1	Ejemplos de Resultados en Imágenes Nuevas.....	102
6.4	Conclusión sobre el Rendimiento del Modelo	103
Capítulo 7	Conclusiones y Trabajo Futuro.....	104
7.1	Conclusiones	105
7.2	Trabajo Futuro	106
Capítulo 8	Manual de Reproducción del Proyecto	107
8.1	Introducción.....	108

8.2	Requisitos del Proyecto	108
8.3	Configuración del Servidor.....	109
8.3.1	Google Compute Engine	109
8.3.2	Google Colab.....	111
8.4	Notas Finales.....	112
Capítulo 9	Bibliografía	113
Capítulo 10	Anexos	116
10.1	Plan de gestión de riesgos.....	117
10.1.1	Metodología de Gestión de Riesgos de Boehm	117

ÍNDICE DE ILUSTRACIONES

ILUSTRACIÓN 1. EJEMPLO: PLANTA SOLAR FOTOVOLTAICA VISTA DESDE SATÉLITE.	15
ILUSTRACIÓN 2. EJEMPLO: DETECCIÓN DE PLANTA SOLAR EN IMAGEN DE SATÉLITE USANDO MODELO PROPIO DE SEGMENTACIÓN. EN PRIMER LUGAR IMAGEN RGB DE COPERNICUS, EN EL BLOQUE CENTRAL, EN BLANCO, LA SUPERFICIE DE LA PLANTA, SACADA DE OVERPASS-TURBO Y FINALMENTE EN ÚLTIMO TÉRMINO, EN BLANCO, LA DETECCIÓN DEL MODELO DE LA SUPERFICIE DE LA PLANTA.	16
ILUSTRACIÓN 3. GANTT 1.	31
ILUSTRACIÓN 4. GANTT 2.	31
ILUSTRACIÓN 5. GANTT 3.	31
ILUSTRACIÓN 6. GANTT 4.	31
ILUSTRACIÓN 7. SEGMENTACIÓN BASADA EN BORDES.	54
ILUSTRACIÓN 8. ARQUITECTURA CNN.	55
ILUSTRACIÓN 9. MATRIZ DE CONFUSIÓN BINARIA.	58
ILUSTRACIÓN 10. DETECCIÓN DE PANELES SOLARES.	60
ILUSTRACIÓN 11. DETECCIÓN DE PANELES SOLARES EN EDIFICIOS. [4].....	61
ILUSTRACIÓN 12. CLASIFICACIÓN DE TIPOS DE TERRENO.....	62
ILUSTRACIÓN 13. DETECCIÓN DE PANELES EN EDIFICIOS 2.	62
ILUSTRACIÓN 14. EJEMPLO DE USO DE OVERPASS-TURBO.....	72
ILUSTRACIÓN 15. EJEMPLO DE USO DE GEOJSON.IO.	73
ILUSTRACIÓN 16. FORMULA NDVI.....	75
ILUSTRACIÓN 17. IMAGEN EJEMPLO SENTINEL-2.....	78
ILUSTRACIÓN 18. IMAGEN DESPUÉS DE LA NORMALIZACIÓN.	79
ILUSTRACIÓN 19. EJEMPLO: MASCARA BINARIA MONOBANDA.....	80
ILUSTRACIÓN 20. EJEMPLO: RANDOM CROPPING, IMAGEN Y MASCARA.....	83
ILUSTRACIÓN 21. EJEMPLO: IMAGEN CON DATA AUGMENTATION APLICADA.	85
ILUSTRACIÓN 22. SEPARACIÓN DEL DATASET EN 3.	87
ILUSTRACIÓN 23. DISTRIBUCIÓN DE PÍXELES BLANCOS EN LAS MÁSCARAS DEL DATASET.	89
ILUSTRACIÓN 24. EJEMPLO: RED NEURONAL Y NEURONA.	90
ILUSTRACIÓN 25. CAPAS CONVOLUCIONALES 1.	91
ILUSTRACIÓN 26. CAPAS CONVOLUCIONALES 2.	94
ILUSTRACIÓN 27. IMAGEN DE LA GRÁFICA DE PRECISIÓN DEL MODELO DEEPLABV3+.....	100
ILUSTRACIÓN 28. IMAGEN DE LA GRÁFICA DE LA PÉRDIDA DEL MODELO DEEPLABV3+.....	101
ILUSTRACIÓN 29. RESULTADO DEL MODELO CON DATASET NUEVO 1.	102
ILUSTRACIÓN 30. RESULTADO DEL MODELO CON DATASET NUEVO 2.	102
ILUSTRACIÓN 31. RESULTADO DEL MODELO CONTRA DATASET DE VALIDACIÓN.	103
ILUSTRACIÓN 32. METODOLOGÍA BOEHM.	117

Índice de Tablas

TABLA 1. IDENTIFICACIÓN DE INTERESADOS.....	21
TABLA 2. OBS.....	22
TABLA 3. PBS ESTRUCTURA GENERAL.....	23
TABLA 4. PBS PLANIFICACIÓN Y DEFINICIÓN DE OBJETIVOS.....	23
TABLA 5. PBS INVESTIGACIÓN Y SELECCIÓN DE TECNOLOGÍAS.....	24
TABLA 6. PBS DESARROLLO DEL MODELO.....	24
TABLA 7. PBS VALIDACIÓN Y EVALUACIÓN DE RESULTADOS.....	25
TABLA 8. PBS DOCUMENTACIÓN Y REDACCIÓN DEL PROYECTO.....	25
TABLA 9. FASE DE PLANIFICACIÓN Y DEFINICIÓN DE OBJETIVOS.....	26
TABLA 10. FASE DE INVESTIGACIÓN Y SELECCIÓN DE TECNOLOGÍAS.....	27
TABLA 11. DESARROLLO DEL MODELO.....	28
TABLA 12. FASE DE VALIDACIÓN Y EVOLUCIÓN DE LOS RESULTADOS.....	29
TABLA 13. FASE DE DOCUMENTACIÓN Y REDACCIÓN DEL PROYECTO.....	30
TABLA 14. REGISTRO DE RIESGOS.....	34
TABLA 15. BITÁCORA DE INCIDENCIAS.....	35
TABLA 16. FASE DE PLANIFICACIÓN Y DEFINICIÓN DE OBJETIVOS FINAL.....	36
TABLA 17. FASE DE INVESTIGACIÓN Y SELECCIÓN DE TECNOLOGÍAS FINAL.....	37
TABLA 18. DESARROLLO DEL MODELO FINAL.....	38
TABLA 19. FASE DE VALIDACIÓN Y EVOLUCIÓN DE LOS RESULTADOS FINAL.....	39
TABLA 20. FASE DE DOCUMENTACION Y REDACCION DEL PROYECTO FINAL.....	40
TABLA 21. PRECIO POR HORA DE LOS DIFERENTES PUESTOS.....	41
TABLA 22. FASE DE PLANIFICACIÓN Y DEFINICIÓN DE OBJETIVOS.....	42
TABLA 23. FASE DE INVESTIGACIÓN Y SELECCIÓN DE TECNOLOGÍAS.....	43
TABLA 24. FASE DE DESARROLLO DE MODELO.....	44
TABLA 25. FASE DE VALIDACIÓN Y EVALUACIÓN DE RESULTADOS.....	45
TABLA 26. DOCUMENTACIÓN Y REDACCIÓN DEL PROYECTO.....	46
TABLA 27. TOTALES DEL PRESUPUESTO DE COSTES.....	47
TABLA 28. AJUSTES DE PRESUPUESTO.....	48
TABLA 29. PRESUPUESTO CLIENTE.....	48
TABLA 30. DATOS PARA EL CÁLCULO DE PRESUPUESTO FINAL DE CLIENTE.....	50
TABLA 31. PRESUPUESTO FINAL DEL CLIENTE.....	50
TABLA 32. EJEMPLO: GESTIÓN DE RIESGOS.....	119

Capítulo 2 INTRODUCCIÓN





2.1 MOTIVACIÓN

La detección y monitorización de infraestructuras de energía renovable, como las plantas solares, se ha convertido en una tarea crucial en el contexto actual de transición energética y desarrollo sostenible. La identificación de estas infraestructuras en imágenes satelitales presenta múltiples beneficios, como el control y mantenimiento de grandes áreas de plantas solares de forma remota y eficiente, ayudando a optimizar recursos y a reducir costes operativos.

Con los avances recientes en el campo del Aprendizaje Profundo, especialmente en las Redes Neuronales Convolucionales, y en la Visión por Computador, se han abierto nuevas oportunidades para aplicar técnicas de detección de objetos en imágenes satelitales. La precisión y escalabilidad que ofrecen estos métodos permiten identificar estructuras complejas y abarcar áreas extensas, algo que resultaría difícil y costoso con métodos tradicionales de inspección en campo.

Este proyecto explora la viabilidad de estas técnicas para la detección de plantas solares, combinando diversas herramientas y algoritmos de procesamiento de imágenes y aprendizaje profundo. A través de este enfoque, buscamos contribuir a la gestión eficiente de infraestructuras de energía renovable y aportar valor por medio del seguimiento continuo de proyectos a gran escala.

2.2 EXPLICACIÓN DEL TRABAJO

2.2.1 Problema a Solucionar

La detección y seguimiento de plantas solares a partir de imágenes satelitales es una tarea compleja y laboriosa. Actualmente, la identificación de estas infraestructuras se realiza en gran parte de forma manual, lo que implica revisar extensas áreas de imágenes para localizar plantas solares, lo cual no solo consume tiempo y recursos, sino que también conlleva errores humanos y limitaciones en la cobertura. Esto se agrava en estudios a gran escala, donde el acceso a datos precisos y actualizados es fundamental para la monitorización y mantenimiento de las plantas solares.

Además, la variabilidad de las características visuales entre distintas plantas solares (ubicación geográfica, tipos de paneles, condiciones atmosféricas) dificulta la tarea de los analistas. Sin herramientas avanzadas, estas tareas manuales son ineficientes y no logran adaptarse al ritmo y escala de crecimiento del sector de energía renovable, donde la monitorización en tiempo real y la precisión en la detección son esenciales.



Ilustración 1. Ejemplo: planta solar fotovoltaica vista desde satélite.

2.2.2 Solución Propuesta

Este proyecto busca automatizar la detección de plantas solares en imágenes satelitales utilizando modelos avanzados de aprendizaje profundo. Para ello, se desarrolló un sistema basado en redes neuronales, específicamente usando arquitecturas como DeepLabV3+ e Inception ResNet V2, que permite segmentar de forma precisa las áreas correspondientes a plantas solares en imágenes de alta resolución.

La solución está diseñada para ser escalable y eficiente, permitiendo que imágenes satelitales nuevas se procesen y clasifiquen de forma automatizada, sin intervención manual. Esto se logra mediante la aplicación de técnicas de segmentación, donde los modelos, entrenados con grandes volúmenes de datos, aprenden a distinguir las características propias de las plantas solares, como la reflectancia de los paneles. Así, el sistema ofrece una detección rápida y precisa, que reduce significativamente el tiempo y los costos asociados con la monitorización manual.

2.2.3 Resultados Esperados

Como resultado final, el sistema permite obtener imágenes satelitales en las que las áreas de plantas solares están claramente segmentadas y detectadas automáticamente, con una alta precisión. Este avance no solo mejora la eficiencia en la supervisión de infraestructuras energéticas, sino que también proporciona a las empresas de energía renovable una herramienta que les permite tomar decisiones informadas y realizar mantenimientos preventivos de manera más ágil y exacta.

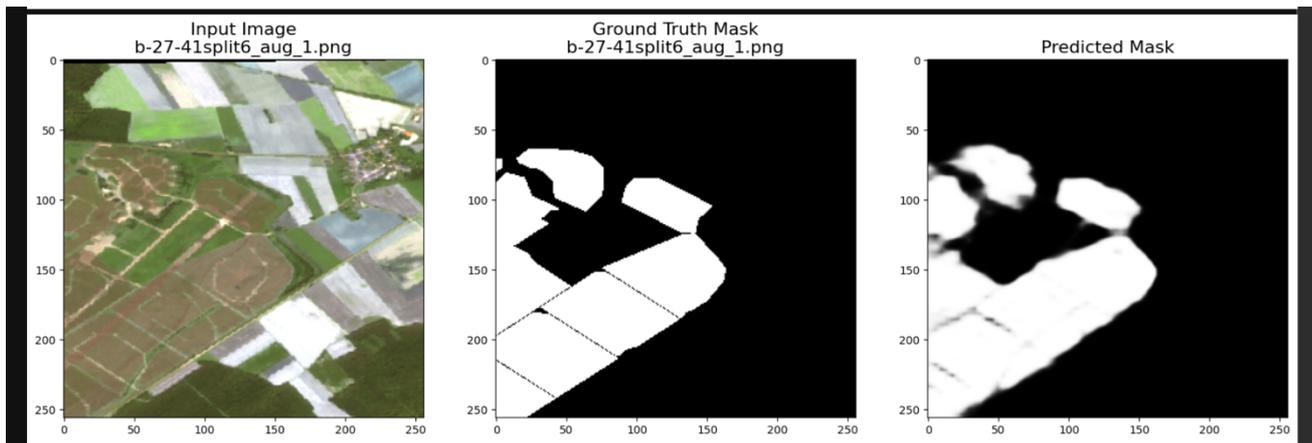


Ilustración 2. Ejemplo: detección de planta solar en imagen de satélite usando modelo propio de segmentación. En primer lugar imagen RGB de Copernicus, en el bloque central, en blanco, la superficie de la planta, sacada de Overpass-turbo y finalmente en último término, en blanco, la detección del modelo de la superficie de la planta.



2.3 OBJETIVOS Y ENFOQUE

Este trabajo tiene como objetivo desarrollar un sistema de detección de plantas solares en imágenes satelitales mediante técnicas avanzadas de aprendizaje profundo. Para ello, el proyecto se estructurará en los siguientes pasos:

- **Revisión y documentación de técnicas y métodos** aplicados en la detección de objetos en imágenes satelitales, con especial atención a los modelos de aprendizaje profundo y al procesamiento de datos geoespaciales.
- **Selección y adquisición de datos:** Uso de una fuente pública de imágenes satelitales, Copernicus Sentinel-2, y obtención de datos vectoriales de plantas solares a través de Overpass Turbo.
- **Implementación de un sistema de detección supervisada** que utilice redes neuronales profundas para la identificación de plantas solares en imágenes satelitales.

El sistema implementado consistirá en:

- **Preprocesamiento de las imágenes y generación de máscaras:** Normalización de las imágenes y creación de máscaras binarizadas que delimiten las áreas de las plantas solares.
- **Entrenamiento de redes neuronales profundas** (DeepLabV3 e Inception ResNet V2) con el *dataset* (conjunto de datos) generado, aplicando técnicas de aumento de datos para mejorar la robustez del modelo.
- **Evaluación y ajuste de los modelos:** Comparación de los resultados obtenidos con cada red neuronal, analizando la precisión y efectividad en la detección de las plantas solares.
- **Validación de resultados y optimización:** Análisis de la precisión del modelo en diferentes configuraciones y selección de los mejores parámetros para su implementación.

Este estudio busca evaluar la viabilidad de las técnicas de aprendizaje profundo en la detección de plantas solares, comprobando la precisión de los modelos en la identificación de áreas específicas y analizando la eficacia del sistema implementado para aplicaciones de monitorización en el sector de las energías renovables.



2.4 ORGANIZACIÓN DE LA MEMORIA

La memoria del proyecto se organiza en los siguientes capítulos:

- **Capítulo 1. ¿Qué es este trabajo?:** Este capítulo incluye el resumen del proyecto en español y en inglés, las palabras clave, en ambos idiomas, proporcionando al lector una introducción general al contenido y propósito del trabajo.
- **Capítulo 2. Introducción:** Se detalla la motivación del proyecto, los objetivos y el enfoque del estudio, y se ofrece una visión general de la estructura de la memoria.
- **Capítulo 3. Planificación del Proyecto:** Este capítulo detalla la planificación del proyecto, la identificación de interesados, el OBS y PBS, el WBS, el cronograma en formato de diagrama de Gantt, el análisis de riesgos y el presupuesto inicial.
- **Capítulo 4. Detección de Plantas Solares en Imágenes Satelitales. Estado del Arte:** Se presenta una introducción al tema de la detección de objetos en imágenes satelitales, revisando las técnicas actuales en el uso de redes neuronales convolucionales y aprendizaje profundo aplicadas a la identificación de plantas solares. También se discuten los *datasets* disponibles, estudios previos relevantes y las herramientas que se van a usar en el proyecto.
- **Capítulo 5. Diseño y Desarrollo:** Se describe el esquema general del sistema propuesto, incluyendo la adquisición de datos satelitales y vectoriales, el preprocesamiento de las imágenes y máscaras, el aumento de datos, y la implementación de redes neuronales (DeepLabV3 e Inception ResNet V2) para la detección de plantas solares.
- **Capítulo 6. Experimentos Realizados y Resultados:** En este capítulo se detallan los experimentos realizados para el entrenamiento y validación de los modelos, así como los resultados obtenidos. Se analiza la precisión de los modelos y se comparan los resultados de cada red neuronal empleada.
- **Capítulo 7. Conclusiones y Trabajo Futuro:** Se resumen los logros del proyecto y se discuten los principales descubrimientos. Además, se plantean futuras mejoras y líneas de investigación adicionales para optimizar la detección de plantas solares en imágenes satelitales.
- **Capítulo 8. Bibliografía:** Se presentan todas las referencias bibliográficas utilizadas en el proyecto, que incluyen artículos científicos, libros y recursos en línea consultados.
- **Capítulo 9. Anexos:** Este capítulo incluye información adicional relevante para el proyecto, como el plan de gestión de riesgos, resultados secundarios, detalles técnicos y otros materiales de apoyo que complementan el contenido principal de la memoria.



Capítulo 3 PLANIFICACIÓN Y GESTIÓN DEL TFG

3.1 PLANIFICACIÓN DEL PROYECTO

En el siguiente apartado se va a detallar la planificación del proyecto, las actividades a realizar, la organización de los responsables, los resultados esperados y el plan de trabajo del proyecto.

Seguidamente se elaborarán los presupuestos y un análisis de riesgos.

3.1.1 Identificación de interesados

Usuario	Perfil	Función
Equipo de Supervisión		
Tutor de TFG	Jefe de Proyecto	Supervisión del desarrollo del proyecto y cumplimiento de los objetivos definidos.
EQUIPO DE DESARROLLO		
Alumno	Investigador	Búsqueda de <i>datasets</i> relevantes y revisión de estudios y tecnologías aplicables.
Alumno	Analista	Análisis de técnicas de procesamiento y segmentación de imágenes satelitales.
Alumno	Arquitecto del Software	Diseño de la arquitectura general para el pipeline de segmentación y análisis de imágenes.
Alumno	Ingeniero de Procesamiento	Diseño y desarrollo de las etapas de preprocesamiento de datos (normalización, recorte, etc.).
Alumno	Programador	Implementación de algoritmos de segmentación y entrenamiento del modelo de aprendizaje profundo.



Alumno	Administrador de Servidores	Configuración y gestión de servidores en la nube para el procesamiento y almacenamiento de datos.
EQUIPO DE PRUEBAS Y VALIDACIÓN		
Alumno	Tester	Realización de pruebas funcionales, de integración y de precisión del modelo de segmentación.
Alumno	Validador de <i>Dataset</i>	Validación de la calidad y precisión del <i>dataset</i> generado para el entrenamiento del modelo.
EQUIPO DE DOCUMENTACIÓN		
Alumno	Documentador Técnico	Redacción de la documentación técnica del proyecto, incluyendo resultados y análisis de precisión

Tabla 1. Identificación de Interesados.

3.1.2 OBS y PBS

A continuación, se van a detallar el OBS y el PBS, de los que posteriormente se obtendrá el WBS.

OBS



Tabla 2. OBS.

PBS

En el PBS se representa en forma de árbol los productos que han de obtenerse para completar el proyecto de manera satisfactoria.

Para facilitar su legibilidad, se ha dividido en varias graficas:

PBS Estructura General



Tabla 3. PBS estructura general.

Fase de Planificación y Definición de Objetivos

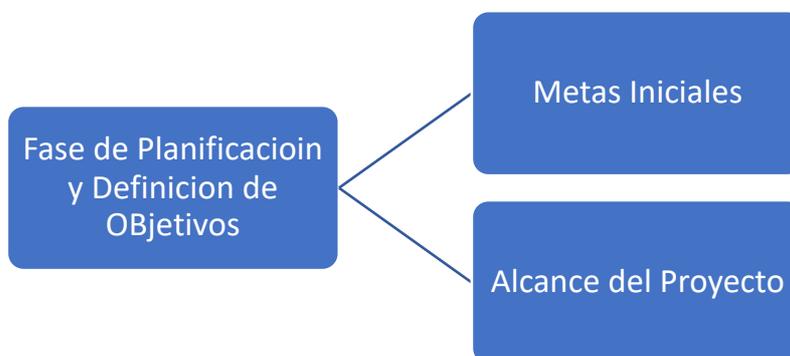


Tabla 4. PBS Planificación y definición de objetivos.

Fase de Investigación y Selección de Tecnologías



Tabla 5. PBS Investigación y selección de tecnologías.

Fase de Desarrollo de Modelo

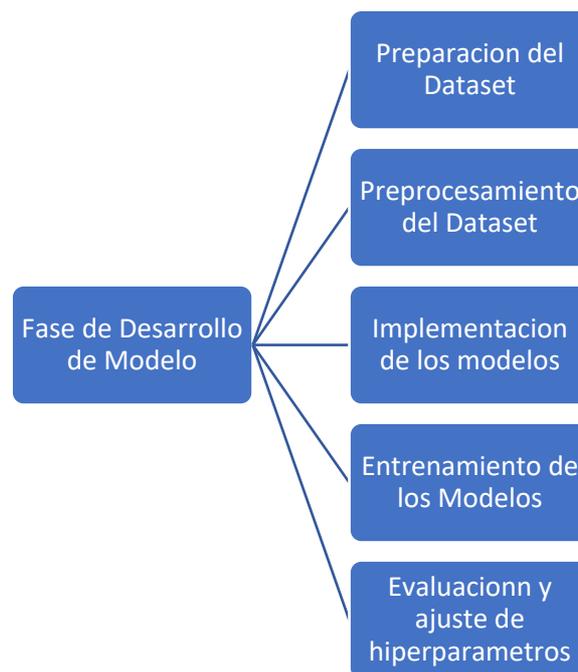


Tabla 6. PBS Desarrollo del modelo.

Fase de Validación y Evaluación de Resultados

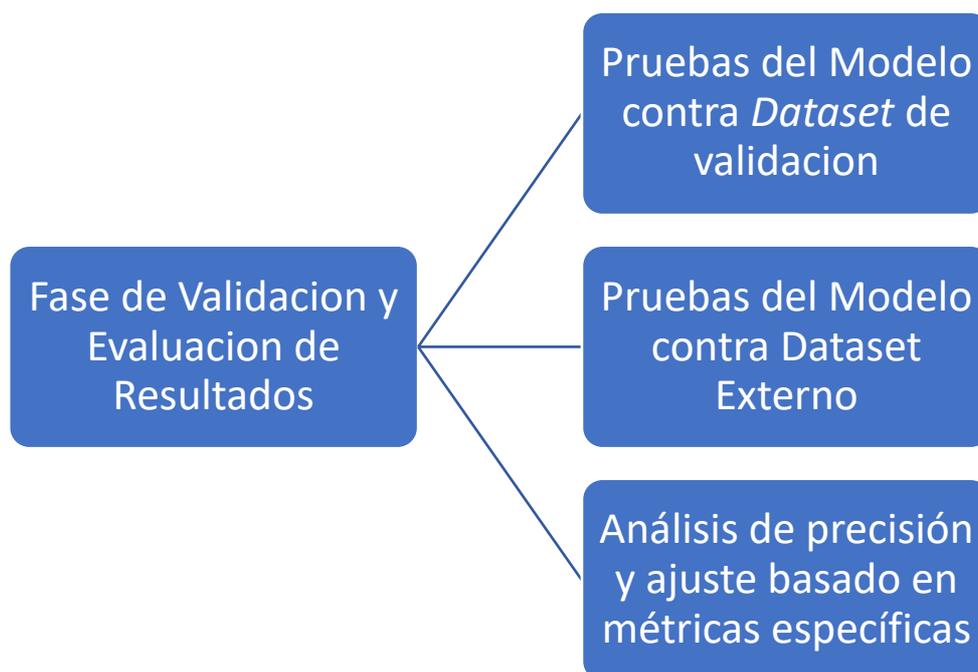


Tabla 7. PBS Validación y evaluación de resultados.

Documentación y Redacción del Proyecto



Tabla 8. PBS Documentación y redacción del proyecto

3.1.3 Planificación inicial. WBS

1. Fase de Planificación y Definición de Objetivos

Número de Esquema	Nombre de Tarea	Duración (Horas)
1	Fase de Planificación y Definición de Objetivos	40
1.1	Definición de objetivos del proyecto	20
1.1.1	Identificación de necesidades y requisitos del proyecto	10
1.1.2	Definición del alcance del proyecto y metas específicas	10
1.2	Planificación inicial	20
1.2.1	Establecimiento de cronograma inicial y principales hitos	10
1.2.2	Asignación de recursos y roles clave	10

Tabla 9. Fase de planificación y definición de objetivos.



2. Fase de Investigación y Selección de Tecnologías

Número de Esquema	Nombre de Tarea	Duración (Horas)
2	Fase de Investigación y Selección de Tecnologías	80
2.1	Revisión del estado del arte	40
2.1.1	Revisión bibliográfica sobre segmentación de imágenes	20
2.1.2	Identificación de técnicas avanzadas	20
2.2	Selección de metodologías y herramientas	40
2.2.1	Evaluación y selección de frameworks de deep learning	20
2.2.2	Selección de arquitecturas de red adecuadas	20

Tabla 10. Fase de Investigación y Selección de Tecnologías.





3. Fase de Desarrollo de Modelo

Número de Esquema	Nombre de Tarea	Duración (Horas)
3	Fase de Desarrollo de Modelo	160
3.1	Preparación y preprocesamiento del <i>dataset</i>	60
3.1.1	Recopilación de imágenes satelitales y generación de máscaras	30
3.1.2	Normalización y división del <i>dataset</i>	30
3.2	Implementación y entrenamiento del modelo	60
3.2.1	Configuración de modelos y definición de arquitecturas	30
3.2.2	Entrenamiento inicial del modelo	30
3.3	Evaluación y ajuste de hiperparámetros	40
3.3.1	Evaluación de métricas de rendimiento	20
3.3.2	Optimización iterativa de hiperparámetros	20

Tabla 11. Desarrollo del modelo.



4. Fase de Validación y Evaluación de Resultados

Número de Esquema	Nombre de Tarea	Duración (Horas)
4	Fase de Validación y Evaluación de Resultados	120
4.1	Validación con <i>datasets</i> internos	40
4.2	Validación con <i>datasets</i> externos	40
4.2.1	Pruebas de generalización con datos nuevos	20
4.2.2	Ajustes y correcciones según resultados	20
4.3	Análisis de precisión y métricas	40
4.3.1	Cálculo de métricas clave y análisis cualitativo	20
4.3.2	Comparación de resultados entre modelos	20

Tabla 12. Fase de validación y evolución de los resultados.





5. Documentación y Redacción del Proyecto

Número de Esquema	Nombre de Tarea	Duración (Horas)
5	Documentación y Redacción del Proyecto	160
5.1	Redacción de la metodología	60
5.1.1	Documentación detallada de cada fase	30
5.1.2	Explicación de decisiones clave	30
5.2	Análisis y presentación de resultados	60
5.2.1	Presentación gráfica de resultados	30
5.2.2	Discusión de logros y limitaciones	30
5.3	Redacción final y conclusiones	40
5.3.1	Redacción de conclusiones y mejoras futuras	20
5.3.2	Preparación de la versión final del proyecto	20

Tabla 13. Fase de Documentación y Redacción del Proyecto.

Diagrama de Gantt

A continuación, se mostrarán los diagramas de Gantt para cada una de las fases detalladas en el apartado anterior.



Ilustración 3. Gantt 1.



Ilustración 4. Gantt 2.



Ilustración 5. Gantt 3.



Ilustración 6. Gantt 4.



3.2 RIESGOS

Plan de gestión de riesgos

El plan de gestión de riesgos sirve para detallar la manera controlar y actuar contra los riesgos. Este plan aparecerá detallado en la sección riesgos, dónde se explicará cómo gestionar cada uno de los riesgos identificados.

Identificación de riesgos

- **Riesgo de disponibilidad de datos de alta calidad:** Escasez de imágenes satelitales con la resolución y frecuencia necesarias para el proyecto.
- **Riesgo de selección inadecuada de arquitectura de modelo:** Elección de una arquitectura que no sea adecuada para el contexto de segmentación de plantas solares.
- **Riesgo de sobreajuste del modelo (overfitting):** Riesgo de que el modelo funcione bien en el *dataset* de entrenamiento, pero no generalice bien con datos nuevos.
- **Riesgo de tiempos de entrenamiento prolongados:** Que el entrenamiento del modelo demore más de lo planeado debido a limitaciones computacionales.
- **Riesgo de problemas de compatibilidad entre librerías y frameworks:** Posibles conflictos entre versiones de TensorFlow, Keras, y otras dependencias.
- **Riesgo de insuficiente capacidad de almacenamiento:** Almacenamiento insuficiente para gestionar el volumen de datos generados durante el preprocesamiento y entrenamiento.
- **Riesgo de interpretabilidad del modelo:** Dificultad para explicar el comportamiento del modelo, lo cual puede complicar la validación y presentación de resultados.
- **Riesgo de sesgo en el *dataset*:** Presencia de sesgos en los datos de entrenamiento que dificulten la generalización del modelo.
- **Riesgo de cambios en los requisitos del proyecto:** Modificación de los objetivos del proyecto que requiera ajustar el alcance de este.
- **Riesgo de falta de tiempo para documentación completa:** Que el tiempo asignado para documentar cada fase no sea suficiente.
- **Riesgo de indisponibilidad de recursos de hardware:** Falta de acceso a servidores o GPU de alto rendimiento.
- **Riesgo de fallos en el software o pérdida de datos:** Posible pérdida de datos debido a fallos de hardware o software.
- **Riesgo de problemas de licencia o permisos en el uso de herramientas:** Limitaciones en el uso de ciertos *datasets* o librerías.
- **Riesgo de incompatibilidad en formato de datos:** Problemas de compatibilidad entre formatos de imágenes y librerías utilizadas.
- **Riesgo de insuficiente validación con datos externos:** Falta de acceso a *datasets* externos para una evaluación robusta.



Registro de riesgos

ID	Riesgo	Probabilidad	Impacto	Mitigación	Responsable
1	Disponibilidad de datos de alta calidad	Media	Alta	Usar fuentes alternativas de datos y considerar <i>data augmentation</i> .	Investigador
2	Selección inadecuada de arquitectura	Media	Media	Revisar modelos previos y realizar pruebas preliminares de varias arquitecturas.	Analista
3	Sobreajuste del modelo	Alta	Media	Implementar técnicas de regularización y aumentar el tamaño del <i>dataset</i> si es posible.	Programador
4	Tiempos de entrenamiento prolongados	Alta	Alta	Ajustar batch sizes y realizar optimización de hiperparámetros para mejorar el rendimiento.	Programador
5	Problemas de compatibilidad entre librerías	Media	Alta	Utilizar un entorno virtual específico para el proyecto y controlar versiones.	Administrador de Servidores
6	Insuficiente capacidad de almacenamiento	Baja	Media	Evaluar requisitos de almacenamiento y usar soluciones en la nube si es necesario.	Administrador de Servidores
7	Interpretabilidad del modelo	Media	Baja	Seleccionar métricas explicativas y agregar visualización de capas.	Analista



8	Sesgo en el <i>dataset</i>	Media	Alta	Ampliar y diversificar el <i>dataset</i> de entrenamiento.	Investigador
9	Cambios en los requisitos del proyecto	Media	Media	Mantener reuniones de actualización periódicas con stakeholders.	Jefe de proyecto
10	Falta de tiempo para documentación completa	Alta	Alta	Asignar tiempo extra al cronograma de documentación.	Documentador técnico
11	Indisponibilidad de recursos de hardware	Alta	Media	Reservar acceso a recursos con anticipación y explorar servicios en la nube como alternativa.	Administrador de Servidores
12	Fallos en el software o pérdida de datos	Media	Alta	Realizar copias de seguridad regulares y utilizar control de versiones.	Administrador de Servidores
13	Problemas de licencia o permisos en herramientas	Baja	Alta	Revisar y gestionar licencias con anticipación.	Administrador de Servidores
14	Incompatibilidad en formato de datos	Media	Media	Definir y estandarizar formatos de datos desde el inicio.	Analista
15	Insuficiente validación con datos externos	Alta	Media	Asegurar el acceso a <i>datasets</i> externos y planificar su adquisición temprana.	Validador de <i>Dataset</i>

Tabla 14. Registro de riesgos

3.3 BITÁCORA DE INCIDENCIAS

A lo largo del desarrollo del proyecto, surgieron varias incidencias que afectaron tanto la planificación como el presupuesto inicial. Estos contratiempos, en su mayoría relacionados con los riesgos previamente identificados, llevaron a la necesidad de realizar ajustes en el cronograma y en la asignación de recursos. A continuación, se presenta la bitácora de incidencias, detallando cada problema, su impacto y las horas adicionales requeridas, lo cual permitirá una estimación más precisa del presupuesto final.

ID	Tipo de Incidencia	Fecha de Creación	Fecha de Cierre	Descripción de la Incidencia	Efecto en la Planificación
1	Disponibilidad de datos de alta calidad	15/8/24	18/8/24	Escasez de imágenes satelitales adecuadas para el proyecto, lo que obligó a utilizar fuentes alternativas.	8 horas adicionales en la fase de recopilación de datos.
2	Selección inadecuada de arquitectura	25/8/24	27/8/24	Se seleccionó inicialmente una arquitectura de modelo no óptima, requiriendo una re-evaluación de alternativas.	6 horas adicionales en la fase de configuración de modelos y definición de arquitecturas.
3	Sobreajuste del modelo	5/9/24	8/9/24	El modelo presentó sobreajuste en datos de prueba, por lo que se implementaron técnicas de regularización.	10 horas adicionales en la fase de Ajustes y correcciones según resultados.
4	Problemas de compatibilidad entre librerías	1/10/24	3/10/24	Conflictos entre versiones de librerías en el entorno de desarrollo, que requirieron ajustar dependencias.	5 horas adicionales en la fase de configuración de modelos y definición de arquitecturas.
5	Falta de tiempo para documentación completa	28/10/24	30/10/24	El tiempo asignado para documentar los resultados no fue suficiente, requiriendo horas adicionales.	6 horas adicionales en la fase de documentación técnica.

Tabla 15. Bitácora de incidencias.



3.4 PLANIFICACIÓN FINAL

Como se mencionó en el apartado de Bitácora de Incidencias, el proyecto ha sufrido varios ajustes debido a algunos retrasos y problemas imprevistos, derivados de la planificación inicial o de incidencias que surgieron durante su desarrollo. Esta planificación final refleja el tiempo adicional requerido para completar las fases afectadas y ajusta la carga horaria según los incidentes registrados.

Número de Esquema	Nombre de Tarea	Duración (Horas)
1	Fase de Planificación y Definición de Objetivos	40
1.1	Definición de objetivos del proyecto	20
1.1.1	Identificación de necesidades y requisitos del proyecto	10
1.1.2	Definición del alcance del proyecto y metas específicas	10
1.2	Planificación inicial	20
1.2.1	Establecimiento de cronograma inicial y principales hitos	10
1.2.2	Asignación de recursos y roles clave	10

Tabla 16. Fase de planificación y definición de objetivos final.



Número de Esquema	Nombre de Tarea	Duración (Horas)
2	Fase de Investigación y Selección de Tecnologías	80
2.1	Revisión del estado del arte	40
2.1.1	Revisión bibliográfica sobre segmentación de imágenes	20
2.1.2	Identificación de técnicas avanzadas	20
2.2	Selección de metodologías y herramientas	40
2.2.1	Evaluación y selección de frameworks de deep learning	20
2.2.2	Selección de arquitecturas de red adecuadas	20

Tabla 17. Fase de investigación y selección de tecnologías final.





Número de Esquema	Nombre de Tarea	Duración (Horas)
3	Fase de Desarrollo de Modelo	179
3.1	Preparación y preprocesamiento del <i>dataset</i>	68
3.1.1	Recopilación de imágenes satelitales y generación de máscaras	38
3.1.2	Normalización y división del <i>dataset</i>	30
3.2	Implementación y entrenamiento del modelo	71
3.2.1	Configuración de modelos y definición de arquitecturas	41
3.2.2	Entrenamiento inicial del modelo	30
3.3	Evaluación y ajuste de hiperparámetros	40
3.3.1	Evaluación de métricas de rendimiento	20
3.3.2	Optimización iterativa de hiperparámetros	20

Tabla 18. Desarrollo del modelo final.



Número de Esquema	Nombre de Tarea	Duración (Horas)
4	Fase de Validación y Evaluación de Resultados	130
4.1	Validación con <i>datasets</i> internos	40
4.2	Validación con <i>datasets</i> externos	50
4.2.1	Pruebas de generalización con datos nuevos	20
4.2.2	Ajustes y correcciones según resultados	30
4.3	Análisis de precisión y métricas	40
4.3.1	Cálculo de métricas clave y análisis cualitativo	20
4.3.2	Comparación de resultados entre modelos	20

Tabla 19. Fase de validación y evolución de los resultados final.





Número de Esquema	Nombre de Tarea	Duración (Horas)
5	Documentación y Redacción del Proyecto	166
5.1	Redacción de la metodología	66
5.1.1	Documentación detallada de cada fase	36
5.1.2	Explicación de decisiones clave	30
5.2	Análisis y presentación de resultados	60
5.2.1	Presentación gráfica de resultados	30
5.2.2	Discusión de logros y limitaciones	30
5.3	Redacción final y conclusiones	40
5.3.1	Redacción de conclusiones y mejoras futuras	20
5.3.2	Preparación de la versión final del proyecto	20

Tabla 20. Fase de documentación y redacción del proyecto final

3.5 PRESUPUESTOS

3.5.1 Presupuesto Inicial

Para realizar el presupuesto se han utilizado los salarios que se utilizaron en el trabajo realizado en la asignatura Dirección y planificación de proyectos informáticos del grado, obtenidos del sitio web Glassdoor [1] (los salarios han sido redondeados por simplicidad a la hora de hacer los cálculos).

Rol	Precio por Hora (€)
Jefe de Proyecto (Tutor de TFG)	35
Investigador	22
Analista	20
Arquitecto de Software	30
Ingeniero de Procesamiento	22
Programador	25
Administrador de Servidores	24
Tester	22
Validador de <i>Dataset</i>	22
Documentador Técnico	20

Tabla 21. Precio por hora de los diferentes puestos.

3.5.2 Presupuesto de Costes

El **presupuesto de costes** detalla los recursos financieros necesarios para llevar a cabo cada fase del proyecto. Cada tarea se valora en función de las horas estimadas en la planificación del WBS y del coste por hora de los roles asignados, según los datos de salarios obtenidos en Glassdoor. Este presupuesto proporciona una visión clara de la inversión requerida y permite evaluar los recursos necesarios en cada fase.

Número de esquema	Nombre de tarea	Rol	Horas	Tarifa por Hora (€)	Coste Total (€)
1.1.1	Identificación de necesidades y requisitos	Jefe de Proyecto	10	35	350
1.1.2	Definición del alcance y metas específicas	Jefe de Proyecto	10	35	350
1.2.1	Establecimiento del cronograma inicial	Analista	10	20	200
1.2.2	Asignación de recursos y roles clave	Analista	10	20	200
	Total (Fase 1)		40		1.100

Tabla 22. Fase de Planificación y Definición de Objetivos.



Número de esquema	Nombre de tarea	Rol	Horas	Tarifa por Hora (€)	Coste Total (€)
2.1.1	Revisión bibliográfica sobre segmentación	Investigador	20	20	400
2.1.2	Identificación de técnicas avanzadas	Investigador	20	20	400
2.2.1	Evaluación y selección de frameworks	Analista	20	20	400
2.2.2	Selección de arquitecturas de red adecuadas	Arquitecto de Software	20	25	500
	Total (Fase 2)		80		1.700

Tabla 23. Fase de Investigación y Selección de Tecnologías.



Número de esquema	Nombre de tarea	Rol	Horas	Tarifa por Hora (€)	Coste Total (€)
3.1.1	Recopilación de imágenes y generación de máscaras	Ingeniero de Procesamiento	30	22	660
3.1.2	Normalización y división del <i>dataset</i>	Ingeniero de Procesamiento	30	22	660
3.2.1	Configuración de modelos y definición de arquitecturas	Programador	30	25	750
3.2.2	Entrenamiento inicial del modelo	Programador	30	25	750
3.3.1	Evaluación de métricas de rendimiento	Programador	20	25	500
3.3.2	Optimización iterativa de hiperparámetros	Programador	20	25	500
	Total (Fase 3)		160		3.820

Tabla 24. Fase de Desarrollo de Modelo.



Número de esquema	Nombre de tarea	Rol	Horas	Tarifa por Hora (€)	Coste Total (€)
4.1	Validación con <i>datasets</i> internos	Tester	40	22	880
4.2.1	Pruebas de generalización con datos nuevos	Validador de <i>Dataset</i>	20	22	440
4.2.2	Ajustes y correcciones según resultados	Validador de <i>Dataset</i>	20	22	440
4.3.1	Cálculo de métricas clave y análisis cualitativo	Analista	20	20	400
4.3.2	Comparación entre distintos modelos	Analista	20	20	400
	Total (Fase 4)		120		2.560

Tabla 25. Fase de Validación y Evaluación de Resultados.





Número de esquema	Nombre de tarea	Rol	Horas	Tarifa por Hora (€)	Coste Total (€)
5.1.1	Documentación detallada de fases	Documentador Técnico	30	20	600
5.1.2	Explicación de decisiones clave	Documentador Técnico	30	20	600
5.2.1	Presentación gráfica de resultados	Documentador Técnico	30	20	600
5.2.2	Discusión de logros y limitaciones	Documentador Técnico	30	20	600
5.3.1	Redacción de conclusiones y mejoras futuras	Documentador Técnico	20	20	400
5.3.2	Preparación de la versión final	Documentador Técnico	20	20	400
	Total (Fase 5)		160		3.200

Tabla 26. Documentación y Redacción del Proyecto.



3.5.3 Resumen del presupuesto de costes

A continuación, se muestra el resumen del presupuesto de costes, que asciende a 12,380 €. Como se ha señalado anteriormente, esta es únicamente una estimación basada en la planificación inicial. Los valores finales pueden experimentar cambios significativos debido a factores como retrasos, posibles riesgos, problemas imprevistos o errores en la estimación inicial.

Fase	Total (€)
Fase de Planificación y Definición de Objetivos	1.100
Fase de Investigación y Selección de Tecnologías	1.700
Fase de Desarrollo de Modelo	3.820
Fase de Validación y Evaluación de Resultados	2.560
Documentación y Redacción del Proyecto	3.200
Total del Proyecto	12.380

Tabla 27. Totales del presupuesto de costes.



3.5.4 Presupuesto de Cliente

Tras realizar el presupuesto general, se ha creado un presupuesto específico para el cliente, en el cual se omiten partidas que no serán visibles para ellos, tales como la planificación inicial y la documentación interna de gestión. Además, se ha añadido un porcentaje de beneficio estimado en un 25%, basado en el tiempo invertido y la complejidad técnica del proyecto.

Para ajustar el presupuesto, el beneficio y las partidas no visibles se han distribuido entre las partidas que serán presentadas al cliente.

Concepto	Importe (€)
Beneficio 25%	3,095.00
Partidas no visibles	1,200.00
Total a ponderar	4,295.00
Porcentaje de ponderación	35%

Tabla 28. Ajustes de presupuesto.

Como se observa en la tabla anterior, el porcentaje de ponderación se ha calculado dividiendo el total a ponderar entre la suma de los precios de las partidas visibles para el cliente. Este ajuste asegura que el presupuesto al cliente refleje los costes necesarios y el margen de beneficio deseado.

Código	Partida	Importe (€)
1	Fase de Investigación y Selección	2,295.00
2	Fase de Desarrollo de Modelo	5,157.00
3	Fase de Validación y Evaluación de Resultados	3,456.00
4	Documentación y Redacción del Proyecto	4,121.00
Total	Presupuesto Cliente	15,029.00

Tabla 29. Presupuesto cliente.

El presupuesto para el cliente asciende a 15,029 €, lo que incluye un beneficio estimado de 3,095 €. Es importante señalar que todos los montos se presentan sin incluir el IVA, y están sujetos a ajustes según las condiciones finales del proyecto.



3.6 PRESUPUESTO FINAL DE COSTES

En el siguiente apartado se muestra el resumen del presupuesto final de costes teniendo en cuenta los cambios hechos en la planificación.

Fase	Total (€)
Fase de Planificación y Definición de Objetivos	1.280
Fase de Investigación y Selección de Tecnologías	1.700
Fase de Desarrollo de Modelo	4.271
Fase de Validación y Evaluación de Resultados	2.780
Documentación y Redacción del Proyecto	3.320
Total del Proyecto	13.351

Finalmente, también se mostrará un resumen del presupuesto percibido por el cliente.

3.7 PRESUPUESTO FINAL PARA EL CLIENTE

Concepto	Importe (€)
Beneficio 25%	3.292,75
Partidas no visibles	1.280,00
Total a ponderar	4.572,75
Porcentaje de ponderación	36%

Tabla 30. Datos para el cálculo de presupuesto final de Cliente.

Código	Partida	Importe (€)
1	Fase de Investigación y Selección	2.312,00
2	Fase de Desarrollo de Modelo	5.805,36
3	Fase de Validación y Evaluación de Resultados	3.776,80
4	Documentación y Redacción del Proyecto	4.515,20
Total	Presupuesto Cliente	16.409,36

Tabla 31. Presupuesto final del cliente.



3.8 INFORME DE LECCIONES APRENDIDAS

A lo largo de este proyecto, he adquirido una visión práctica sobre la gestión integral de un desarrollo técnico, enfrentándome a desafíos reales que me han permitido aprender y mejorar en múltiples áreas. Uno de los aspectos más relevantes ha sido la gestión de riesgos. Identificar y anticipar posibles obstáculos me enseñó a desarrollar un enfoque preventivo, logrando que los problemas tuvieran un menor impacto en el avance del proyecto. Aspectos como la disponibilidad de datos y las dificultades técnicas no previstas fueron superados gracias a un esfuerzo constante en la planificación de contingencias.

La experiencia de adaptar el cronograma inicial y realizar ajustes en la planificación me hizo comprender que un proyecto exitoso no siempre sigue el plan original. Los cambios fueron necesarios, y aprender a ser flexible y a reajustar tareas y tiempos fue fundamental para cumplir con los objetivos a pesar de las adversidades.

Por otro lado, el cálculo del presupuesto y su gestión fueron un aprendizaje valioso. Desde la estimación inicial hasta el ajuste final, esta tarea me mostró la importancia de estructurar y controlar los costos en función de las necesidades reales del proyecto, al tiempo que se mantenía la claridad en la presentación al cliente. Incorporar un margen de beneficio de manera transparente fue también esencial, ayudándome a entender mejor cómo presentar el valor del trabajo de forma que sea comprensible y justo para todas las partes.

En conjunto, este proyecto ha sido una experiencia que me ha brindado habilidades para abordar futuros desafíos de manera más estratégica y eficaz, entendiendo que la adaptabilidad, la previsión y la gestión cuidadosa de recursos y tiempos son pilares para el éxito en cualquier desarrollo técnico.

Capítulo 4 DETECCIÓN DE PLANTAS SOLARES EN IMÁGENES SATELITALES

ESTADO DEL ARTE

Este trabajo tiene como objetivo desarrollar un modelo que permita detectar y segmentar áreas ocupadas por plantas solares en imágenes satelitales. Utilizando datos de bandas espectrales, el proyecto se enfoca en identificar de manera precisa parques solares en grandes superficies mediante técnicas avanzadas de aprendizaje profundo. Esto facilitará la localización y cuantificación de las plantas, aportando una visión detallada de su distribución y extensión para optimizar su seguimiento y gestión.

4.1 ESTADO DEL ARTE EN LA SEGMENTACIÓN DE IMÁGENES SATELITALES Y PLANTAS SOLARES

La segmentación de imágenes satelitales ha evolucionado como un campo crucial para la observación remota y el análisis de infraestructuras desde una perspectiva aérea. Identificar elementos específicos en imágenes satelitales requiere técnicas avanzadas que puedan lidiar con la gran cantidad de datos y la variabilidad en las condiciones ambientales. La segmentación de plantas solares, en particular, es un desafío debido a la similitud espectral con otros elementos y a la gran extensión de las infraestructuras. El desarrollo de este campo ha pasado de métodos tradicionales a enfoques modernos de Aprendizaje Profundo, permitiendo una mayor precisión y generalización en tareas de segmentación.

Los métodos tradicionales, que dominaban la segmentación hasta hace poco, se basan en propiedades sencillas de los píxeles, como intensidad y bordes, para distinguir entre diferentes elementos de la imagen. Sin embargo, estos métodos presentan limitaciones en entornos complejos. Con el avance del Aprendizaje Profundo, las redes neuronales profundas, especialmente las Redes Neuronales Convolucionales (CNNs), han mostrado una notable mejora en la capacidad de identificar y segmentar infraestructuras complejas como plantas solares, aprovechando su habilidad para aprender representaciones de características de alto nivel y contextos a diferentes escalas.

4.2 TÉCNICAS EXISTENTES PARA LA SEGMENTACIÓN DE PLANTAS SOLARES EN IMÁGENES DE SATÉLITE

Existen diversas técnicas para la segmentación de plantas solares en imágenes satelitales, las cuales se dividen principalmente en dos categorías: métodos tradicionales y métodos basados en Aprendizaje Profundo. Los primeros se centran en técnicas de procesamiento de imágenes y análisis de píxeles, mientras que los segundos utilizan modelos complejos que pueden aprender patrones directamente de los datos, proporcionando una segmentación más precisa en escenarios complejos. A continuación, se presentan ambas metodologías en detalle.

4.2.1 Métodos Tradicionales de Visión por Computador para Segmentación de Imágenes

Los métodos tradicionales incluyen técnicas como la umbralización¹¹, la segmentación basada en bordes¹² y la segmentación por regiones¹³:

- **Umbralización:** Este método separa los píxeles en dos grupos (objeto y fondo) mediante un valor de intensidad. Aunque es rápido y simple, su efectividad se reduce en condiciones de iluminación no uniforme o con bajo contraste entre objeto y fondo.
- **Segmentación Basada en Bordes:** Detecta discontinuidades en la intensidad de los píxeles para trazar los contornos de los objetos. Aunque es útil en imágenes de alto contraste, puede ser ineficaz en entornos con bordes difusos o mucho ruido.
- **Segmentación por Regiones:** Agrupa píxeles adyacentes con características similares, utilizando técnicas como el crecimiento de regiones. Aunque puede ofrecer buenos resultados en algunas aplicaciones, es computacionalmente intensivo y menos eficiente en imágenes complejas.



Ilustración 7. Segmentación basada en bordes.

¹¹ Umbralización: técnica de segmentación que convierte una imagen en blanco y negro estableciendo un umbral de intensidad, separando píxeles por encima y debajo de este valor. Véase[10].

¹² Segmentación Basada en Bordes: método que localiza objetos en una imagen mediante la detección de bordes, típicamente utilizando operadores como Sobel o Canny. Consúltese [11].

¹³ Segmentación por Regiones: método que agrupa píxeles basándose en su homogeneidad dentro de regiones, facilitando el reconocimiento de áreas específicas en imágenes complejas. Para más detalle, consúltese [12].

4.2.2 Métodos Basados en Aprendizaje Profundo

Los enfoques modernos de Aprendizaje Profundo han superado las limitaciones de los métodos tradicionales mediante el uso de redes neuronales complejas que aprenden representaciones profundas de los datos:

- **Redes Neuronales Convolucionales (CNNs):** Arquitecturas como U-Net¹⁴ y SegNet¹⁵ se utilizan ampliamente para segmentación semántica, ya que capturan características a diferentes niveles, lo que resulta en una segmentación detallada.
- **Modelos Avanzados de Segmentación:** Arquitecturas como DeepLabV3+¹⁶ utilizan módulos de pirámide espacial dilatada para capturar el contexto a múltiples escalas. Esto permite una segmentación precisa de objetos con variabilidad de forma y tamaño, como los paneles solares.
- **Transferencia de Aprendizaje¹⁷:** Emplear modelos preentrenados en grandes conjuntos de datos y ajustarlos para tareas específicas ha aumentado la eficiencia y precisión en la segmentación de imágenes satelitales. Esto reduce el tiempo de entrenamiento y mejora la capacidad del modelo en entornos específicos.

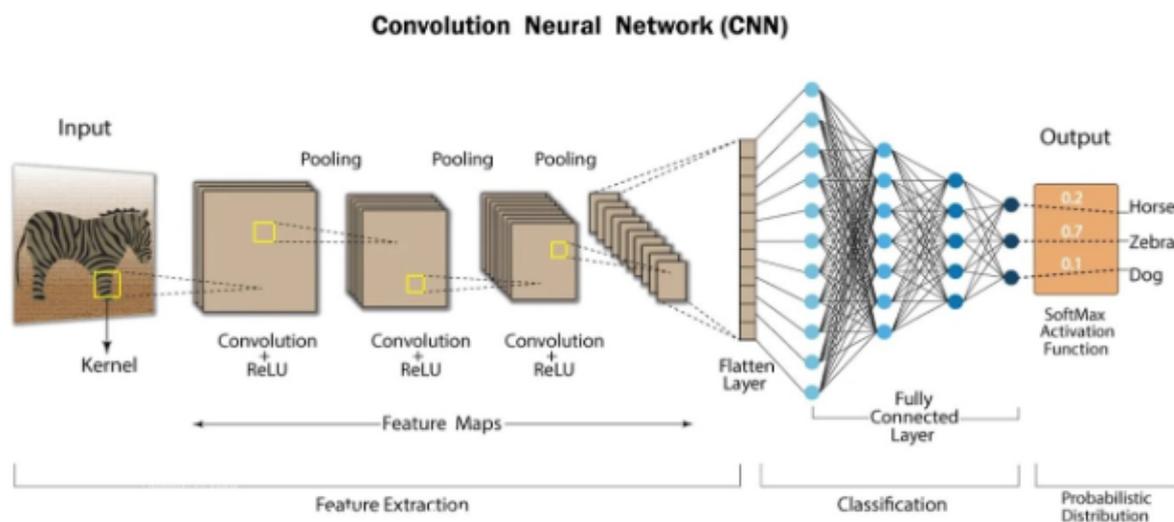


Ilustración 8. Arquitectura CNN.

¹⁴ U-Net es una arquitectura de redes neuronales convolucionales diseñada para la segmentación de imágenes, capturando características en múltiples niveles de detalle mediante convoluciones y capas de 'up-sampling'. Véase artículo de U-Net [13].

¹⁵ SegNet es una arquitectura de redes neuronales convolucionales orientada a la segmentación, que emplea un enfoque de encoder-decoder para reconstruir la imagen segmentada a partir de características comprimidas. Véase SegNet [14].

¹⁶ DeepLabV3+: modelo de segmentación que integra módulos dilatados para captar información contextual a distintas escalas, ideal para segmentación de formas complejas. Para un análisis completo, consúltese [15].

¹⁷ Transferencia de Aprendizaje: técnica que adapta modelos entrenados en grandes *datasets* a nuevas tareas específicas, optimizando su rendimiento y eficiencia. Más información en [16].



4.2.3 Abordaje mediante Visión por Computador e Inteligencia Artificial

La combinación de técnicas de Visión por Computador e Inteligencia Artificial ha permitido avances significativos en la segmentación y clasificación de objetos en imágenes satelitales. La Visión por Computador ofrece herramientas para el tratamiento, procesamiento y análisis de imágenes, facilitando la extracción de características que permiten que los ordenadores procesen y entiendan la información visual de manera similar a los humanos.

La Inteligencia Artificial, y en particular el Aprendizaje Profundo, permite el desarrollo de modelos capaces de realizar tareas complejas de clasificación y detección en grandes volúmenes de datos. En el contexto de la segmentación de plantas solares, el Aprendizaje Profundo permite a las redes neuronales aprender representaciones jerárquicas y complejas de los datos.

Para llevar a cabo una segmentación eficaz en imágenes satelitales, se deben considerar varios factores clave:

- **Elección del conjunto de datos:** Es fundamental seleccionar un *dataset* representativo para entrenar los clasificadores, especialmente en sistemas supervisados. La calidad y diversidad de los datos determinan en gran medida el rendimiento del modelo.
- **Clasificadores Supervisados y No Supervisados:** Los clasificadores supervisados requieren un conjunto de datos etiquetado, mientras que los no supervisados pueden agrupar muestras basándose en sus características inherentes. En la segmentación de plantas solares, el enfoque supervisado suele ser el preferido debido a su mayor precisión.
- **Técnicas de Preprocesamiento:** La normalización de valores, eliminación de ruido y ajuste de contraste son técnicas esenciales para mejorar la calidad de los datos y facilitar la segmentación.
- **Extracción de Características:** La selección de características como bordes, textura y color mejora la precisión en la clasificación. Para la segmentación de plantas solares, se consideran características como la reflectancia de los paneles solares y los patrones espectrales.

La elección del clasificador y del método de segmentación dependerá del objetivo del análisis, la disponibilidad de datos etiquetados y los recursos computacionales. En el caso de la segmentación de plantas solares, los enfoques supervisados basados en aprendizaje profundo son los más comunes debido a su alta precisión y capacidad de generalización.



4.2.4 Métodos para Mejorar la Precisión de los Resultados Obtenidos

Es común suponer que los errores en los resultados se deben exclusivamente a fallos en el modelo de clasificación; sin embargo, estos pueden originarse en diversas fuentes. Para implementar un sistema preciso y robusto, es fundamental identificar las causas de los errores antes de atribuirlos al algoritmo de segmentación. Los errores pueden tener diferentes orígenes, tales como:

- **Calidad del conjunto de datos:** Un *dataset* de entrenamiento insuficiente o no representativo puede llevar a una clasificación incorrecta en nuevas imágenes.
- **Calidad de las imágenes:** Aspectos como la resolución, el ruido y las condiciones de iluminación pueden afectar los resultados de la segmentación y dificultar la detección de los objetos de interés.
- **Preprocesamiento de las imágenes:** Un preprocesamiento deficiente, como un ajuste inadecuado del contraste o una eliminación ineficaz del ruido, también puede introducir errores que impactan en la precisión del modelo.

Detectar y abordar estas fuentes de error es clave para mejorar la precisión del modelo, ya que no siempre se debe asumir que el problema reside en el algoritmo de clasificación en sí. Uno de los métodos más utilizados para evaluar la precisión de los resultados en una **clasificación binaria** es la **matriz de confusión**, que permite analizar detalladamente cómo el modelo clasifica las dos clases: "presencia de planta solar" y "ausencia de planta solar".

En una clasificación binaria, la matriz de confusión se representa como una matriz 2×2 en la que:

- **Verdaderos Positivos (VP):** Casos en los que el modelo predice correctamente la presencia de una planta solar.
- **Falsos Positivos (FP):** Casos en los que el modelo predice la presencia de una planta solar cuando no la hay.
- **Verdaderos Negativos (VN):** Casos en los que el modelo predice correctamente la ausencia de una planta solar.
- **Falsos Negativos (FN):** Casos en los que el modelo predice la ausencia de una planta solar cuando en realidad está presente.

El análisis de la matriz de confusión en este contexto permite calcular métricas importantes como la **precisión**¹⁸, **sensibilidad (recall)**¹⁹ y **especificidad**²⁰, proporcionando una visión clara de la efectividad del modelo en cada categoría. Este tipo de análisis ayuda a ajustar y refinar el modelo, permitiendo una detección más precisa de las plantas solares en imágenes satelitales.

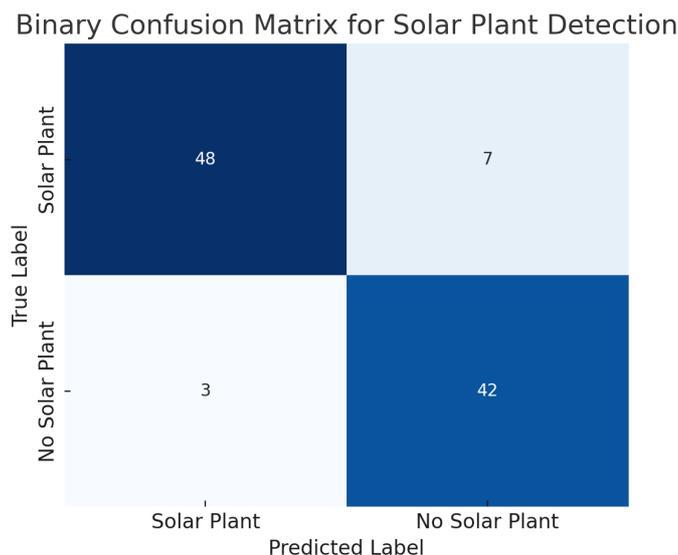


Ilustración 9. Matriz de confusión binaria.

¹⁸ Mide la proporción de predicciones correctas entre las muestras que el modelo ha clasificado como positivas. Indica cuán confiable es el modelo cuando predice una clase positiva.

¹⁹ Refleja la capacidad del modelo para identificar correctamente todas las muestras positivas. Es la proporción de verdaderos positivos entre el total de positivos reales, y mide qué tan bien el modelo detecta todos los casos de la categoría deseada.

²⁰ Cálculo de la proporción de verdaderos negativos entre el total de negativos reales. Esta métrica evalúa la efectividad del modelo al excluir correctamente las muestras que no pertenecen a la clase positiva.

4.3 TIPOS DE PLANTAS SOLARES

Las plantas solares pueden clasificarse en:

- **Plantas Fotovoltaicas:** Emplean paneles solares para convertir la luz solar directamente en electricidad. Estos paneles pueden ser estáticos o móviles (seguimiento solar) para maximizar la captación de energía.
- **Plantas Termosolares:** Utilizan espejos o lentes para concentrar la luz solar y generar calor, que posteriormente se convierte en electricidad mediante turbinas.

Para este estudio, nos enfocamos en las plantas fotovoltaicas con paneles estáticos debido a su prevalencia y a las características distintivas que presentan en imágenes satelitales.

4.4 BASES DE DATOS UTILIZADAS

La construcción y selección de una base de datos adecuada es un paso crítico en cualquier proyecto de segmentación y clasificación, ya que una mala elección puede afectar la precisión y aplicabilidad de los resultados. Para llevar a cabo este proyecto, en lugar de utilizar *datasets* ya existentes en plataformas como Kaggle o ImageNet, se optó por generar un conjunto de datos propio que cubriera las necesidades específicas de detección de plantas solares en imágenes satelitales.

Para la creación de este *dataset*, se emplearon imágenes de Sentinel-2 proporcionadas por el programa Copernicus, obtenidas a través de la plataforma STAC API. Estas imágenes fueron recolectadas de diferentes ubicaciones alrededor del mundo y en distintas estaciones del año, asegurando una gran variedad en las condiciones ambientales, como iluminación y vegetación circundante. Este enfoque permitió capturar la diversidad visual necesaria para entrenar un modelo robusto en la detección de plantas solares en contextos geográficos y temporales variados.

El modelo ResNet50, preentrenado en grandes conjuntos de datos generales, se utilizó como base, adaptándolo y ajustándolo con el *dataset* propio generado. Este proceso de "fine-tuning" con datos específicos de Sentinel-2 ayudó a mejorar la capacidad del modelo para identificar correctamente las plantas solares en imágenes satelitales.

Al crear un *dataset* específico para este proyecto, fue posible capturar las características relevantes de las plantas solares y optimizar el modelo para escenarios reales de detección. Este enfoque personalizado ha sido fundamental para maximizar la precisión y minimizar los errores en la segmentación de infraestructuras solares en imágenes satelitales.

4.5 ESTUDIOS REALIZADOS

La detección de infraestructuras solares en imágenes satelitales ha sido un área de interés en la investigación, con enfoques que tienden a dividirse en dos líneas principales. Por un lado, existen estudios dedicados a la identificación de paneles solares individuales, especialmente en entornos urbanos sobre tejados de edificios. Por otro lado, se ha investigado la segmentación de distintos tipos de terreno, como áreas agrícolas o forestales, con aplicaciones en la clasificación del uso del suelo y la gestión de estos paisajes.

A continuación, se examinan algunos ejemplos representativos de estudios en ambos enfoques, resaltando los métodos aplicados y los resultados obtenidos en la detección de paneles solares y la segmentación de coberturas terrestres en imágenes satelitales.

4.5.1 Detección y Segmentación de Módulos Fotovoltaicos en Imágenes RGB Aéreas

Este estudio se centra en la identificación de módulos fotovoltaicos en imágenes RGB obtenidas con drones sobre plantas solares en Chile. Los investigadores aplicaron técnicas de aprendizaje automático para desarrollar un proceso de segmentación que permite detectar módulos en arreglos fotovoltaicos. La aplicación de estos métodos facilita la inspección y la monitorización de plantas solares en gran escala, mejorando la eficiencia en la gestión de estos activos energéticos. [2]

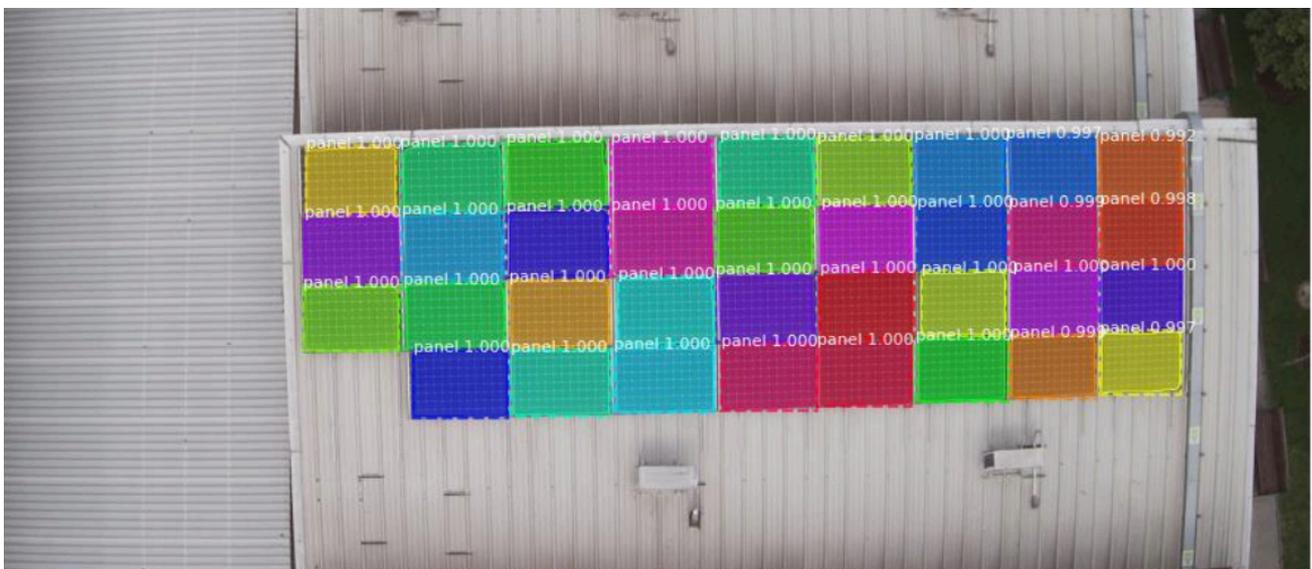


Ilustración 10. Detección de paneles solares.

4.5.2 Procesamiento Inteligente de Imágenes Satelitales: Métodos Avanzados y Aplicaciones

Este trabajo abarca el desarrollo de modelos y algoritmos avanzados para la clasificación orientada a objetos en imágenes ópticas y de radar SAR. Este enfoque, relativamente reciente en imágenes satelitales, permite clasificar de manera precisa diferentes tipos de coberturas terrestres y cuenta con aplicaciones en áreas como la agricultura y la gestión ambiental. [3]

4.5.3 Identificación de Sistemas Fotovoltaicos Residenciales mediante Imágenes Satelitales de Alta Resolución

Investigadores de Maxar Technologies utilizaron imágenes satelitales de alta resolución (31-15,5 cm) para identificar instalaciones fotovoltaicas en tejados en el sur de Alemania. Gracias a la alta resolución de las imágenes, este método logró una precisión del 97,8% en la detección de sistemas solares residenciales, evidenciando la importancia de contar con datos de alta calidad en la identificación de infraestructuras fotovoltaicas.

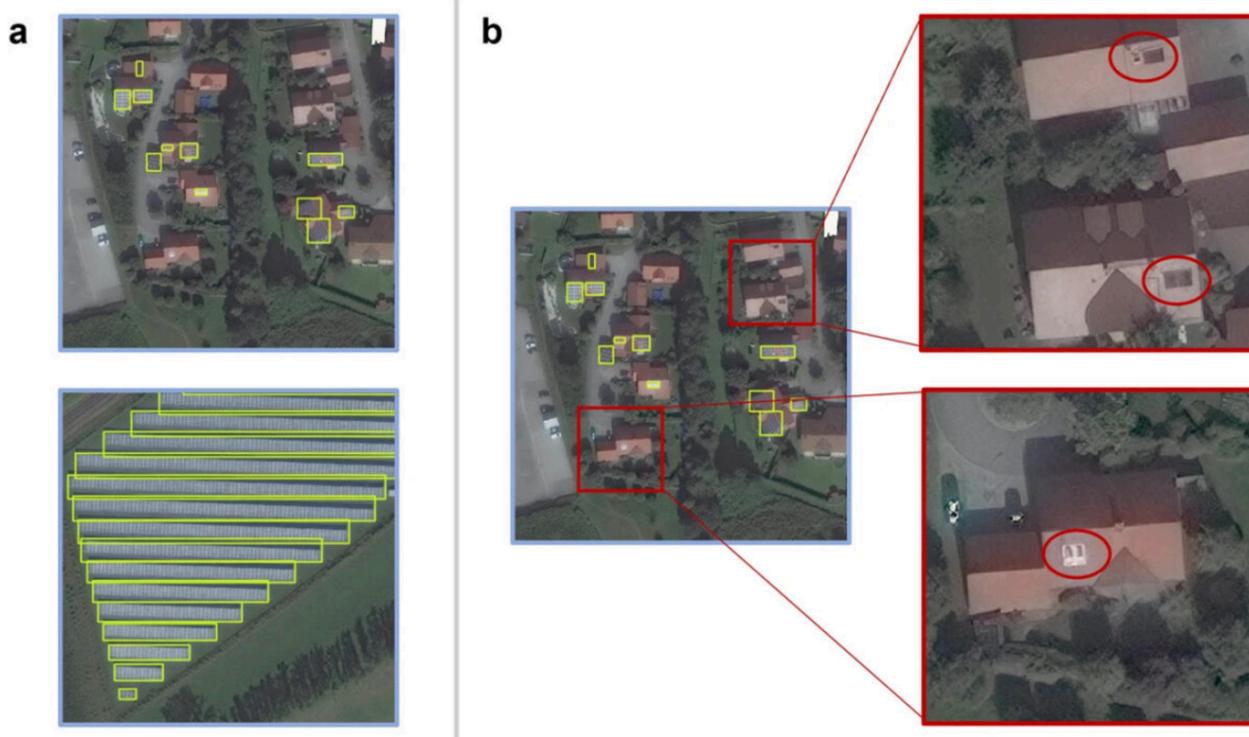


Ilustración 11. Detección de paneles solares en edificios. [4]

4.5.4 Modelo de Segmentación Semántica en Imágenes Satelitales basado en Redes Neuronales Convolucionales

Este estudio propone un modelo de segmentación semántica utilizando redes neuronales convolucionales (CNN) para clasificar tipos de coberturas en ecosistemas de páramo. La precisión en la clasificación de coberturas terrestres permite una mejor conservación y gestión de estos ecosistemas. [5]

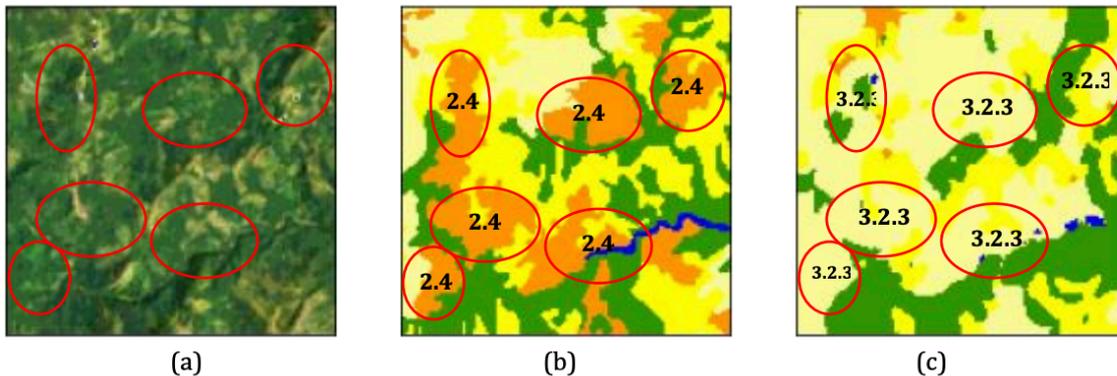


Ilustración 12. Clasificación de tipos de terreno.

4.5.5 Identificación de Sistemas Solares Descentralizados en Imágenes Aéreas mediante Deep Learning

En este estudio, se utilizó una arquitectura U-Net para segmentar rápidamente sistemas solares descentralizados en imágenes aéreas. El modelo requería un número reducido de datos de entrada y menos recursos de hardware en comparación con otros enfoques, destacando su efectividad en la identificación de pequeñas infraestructuras solares mediante aprendizaje profundo. [6]



Ilustración 13. Detección de paneles en edificios 2.



4.6 ESCASEZ DE ESTUDIOS SOBRE LA DETECCIÓN Y SEGMENTACIÓN DE PLANTAS SOLARES

Si bien existen numerosos estudios sobre la detección de paneles solares individuales y la segmentación de coberturas terrestres agrícolas o forestales, los estudios enfocados específicamente en la **detección y segmentación de plantas solares completas** son relativamente escasos. La mayoría de las investigaciones se centran en el reconocimiento de paneles solares en tejados residenciales o en la clasificación de terrenos agrícolas, mientras que la identificación de plantas solares como unidades de infraestructura a gran escala es un área menos explorada.

Esta falta de estudios específicos sobre plantas solares puede deberse a varios factores, entre ellos:

- **Enfoque en Aplicaciones Tradicionales de Segmentación:** La demanda de herramientas para el análisis de coberturas terrestres, como zonas agrícolas y forestales, ha sido históricamente alta y ha impulsado el desarrollo de métodos de segmentación en esas áreas.
- **Reciente Crecimiento del Sector de Energía Renovable:** La expansión de las energías renovables, especialmente la solar, ha ganado interés en años recientes, y la necesidad de herramientas específicas para monitorear grandes plantas solares está surgiendo a medida que este sector sigue creciendo.

El presente estudio surge precisamente de esta necesidad. Al centrarse en la detección y segmentación de plantas solares como una infraestructura unificada, pretende llenar un vacío en la investigación actual. El objetivo es proporcionar un enfoque robusto para la identificación de plantas solares en imágenes satelitales, permitiendo una monitorización eficaz de estos activos energéticos y apoyando su gestión a gran escala.

4.7 REDES NEURONALES CONVOLUCIONALES PARA EXTRACCIÓN DE CARACTERÍSTICAS

Las Redes Neuronales Convolucionales, o CNNs, son modelos de aprendizaje profundo diseñados para procesar y analizar datos con una estructura de cuadrícula, como las imágenes. Inspiradas en la conectividad de las neuronas en el córtex visual humano, estas redes imitan el procesamiento jerárquico del cerebro para reconocer patrones y objetos en imágenes. Las CNNs son especialmente útiles en Visión Artificial, ya que permiten realizar tareas como la detección de bordes y la clasificación de objetos en imágenes y vídeos.

4.7.1 Arquitectura y Funcionamiento

La arquitectura de una CNN se organiza en capas que incluyen:

- **Capas Convolucionales:** Estas capas aplican filtros o "*kernels*" sobre la imagen de entrada para detectar características fundamentales, como bordes, texturas y formas. Cada filtro genera un "mapa de características" que representa aspectos específicos de la imagen, permitiendo que la red se enfoque en distintos patrones visuales.
 - **Kernel:** Es una pequeña matriz de valores que se desplaza por la imagen, realizando operaciones de convolución para detectar patrones específicos como bordes y esquinas.
 - **Padding y Stride:** El "*padding*" añade píxeles alrededor de la imagen para preservar sus dimensiones tras la convolución, mientras que el "*stride*" controla la cantidad de desplazamiento del *kernel*, ajustando el tamaño de salida.
- **Capas de Pooling o Agrupamiento:** La capa de *pooling* reduce las dimensiones de los mapas de características, lo que simplifica los datos y acelera el procesamiento. Existen varios métodos de *pooling*, como:
 - **Max-Pooling:** Selecciona el valor máximo dentro de un área específica, destacando los elementos más prominentes.
 - **Average-Pooling:** Calcula el promedio de los valores en el área seleccionada, suavizando la información.
- **Capas de Clasificación:** Después de extraer las características, la CNN utiliza capas totalmente conectadas (*fully connected*) para clasificar las características obtenidas en categorías. La última capa de la red suele ser específica para cada problema, con neuronas que representan las posibles clases de salida, activadas por funciones como softmax para generar una probabilidad de clasificación.

4.7.2 Entrenamiento de CNNs

El entrenamiento de las CNNs implica ajustar los pesos de los filtros para optimizar la extracción de características y mejorar la precisión del modelo. El proceso de entrenamiento se basa en la **propagación hacia adelante** (*feed-forward*) y la **retro propagación del error** (*backpropagation*):

- En ***feed-forward***, los datos de entrada se procesan a través de las capas, generando una salida que se compara con la salida deseada.
- ***Backpropagation*** ajusta los pesos de los filtros en función del error de salida, permitiendo que la red aprenda a reconocer las características relevantes.

A medida que las imágenes avanzan en la red, las capas iniciales detectan características generales, como bordes, mientras que las capas profundas capturan patrones más complejos, como texturas y formas específicas.

4.7.3 Desafíos y Ventajas en la Clasificación de Imágenes Satelitales

Aplicar CNNs en la clasificación de imágenes satelitales presenta desafíos, como la necesidad de grandes volúmenes de datos etiquetados y plataformas computacionales de alto rendimiento. Sin embargo, las CNNs ofrecen ventajas únicas para el análisis de imágenes multiespectrales y satelitales:

- **Variabilidad Espectral y Espacial:** Las CNNs pueden adaptarse a la variabilidad en composición y patrones espaciales, característica de las imágenes satelitales.
- **Desbalance de Clases:** Las redes pueden enfrentarse a conjuntos de datos desbalanceados, ajustando los modelos para capturar mejor las clases minoritarias.
- **Capacidad para Múltiples Bandas Espectrales:** En imágenes multiespectrales, cada banda aporta información específica sobre el terreno o el objeto de estudio, y las CNNs pueden procesar estas bandas para mejorar la precisión de la clasificación.

Las CNNs han demostrado su eficacia en aplicaciones de teledetección para la clasificación de coberturas terrestres, análisis de patrones de vegetación y detección de infraestructuras como plantas solares, lo que las convierte en una herramienta esencial en Visión Artificial aplicada a imágenes satelitales.



4.8 SELECCIÓN DE HERRAMIENTAS

En el desarrollo de este proyecto, fue necesario realizar una selección fundamentada de las herramientas y metodologías para asegurar que cada decisión estuviera orientada a maximizar la eficiencia y calidad del proceso de segmentación de plantas solares en imágenes satelitales. A continuación, se detallan las opciones consideradas y las razones por las que se eligieron las herramientas finales.

4.8.1 Modelos de Segmentación: Inception ResNet V2 y DeepLabV3+

Para la segmentación de las plantas solares, se optó por emplear los modelos **Inception ResNet V2** y **DeepLabV3+** debido a sus capacidades avanzadas en tareas de visión por computadora y segmentación de imágenes complejas. Inception ResNet V2 combina la eficiencia de la red Inception con las mejoras en precisión de ResNet, lo que lo hace particularmente adecuado para capturar detalles en imágenes de gran tamaño. **DeepLabV3+**, por otro lado, es una arquitectura de segmentación de última generación que integra un mecanismo de convolución atrous (dilated convolutions) para capturar contextos en múltiples escalas, lo cual es esencial para obtener segmentaciones precisas en imágenes satelitales, donde los objetos a identificar suelen ser de tamaños variables y distribuciones irregulares.

Otras alternativas consideradas incluían **U-Net** y **Mask R-CNN**. U-Net es muy popular en tareas de segmentación médica y en algunos proyectos de imágenes de satélite, pero su arquitectura se optimiza principalmente para tareas de segmentación más pequeñas y no es tan eficiente en grandes volúmenes de datos. Mask R-CNN, aunque preciso, requiere un mayor tiempo de entrenamiento y recursos computacionales más elevados, lo que limita su aplicabilidad en proyectos con plazos ajustados. En comparación, DeepLabV3+ ofreció un equilibrio óptimo entre precisión y velocidad de entrenamiento en nuestro contexto, mientras que Inception ResNet V2 se destacó por su capacidad para manejar detalles finos, algo esencial para la correcta detección de plantas solares.



4.8.2 Proveedor de Imágenes Satelitales: Sentinel-2

Para obtener las imágenes satelitales, se decidió utilizar el satélite **Sentinel-2**, que es parte del programa Copernicus de la Unión Europea y ofrece acceso gratuito a imágenes de resolución de 10 metros. Esta elección se fundamentó en la disponibilidad gratuita de datos actualizados, lo cual permite realizar estudios de larga duración sin incurrir en costos elevados.

Se consideró también la posibilidad de utilizar imágenes de **PlanetScope**, que ofrecen una resolución espacial superior (3 metros) y mayor frecuencia de actualización. Sin embargo, el coste asociado a PlanetScope resultaba prohibitivo para este proyecto, por lo que se decidió priorizar Sentinel-2. Aunque la resolución es menor, Sentinel-2 proporciona una calidad adecuada para la detección de plantas solares en extensiones amplias y garantiza un acceso continuo y sin restricciones, lo cual es esencial para la viabilidad económica y operativa del proyecto.

4.8.3 Infraestructura de Procesamiento y Almacenamiento: Google Cloud y Amazon Web Services

La infraestructura de procesamiento y almacenamiento es un componente crucial para el manejo y análisis de los grandes volúmenes de datos generados en este proyecto. Para entrenar los modelos de segmentación, se optó por utilizar **Google Cloud Platform (GCP)** debido a la disponibilidad de GPUs de alto rendimiento, lo cual facilitó significativamente el entrenamiento de los modelos de aprendizaje profundo en plazos razonables. Google Cloud ofreció una infraestructura más accesible en cuanto a disponibilidad de gráficas (GPUs), algo que resultó ser más complejo en Amazon Web Services (AWS), donde la demanda de GPUs suele ser mayor y la asignación más restringida.

En cuanto a la obtención y almacenamiento de imágenes, se eligió **Amazon Web Services (AWS)** debido a su facilidad de configuración y la disponibilidad de servicios como S3, que permite almacenar y gestionar grandes volúmenes de datos de manera flexible. AWS también se utilizó en parte del preprocesamiento, ya que ofrece herramientas y APIs de integración que simplifican el flujo de trabajo inicial. En comparación, montar el sistema de obtención de imágenes en Google Cloud hubiera requerido una configuración más compleja, por lo que AWS resultó más eficiente para estas tareas específicas.



4.8.4 Otras Herramientas y Librerías

Además de las herramientas principales, el proyecto hizo uso de diversas librerías y plataformas para el procesamiento de datos y la gestión de imágenes satelitales. **Geopandas** y **Rasterio** fueron empleadas para el manejo de datos espaciales y el preprocesamiento de imágenes, permitiendo un tratamiento más efectivo de los datos de entrada antes de su uso en los modelos de aprendizaje profundo. Asimismo, la **STAC API** y **Overpass Turbo** facilitaron el acceso a imágenes y datos vectoriales específicos de plantas solares, acelerando la recopilación de *datasets* necesarios.

4.8.5 Conclusión

La selección de herramientas en este proyecto se basó en un análisis detallado de las alternativas disponibles, considerando factores como el costo, la eficiencia, la facilidad de uso y la adecuación a los objetivos específicos. Las elecciones realizadas permitieron abordar eficazmente los desafíos del proyecto, optimizando recursos y garantizando resultados de calidad. La combinación de modelos avanzados, datos accesibles y plataformas de computación adecuadas fue clave para el éxito del proyecto.

Capítulo 5 DISEÑO Y DESARROLLO

FASE DE DESARROLLO



5.1 ESQUEMA GENERAL DEL ALGORITMO PROPUESTO

El diseño y desarrollo de este proyecto se centra en la creación de un algoritmo de segmentación para la identificación de plantas solares en imágenes satelitales. Este algoritmo ha sido concebido para abordar los desafíos específicos de la teledetección en infraestructuras solares, utilizando técnicas avanzadas de procesamiento de imágenes y aprendizaje profundo. A lo largo de este capítulo, se describirá cada fase del proceso, desde la recopilación de datos hasta el entrenamiento y evaluación del modelo, enfatizando los aspectos técnicos y las decisiones metodológicas clave que sustentan la estructura del sistema.

Para optimizar el flujo de trabajo y mejorar la eficiencia de procesamiento, se ha distribuido el proyecto entre dos servidores en la nube:

- **Servidor en AWS:** Se utiliza un servidor en Amazon Web Services (AWS) para la obtención y procesamiento inicial de las imágenes satelitales de Copernicus Sentinel-2. Este servidor facilita la conexión con la STAC API, la descarga de imágenes y el procesamiento preliminar, incluyendo el recorte y la normalización de los datos.
- **Servidor en Google Compute Engine:** Para la creación de máscaras, el entrenamiento del modelo y el procesamiento intensivo de datos, se emplea un servidor en Google Cloud Platform (GCP) con una tarjeta gráfica NVIDIA T4. Esta GPU es fundamental para realizar las operaciones de deep learning de manera eficiente, especialmente en el entrenamiento de redes neuronales convolucionales para segmentación de imágenes.

A continuación, se presenta el esquema general del algoritmo, dividido en fases:

1. **Obtención de Datos:** La calidad y relevancia de los datos son fundamentales para el éxito de cualquier modelo de aprendizaje profundo. En esta fase inicial, se realiza la recopilación de imágenes satelitales y datos vectoriales.
2. **Generación de Máscaras y Preprocesamiento del Dataset:** Para entrenar eficazmente el modelo de segmentación, es necesario disponer de una representación precisa de las áreas de interés (las plantas solares). Esto se consigue mediante la generación de máscaras a partir de los datos vectoriales. Adicionalmente, el preprocesamiento de los datos permite estructurar y enriquecer el conjunto de entrenamiento, mejorando la capacidad del modelo para generalizar en diferentes entornos y condiciones visuales.
3. **Entrenamiento y Evaluación del Modelo de Segmentación:** La elección y configuración del modelo de segmentación es crucial para cumplir con los objetivos de precisión y eficiencia en la identificación de plantas solares. En esta etapa, se explica cómo se seleccionó y ajustó un modelo de red neuronal convolucional (CNN), cómo se entrenó con los datos preparados y cómo se evaluó para asegurar su efectividad.



Este enfoque secuencial asegura que cada componente del sistema esté alineado con el objetivo general de detectar y segmentar infraestructuras solares a gran escala. En los siguientes apartados, se describirán en profundidad cada uno de estos pasos, destacando tanto los aspectos técnicos como las decisiones clave que sustentan el desarrollo del algoritmo propuesto.

5.2 OBTENCIÓN DE DATOS

El primer paso en la creación del dataset consistió en la obtención de imágenes satelitales que contienen plantas solares fotovoltaicas de distintas partes del mundo. Esto es fundamental para el entrenamiento adecuado de la red neuronal, ya que permite contar con ejemplos específicos de áreas que contienen plantas solares, asegurando así que el modelo pueda aprender a identificarlas. Este proceso comenzó con la identificación de las áreas de las plantas utilizando **Overpass Turbo** [7]. Posteriormente a estas se les aplicó un buffer ²¹a través de la plataforma **geojson.io** [8], para definir con precisión las zonas de interés. Con esta información geoespacial, se descargaron las imágenes satelitales correspondientes, y se recortaron las zonas específicas de las plantas solares utilizando un notebook²² en Python en un servidor de AWS²³ con JupyterLab²⁴, automatizando la descarga y procesamiento de las imágenes mediante una STAC API.

Este flujo de trabajo permitió generar un conjunto de imágenes ráster con las características específicas necesarias para el análisis.

²¹ En geoprocésamiento, un buffer es una zona delimitada alrededor de un objeto geográfico, que define un área de influencia o proximidad a ese objeto.

²² Es un documento interactivo que permite combinar código ejecutable, visualizaciones y texto, facilitando el análisis de datos y la documentación en un solo archivo.

²³ Amazon Web Services es una plataforma de servicios en la nube que proporciona una amplia gama de recursos informáticos y de almacenamiento, utilizada para alojar y ejecutar aplicaciones de manera escalable y segura.

²⁴ Es un entorno de desarrollo interactivo para trabajar con notebooks, código y datos en el lenguaje de programación Python, ampliamente utilizado en análisis de datos y aprendizaje automático.

5.2.1 Obtención de Vectoriales con Overpass Turbo

Overpass Turbo es una herramienta de consulta para la base de datos de OpenStreetMap que permite extraer información geoespacial específica mediante consultas en lenguaje Overpass Query Language. En este caso, se utilizó para obtener los límites de las plantas solares fotovoltaicas mediante una consulta que filtra por tipo de infraestructura y método de generación.

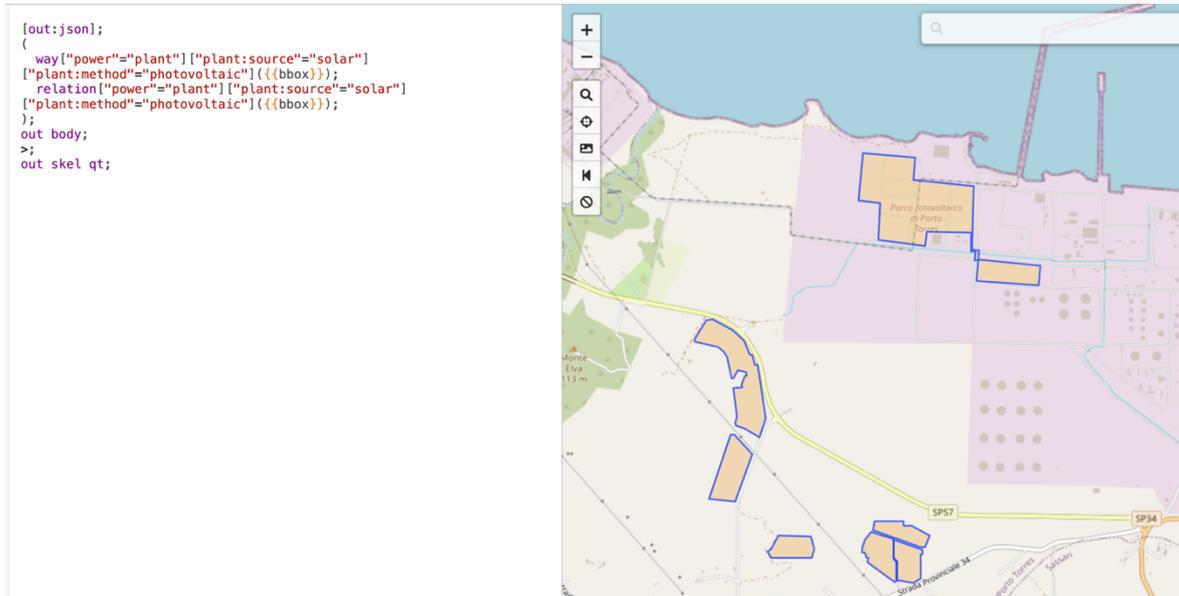


Ilustración 14. Ejemplo de uso de Overpass-Turbo.

Esta consulta selecciona todas las entidades de OpenStreetMap etiquetadas como plantas de energía cuyo método de generación es "fotovoltaico" y cuya fuente es "solar". La consulta se define para un área delimitada ({{bbox}}), lo que permite obtener solo las plantas solares dentro de una región específica del mapa. La salida de Overpass Turbo se guarda en formato **GeoJSON**, un formato de datos geoespaciales ampliamente utilizado que permite almacenar geometrías, atributos y relaciones espaciales de los objetos.

5.2.2 Edición de Vectoriales en geojson.io

Una vez exportados los resultados en formato .geojson, los archivos se cargan en **geojson.io** para realizar ajustes manuales. Geojson.io es una herramienta en línea que permite visualizar, editar y modificar archivos GeoJSON, facilitando la manipulación de los polígonos que representan las plantas solares. Dado que las imágenes de satélite deben incluir áreas circundantes, es importante agrupar las geometrías en un área más amplia de unos 15-25 km². Esto no solo asegura que todas las plantas solares y las áreas adyacentes estén correctamente representadas, sino que también facilita el preprocesamiento posterior, como el **random cropping** (recorte aleatorio) del *dataset*.

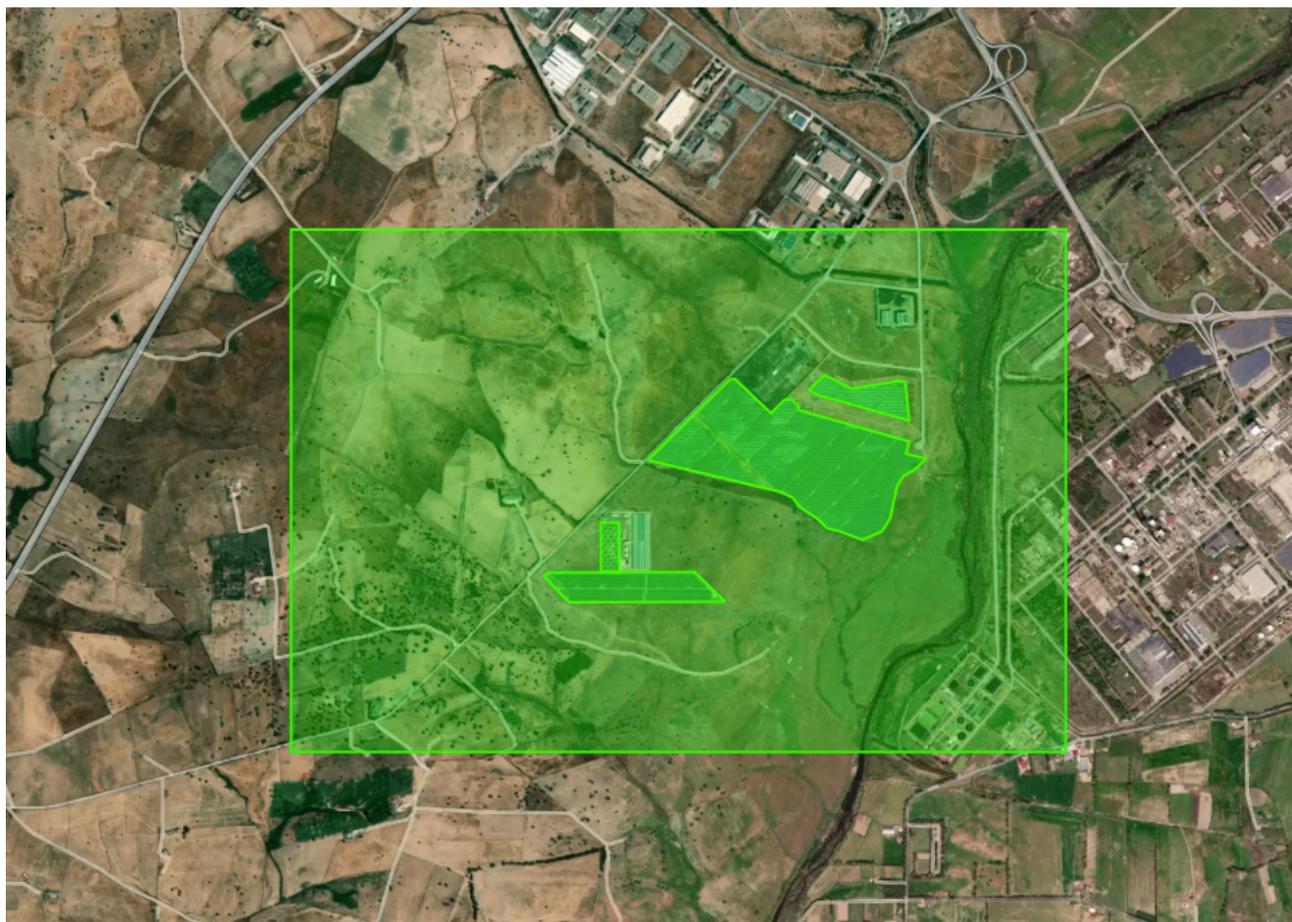


Ilustración 15. Ejemplo de uso de geojson.io.

El archivo .geojson resultante, con los límites de la planta solar y sus áreas circundantes, se carga en el servidor de AWS, donde se utiliza **JupyterLab** para gestionar de manera ágil los notebooks de procesamiento automático de las imágenes.



5.2.3 Descarga y Procesamiento de Imágenes Satelitales con la STAC API

Para obtener las imágenes satelitales, se utiliza el servicio **Sentinel-2 L2A** de la **STAC API** alojada en <https://earth-search.aws.element84.com/v1/>. Esta API permite acceder a datos de Sentinel-2 almacenados en AWS y ofrece una interfaz estructurada que facilita la búsqueda de escenas por área, fecha, nivel de nubosidad y otras características específicas. Sentinel-2 L2A es un conjunto de datos que proporciona imágenes multiespectrales con 13 bandas y una resolución de hasta 10 metros por píxel en las bandas de interés (RGB y NIR). Las bandas utilizadas en este proyecto fueron las bandas 3, 4, 5 y 8, que corresponden a los canales RGB y NIR.

El notebook en Python automatiza el proceso de consulta a la STAC API, donde se especifican varios parámetros, tales como:

- **Área de interés:** obtenida del bounding box del archivo GeoJSON cargado previamente.
- **Rango de fechas:** configurado para seleccionar imágenes de una temporada específica o durante un período determinado.
- **Porcentaje de nubosidad:** se define un límite máximo de nubosidad para evitar imágenes obstruidas, garantizando así la claridad de las áreas analizadas.

5.2.4 Georreferenciación, Re-proyección y Creación de Rasters

Para procesar las imágenes descargadas, se utilizan varias librerías de Python, entre ellas **Rasterio** y **Geopandas**, que permiten realizar operaciones avanzadas de manejo y manipulación de datos geoespaciales. Las etapas principales son:

- **Georreferenciación y Reproyección:** Las imágenes descargadas se re proyectan al sistema de coordenadas EPSG:4326, conocido como **WGS 84**, un sistema de referencia geoespacial ampliamente utilizado para datos de coordenadas globales. La re proyección asegura que todas las imágenes estén en un sistema de referencia común, lo cual es esencial para integrarlas con otros datos geoespaciales y analizarlas correctamente.
- **Recorte por Área de Interés:** Utilizando los límites definidos en el archivo GeoJSON, cada imagen se recorta para incluir solo el área de la planta solar y sus alrededores. Este recorte optimiza el tamaño de las imágenes y facilita su manejo en el procesamiento posterior.
- **Composición de Bandas en un Raster RGB:** Finalmente, las bandas RGB se combinan en un único archivo ráster multibanda en formato .tif. Este paso garantiza que las imágenes estén listas para ser procesadas en el modelo de segmentación.

5.2.5 Evaluación de la Utilidad del Índice NDVI

Inicialmente, se consideró el uso del índice **NDVI** (Normalized Difference Vegetation Index) para diferenciar las áreas con vegetación de las artificiales. El NDVI se calcula mediante la fórmula:

$$\text{NDVI} = \frac{\text{NIR} - \text{Red}}{\text{NIR} + \text{Red}}$$

Ilustración 16. Fórmula NDVI.

donde NIR corresponde a la banda del infrarrojo cercano y Red a la banda roja. El NDVI es comúnmente utilizado para análisis de vegetación, ya que permite identificar áreas verdes y diferenciarlas de zonas urbanas o industriales. Sin embargo, en este caso, el NDVI no proporcionó buenos resultados debido a que las áreas urbanas generaban ruido y los terrenos áridos adyacentes a las plantas solares mostraban valores de NDVI similares a los de superficies artificiales. Por esta razón, se decidió utilizar solo las bandas RGB, lo que optimiza la segmentación y reduce la introducción de ruido en el modelo.

5.2.6 Generación de Rasters Georreferenciados

El flujo de trabajo en el notebook finaliza con la creación de un ráster georreferenciado en formato .tif que combina las bandas RGB. Para asegurar la calidad de las imágenes, cada archivo .tif se verifica manualmente en **QGIS**. Esta verificación permite asegurar que las imágenes cumplen con los requisitos de baja nubosidad y claridad visual, evitando problemas en la fase de entrenamiento del modelo.

5.2.7 Resultados de la Obtención de Datos

El resultado de este proceso es un conjunto de imágenes satelitales RGB georreferenciadas con resolución de 10 metros por píxel, adecuadas para el entrenamiento de un modelo de segmentación de plantas solares. Estas imágenes corresponden a ubicaciones conocidas de plantas en diversas regiones del mundo, lo cual asegura que el modelo se entrene con ejemplos precisos y específicos.

Este conjunto de datos constituye la base para el desarrollo del algoritmo propuesto, permitiendo realizar un análisis detallado de las plantas solares en diferentes regiones y condiciones.

5.3 GENERACIÓN DE MÁSCARAS Y PREPROCESAMIENTO DEL *DATASET*

En esta etapa del proyecto, el objetivo es transformar las imágenes satelitales brutas y los datos vectoriales en un conjunto de datos procesado y estructurado, adecuado para el entrenamiento de modelos de aprendizaje profundo. Este proceso involucra varias tareas secuenciales, cada una con un propósito específico en la preparación de las imágenes y máscaras que conformarán el *dataset*. A continuación, se describen en detalle los pasos seguidos, desde la normalización de las imágenes hasta la generación de máscaras y la organización del *dataset* final.

1. **Normalización de Imágenes RGB:** Las imágenes satelitales obtenidas de Sentinel-2 poseen un rango de valores que supera el habitual en imágenes RGB (0-255). Para ajustar las imágenes al rango de valores de un modelo convencional de visión por computadora, se realizó un proceso de normalización. Esto incluyó el recorte de valores atípicos en los canales de color y su ajuste al rango estándar. Este paso es fundamental para asegurar que las imágenes sean interpretadas correctamente en el modelo de aprendizaje profundo.
2. **Generación y Conversión de Máscaras de Segmentación:** Utilizando los archivos vectoriales de las plantas solares, se generaron máscaras iniciales para cada imagen ráster, destacando las áreas de interés. Posteriormente, estas máscaras se convirtieron a un formato binario de un solo canal, donde el valor de 1 representa la presencia de una planta solar y el valor de 0 el fondo. Este proceso permite que el modelo aprenda a identificar y segmentar las plantas solares de manera precisa.
3. **Random Cropping (Recorte Aleatorio):** Dado que Sentinel-2 proporciona imágenes de gran tamaño (en términos de área geográfica), se aplicó un recorte aleatorio de las imágenes y sus respectivas máscaras para obtener fragmentos de un tamaño fijo. Esto no solo reduce el tamaño de cada entrada, sino que también incrementa la variedad espacial en el *dataset*, ayudando al modelo a generalizar mejor.
4. **Aumento de Datos (Data Augmentation):** Para mejorar la robustez del modelo y reducir el riesgo de sobreajuste, se aplicaron técnicas de aumento de datos en las imágenes recortadas. Esto incluyó rotaciones, espejados y otros cambios en las imágenes y sus máscaras, generando un conjunto de datos más diverso a partir de las muestras originales.
5. **División del *Dataset* en Conjuntos de Entrenamiento, Validación y Prueba:** El *dataset* procesado se dividió en tres subconjuntos para asegurar una evaluación adecuada del modelo. Estos conjuntos permiten entrenar el modelo y luego evaluar su rendimiento en datos no vistos, garantizando una medición de su capacidad de generalización.



6. **Evaluación del *Dataset* y Eliminación de Imágenes con Poco Contexto:** Durante la construcción del *dataset*, se evaluó la calidad y el contexto de cada imagen. Las imágenes que no contenían suficiente información o contexto útil fueron descartadas, optimizando el contenido del *dataset* para el entrenamiento y evitando ejemplos ambiguos o irrelevantes.

Cada uno de estos pasos se describe en mayor detalle a continuación, proporcionando una visión clara de cómo se preparó el *dataset* para su uso en el modelo de segmentación de plantas solares.

5.3.1 Normalización de Imágenes RGB

Las imágenes obtenidas de Sentinel-2 presentan valores de intensidad que van más allá del rango estándar RGB (0-255), debido a la calibración de los sensores satelitales y a la naturaleza de la reflectancia en la Tierra. Para asegurar la compatibilidad con los modelos de aprendizaje profundo, fue necesario realizar un proceso de normalización en los canales RGB de las imágenes.

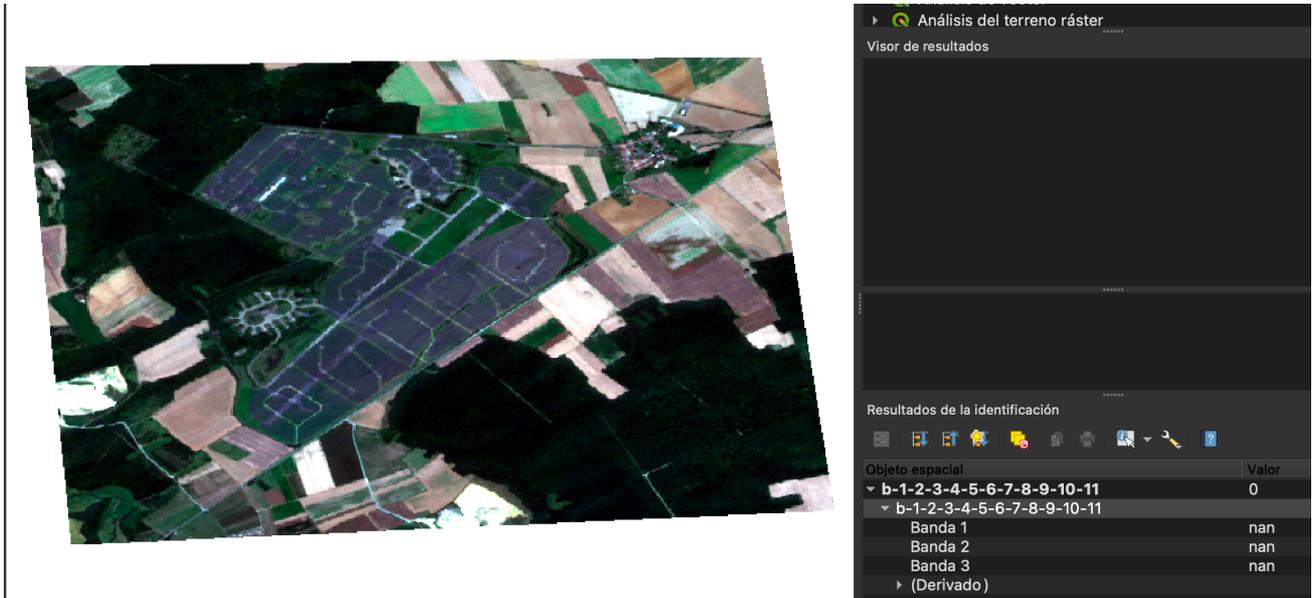


Ilustración 17. Imagen ejemplo Sentinel-2.

La normalización implicó los siguientes pasos:

1. **Eliminación de Valores NaN:** Algunos píxeles contienen valores indefinidos o NaN, que pueden interferir en el procesamiento. Estos valores se reemplazan por cero para evitar problemas de cálculo.
2. **Recorte de Valores Extremos:** Para eliminar los valores atípicos (que suelen representar puntos de alta o baja reflectancia), se calcularon los percentiles 2.5 y 97.5 en cada canal de color, y se recortaron los valores fuera de este rango. Esta técnica permite centrar los datos en un rango intermedio sin perder información relevante.
3. **Escalado al Rango 0-255:** Después del recorte, cada canal se escaló linealmente al rango de 0 a 255, transformando los valores de cada banda en un formato compatible con los modelos de procesamiento de imágenes. Este proceso asegura que las imágenes tengan una escala de intensidad uniforme, optimizando la interpretación de estas en el modelo de segmentación.

Esta normalización estandariza las imágenes y prepara el *dataset* para los siguientes pasos, asegurando que el modelo de aprendizaje profundo reciba entradas consistentes y correctamente escaladas.

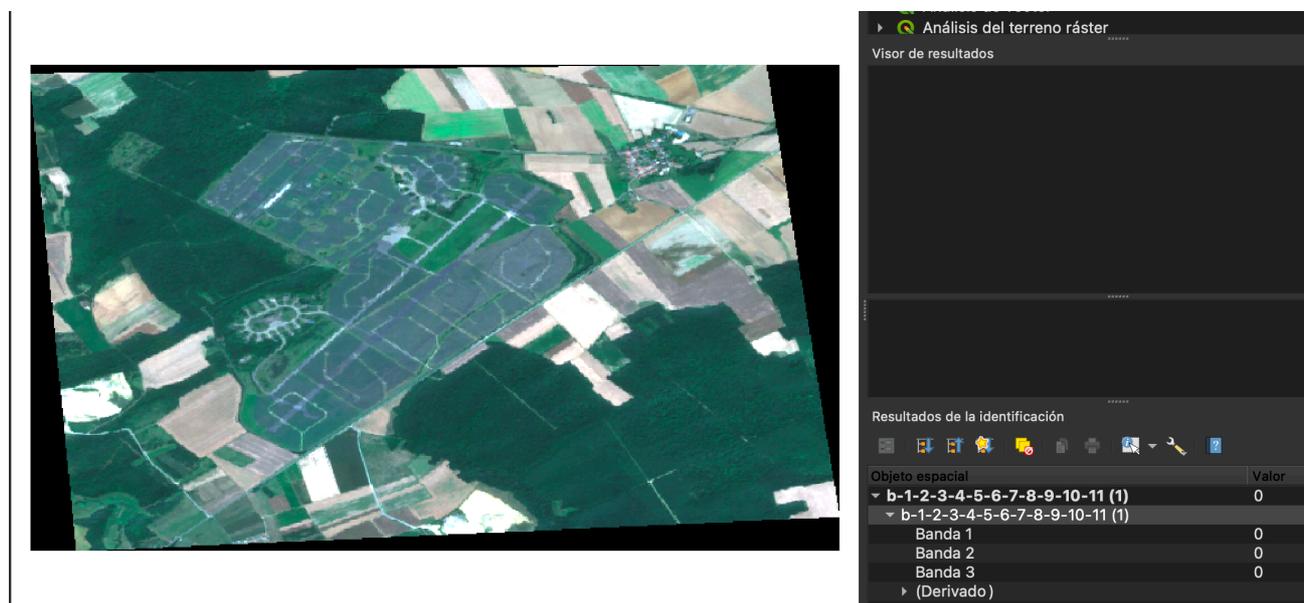


Ilustración 18. Imagen después de la Normalización.

5.3.2 Generación y Conversión de Máscaras de Segmentación

Para preparar el *dataset* de imágenes de plantas solares, el primer paso es generar máscaras de segmentación que marquen claramente las áreas ocupadas por las plantas solares. Este proceso se realiza en dos fases consecutivas, la creación de máscaras iniciales a partir de los vectoriales y la conversión de estas en máscaras binarias de un solo canal.

Generación de Máscaras

Para cada imagen en el *dataset*, se utiliza el archivo vectorial correspondiente, que contiene las geometrías de las plantas solares, y se aplica un proceso de segmentación. Las geometrías se "quemán" en una máscara en formato blanco y negro, donde las áreas de interés se marcan con un valor de 255 (blanco) y el fondo se deja en 0 (negro). Este proceso asegura que el modelo de segmentación pueda reconocer y diferenciar con claridad las plantas solares del fondo.

Conversión a Máscaras Binarias de un Solo Canal

A continuación, las máscaras se procesan para convertir todos los valores superiores a 0 en 1, reduciendo así la máscara a un formato binario. Si la máscara tiene múltiples canales, se reduce a un único canal para facilitar el entrenamiento del modelo. Este paso asegura una representación simplificada de la planta solar, destacando su presencia en la imagen de manera eficiente.

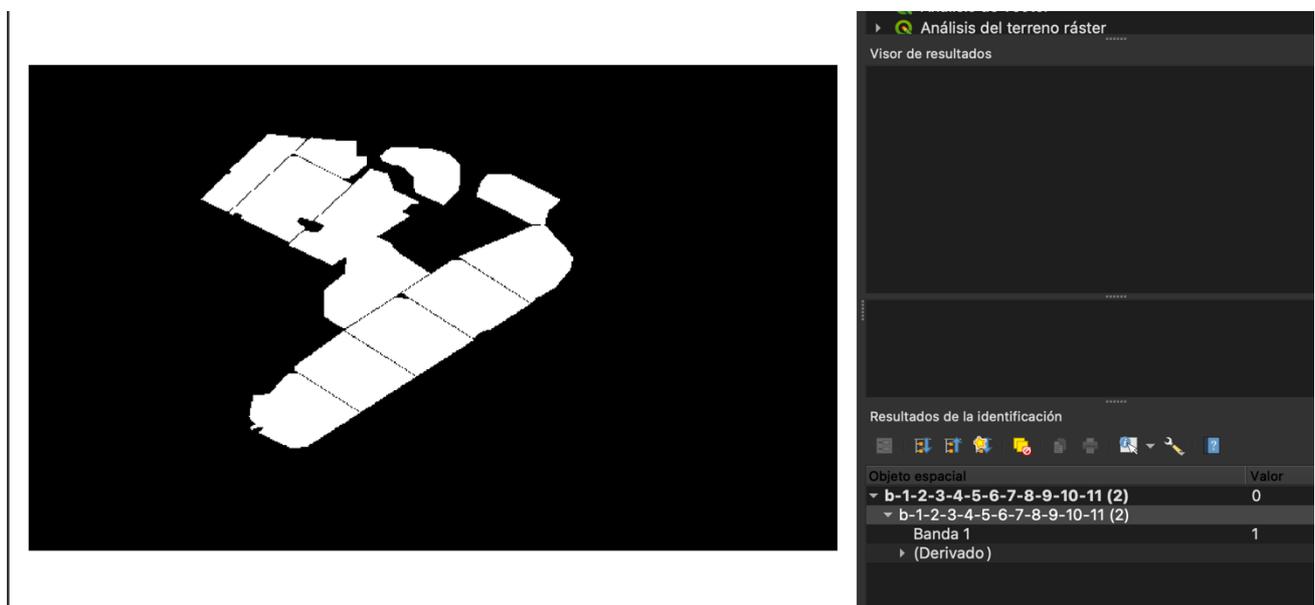


Ilustración 19. Ejemplo: Mascara binaria monobanda.



Importancia de las Máscaras de Segmentación

Las máscaras de segmentación son fundamentales para enseñar al modelo a localizar con precisión las plantas solares en imágenes satelitales. Al proporcionar una referencia explícita sobre qué áreas de la imagen corresponden a plantas solares, el modelo puede aprender a distinguir estas estructuras de otros elementos en el entorno, mejorando la precisión en la detección y reduciendo el error.



5.3.3 *Random Cropping* y Generación de Parches

El proceso de *random cropping* o recorte aleatorio de imágenes es una técnica clave en el preprocesamiento de *datasets* para segmentación, especialmente cuando las imágenes contienen objetos dispersos o pequeños, como las plantas solares en este caso. Este método ayuda a aumentar la variabilidad del *dataset* al dividir cada imagen en varios recortes de tamaño uniforme, llamados "parches". Los modelos de aprendizaje profundo pueden así aprender a reconocer patrones locales en múltiples subregiones de las imágenes, mejorando su precisión y capacidad de generalización.

Razón para Usar Random Cropping

1. **Aumenta la Diversidad del *Dataset*:** Al extraer parches de las imágenes originales, se genera un conjunto de subimágenes que representan diferentes partes de la planta solar y del entorno. Esto amplía el *dataset* de entrenamiento, permitiendo que el modelo se entrene con más ejemplos y aumentando la robustez del aprendizaje.
2. **Optimiza la Resolución en la Entrada del Modelo:** Los modelos de aprendizaje profundo suelen beneficiarse de entradas de tamaño uniforme. Al utilizar parches de un tamaño definido (en este caso, 256x256 píxeles), se facilita el entrenamiento en lotes (batch training), mejorando el rendimiento computacional.
3. **Equilibra la Resolución y las Dimensiones:** Debido a que la resolución de Sentinel-2 es de 10 metros por píxel, seleccionar parches más grandes habría requerido imágenes originales de dimensiones extremadamente grandes. Los recortes de 256x256 píxeles proporcionan un buen equilibrio, capturando suficiente contexto para identificar las plantas solares sin aumentar de forma excesiva el tamaño de las imágenes de origen.
4. **Facilita la Localización de Plantas Solares:** Al proporcionar varias vistas de una misma planta solar desde diferentes posiciones, el modelo aprende a detectar las plantas solares sin necesidad de depender de la imagen completa. Esto es especialmente útil en imágenes de resolución media, como las de Sentinel-2.

Implementación del Proceso de *Random Cropping*

El código implementa el *random cropping* en varios pasos. Para ello, se hace uso de cv2 (OpenCV) y skimage, dos librerías esenciales para la manipulación de imágenes y la creación de parches:

1. **Preparación de Directorios y Listado de Archivos:** Se organizan directorios separados para almacenar los parches de imágenes y sus correspondientes máscaras.
2. **Recorte en Parches Uniformes:**
 - Para cada imagen y su máscara asociada, se determina el tamaño y la cantidad de parches a crear en función de las dimensiones de la imagen y del tamaño del parche (256x256 píxeles).
 - Para cada parche, se calcula el punto inicial de extracción y se recorta la región correspondiente tanto de la imagen como de la máscara. Este paso asegura que cada parche de la máscara esté alineado con el correspondiente en la imagen, lo cual es crucial para la precisión de la segmentación.
3. **Condiciones para Guardar los Parches:**
 - Se comprueba que cada parche tenga el tamaño exacto especificado. Esto garantiza uniformidad y evita problemas de procesamiento en el entrenamiento.
 - Los parches se guardan utilizando cv2.imwrite y skimage.io.imsave, lo cual facilita su almacenamiento en formatos compatibles con modelos de aprendizaje profundo.

Este método de recorte aleatorio asegura que el modelo entrene con un conjunto amplio y diverso de patrones visuales, ayudándole a identificar las plantas solares en distintas configuraciones y posiciones en las imágenes.



Ilustración 20. Ejemplo: *random cropping*, imagen y máscara.

5.3.4 *Data Augmentation*: Aumento del *Dataset* mediante Transformaciones

El *data augmentation* es una técnica fundamental en el entrenamiento de modelos de aprendizaje profundo, especialmente en casos donde la obtención de datos adicionales es limitada o costosa. La estrategia se basa en aplicar transformaciones a las imágenes existentes para generar nuevas muestras de entrenamiento que, aunque derivadas de las originales, presentan variaciones en su orientación o disposición. Esto ayuda al modelo a generalizar mejor y evitar el sobreajuste, ya que se entrena en una variedad de versiones de las imágenes.

Razones para Utilizar Data Augmentation

1. **Incremento del Volumen de Datos sin Nuevas Imágenes:** Al realizar transformaciones sobre las imágenes existentes, como rotaciones, giros o cambios de perspectiva, se incrementa el tamaño del *dataset* sin necesidad de recolectar nuevas imágenes, lo cual reduce tiempo y recursos.
2. **Mejora de la Robustez del Modelo:** Estas variaciones ayudan al modelo a aprender de imágenes con distintas orientaciones y escalas, mejorando su capacidad para identificar patrones en condiciones que no fueron específicamente entrenadas en el conjunto de datos original.
3. **Reducción del Sobreajuste:** Al exponer al modelo a diferentes versiones de la misma imagen, se previene el sobreajuste, ayudando a que el modelo aprenda representaciones más generales y útiles.

Implementación de Data Augmentation

Para este proyecto, el proceso de *data augmentation* incluyó una serie de transformaciones específicas aplicadas tanto a las imágenes como a sus correspondientes máscaras de segmentación. Estas transformaciones fueron seleccionadas para reflejar posibles variaciones en las condiciones en las que las plantas solares podrían aparecer en las imágenes satelitales:

- **Rotación de hasta 180 grados:** Permite que el modelo aprenda a identificar plantas solares en diferentes orientaciones.
- **Cizalladura (shear) de hasta 0.5:** Introduce distorsiones en la perspectiva, simulando posibles cambios de ángulo en las imágenes.
- **Reflejos horizontales y verticales:** Añade simetría y diversidad a los datos, exponiendo al modelo a diversas configuraciones espaciales.



Ilustración 21. Ejemplo: imagen con data augmentation aplicada.

Código de Implementación

El código utiliza la librería ImageDataGenerator de Keras para definir las transformaciones y aplicarlas de forma simultánea a las imágenes y sus máscaras. Esto asegura que las variaciones en las imágenes tengan una correspondencia exacta en las máscaras de segmentación, manteniendo la coherencia en los datos de entrenamiento. Cada imagen original se transforma en tres nuevas versiones, ampliando así el *dataset* de forma significativa.

Proceso General:

1. **Carga y Redimensionamiento de Imágenes:** Las imágenes y máscaras se cargan y se ajustan al tamaño de entrada definido (256x256 píxeles).
2. **Aplicación de Transformaciones:** Se generan tres versiones adicionales de cada imagen y su máscara utilizando las configuraciones de ImageDataGenerator.
3. **Almacenamiento de Imágenes Aumentadas:** Los nuevos archivos se guardan en los directorios de imágenes y máscaras correspondientes, permitiendo su uso inmediato en el entrenamiento.

Este método de *data augmentation* asegura que el modelo disponga de una variedad de ejemplos que reflejan múltiples escenarios y orientaciones, contribuyendo a su capacidad de generalización y eficacia en la tarea de segmentación de plantas solares.



5.3.5 División del *Dataset* en Conjuntos de Entrenamiento, Validación y Prueba

Para asegurar un entrenamiento efectivo y una evaluación precisa del modelo de segmentación, es crucial dividir el *dataset* en tres subconjuntos: entrenamiento, validación y prueba. Esta separación permite:

- **Evitar el Sobreajuste:** Al dedicar un conjunto exclusivo para pruebas, el modelo se evalúa en datos completamente nuevos, lo cual ofrece una medida más objetiva de su rendimiento.
- **Optimización de Parámetros:** El conjunto de validación permite ajustar los hiperparámetros durante el entrenamiento, sin afectar la objetividad del conjunto de prueba.
- **Evaluación Realista:** La partición en conjuntos de datos distintos asegura que el modelo aprenda a generalizar, es decir, a reconocer patrones en imágenes fuera del conjunto de entrenamiento.

Proceso de División del Dataset

El *dataset* fue dividido en las siguientes proporciones:

- **Entrenamiento (75%):** Este conjunto contiene la mayor parte de las imágenes y máscaras, proporcionando suficiente cantidad de datos para que el modelo aprenda patrones de manera robusta.
- **Validación (5%):** Un pequeño subconjunto utilizado para revisar y ajustar el rendimiento del modelo durante el proceso de entrenamiento. Esto permite seleccionar los mejores hiperparámetros y evitar que el modelo se ajuste demasiado a los datos de entrenamiento.
- **Prueba (20%):** Este conjunto de imágenes y máscaras se utiliza únicamente para evaluar el rendimiento final del modelo después del entrenamiento.

```
: import os
import numpy as np
from skimage.io import imread, imsave
import os
#train test split
from sklearn.model_selection import train_test_split

train_labels_path = '/home/Sergiocima/deeplearning/mask-folder/dataset-plants-mask-binary-splited'
train_images_path = '/home/Sergiocima/deeplearning/raster-folder/dataset-plants-copernicus-splited'
# List all images and masks after renaming
all_images = sorted(os.listdir(train_images_path))
all_masks = sorted(os.listdir(train_labels_path))

# Ensure images and masks are in the same order
assert all(img.split('.')[0] == msk.split('.')[0] for img, msk in zip(all_images, all_masks)), "Mismatch between

# Train-test-validation split, cambiar val por test
train_imgs, test_imgs, train_masks, test_masks = train_test_split(all_images, all_masks, test_size=0.2, random_s
train_imgs, val_imgs, train_masks, val_masks = train_test_split(train_imgs, train_masks, test_size=0.05, random_

# Output the results
print(f"Train Images: {len(train_imgs)}, Validation Images: {len(val_imgs)}, Test Images: {len(test_imgs)}")
print(f"Train Masks: {len(train_masks)}, Validation Masks: {len(val_masks)}, Test Masks: {len(test_masks)}")

Train Images: 665, Validation Images: 35, Test Images: 176
Train Masks: 665, Validation Masks: 35, Test Masks: 176
```

Ilustración 22. Separación del dataset en 3.

Implementación del Código

El proceso de separación se llevó a cabo utilizando la función `train_test_split` de la biblioteca *scikit-learn*, lo cual asegura una separación aleatoria y reproducible, manteniendo la consistencia entre las imágenes y sus máscaras. Este proceso involucra:

1. **Carga de Imágenes y Máscaras:** Todas las imágenes y sus respectivas máscaras se ordenan para garantizar que cada imagen coincida con su máscara correspondiente.
2. **División en Conjuntos de Entrenamiento, Validación y Prueba:** Primero se realiza una división de 80-20 entre el conjunto de entrenamiento y prueba, y luego, dentro del conjunto de entrenamiento, se separa el 5% para el conjunto de validación.
3. **Organización en Directorios:** Cada subconjunto de datos se guarda en directorios específicos (train, val, test) y sus correspondientes máscaras (train_mask, val_mask, test_mask), asegurando una estructura de carpetas clara para el entrenamiento.

Esta organización estructurada del *dataset* permite llevar a cabo el entrenamiento, ajuste y evaluación del modelo de manera eficiente y con un flujo de trabajo ordenado, facilitando la reutilización y posibles ajustes adicionales en el futuro.



5.3.6 Evaluación del *Dataset* y Eliminación de Imágenes con Poco Contexto

Para mejorar la calidad del *dataset*, es fundamental asegurarse de que las imágenes utilizadas en el entrenamiento contengan suficiente información relevante sobre plantas solares. En este contexto, los recortes de imagen que carecen de contenido significativo, es decir, aquellos que contienen poca o ninguna presencia de las estructuras de interés, pueden introducir ruido en el entrenamiento, disminuyendo así la precisión del modelo.

Para mitigar esto, se llevó a cabo un análisis cuantitativo del porcentaje de píxeles blancos en cada máscara de segmentación. Este proceso permite identificar la proporción de área cubierta por las plantas solares, ayudando a clasificar y eventualmente eliminar los recortes con valores extremadamente bajos o nulos de píxeles correspondientes a las plantas.

Análisis y Clasificación de Imágenes por Contenido Relevante

El proceso de evaluación del contenido en cada máscara consistió en calcular el porcentaje de píxeles blancos, ya que estos indican la presencia de plantas solares en cada imagen. Posteriormente, las imágenes fueron clasificadas en distintos rangos de porcentaje de cobertura de píxeles blancos, lo que facilitó una revisión sistemática de la distribución del contenido relevante en el *dataset*.

Este análisis reveló que, debido a la naturaleza de las áreas de estudio, muchas imágenes presentaban una baja cobertura de píxeles blancos. Esto se debe a que las plantas solares ocupan una porción relativamente pequeña de los terrenos en las imágenes, dada la resolución de 10 metros por píxel de Sentinel-2 y la extensión amplia de las áreas capturadas.

Distribución de Porcentaje de Píxeles Blancos (Plantas Solares) en Máscaras

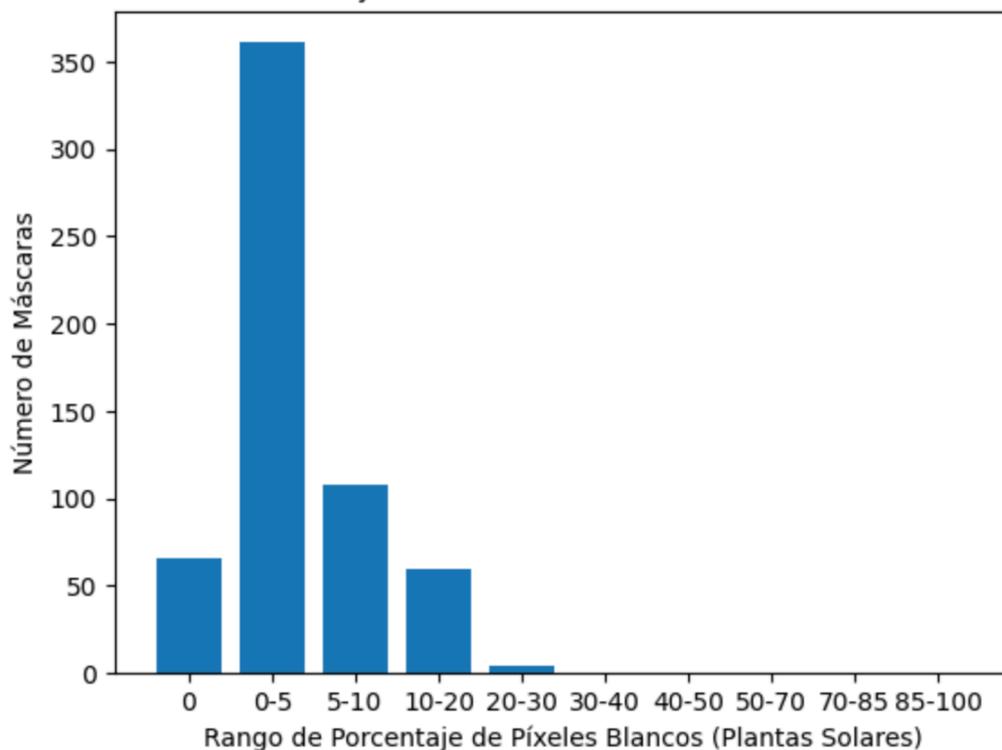


Ilustración 23. Distribución de píxeles blancos en las máscaras del dataset.

Eliminación de Imágenes con Poco Contexto

Para evitar que el modelo se entrene con imágenes que carezcan de información útil sobre plantas solares, se estableció un umbral de eliminación para los recortes de imagen completamente vacíos (0% de píxeles blancos). El código emplea un porcentaje ajustable que permite retener una muestra representativa de estas imágenes vacías para un mejor balance del *dataset*, sin saturar el conjunto de datos con ejemplos irrelevantes.

Este proceso asegura que el *dataset* final sea más informativo y preciso, facilitando que el modelo aprenda de manera más efectiva a identificar plantas solares en un entorno variado, sin ruido adicional.

5.4 ENTRENAMIENTO Y EVALUACIÓN DEL MODELO DE SEGMENTACIÓN

En este apartado se detalla el proceso de entrenamiento y evaluación de un modelo de segmentación con redes neuronales profundas, específicamente adaptado para la detección de plantas solares en imágenes satelitales. El objetivo es explorar configuraciones avanzadas que optimicen la precisión y eficiencia en la segmentación, con dos modelos candidatos: **InceptionResNetV2** y **DeepLabV3+**.

Para comenzar, se presenta una implementación detallada de la arquitectura **InceptionResNetV2**, explicando la elección de cada componente y su adaptación al contexto de la segmentación de imágenes satelitales. La implementación combina bloques de convolución²⁵, normalización²⁶ y activación²⁷ para maximizar la extracción de características, integrando un decodificador tipo U-Net para restaurar las resoluciones espaciales en el proceso de segmentación.

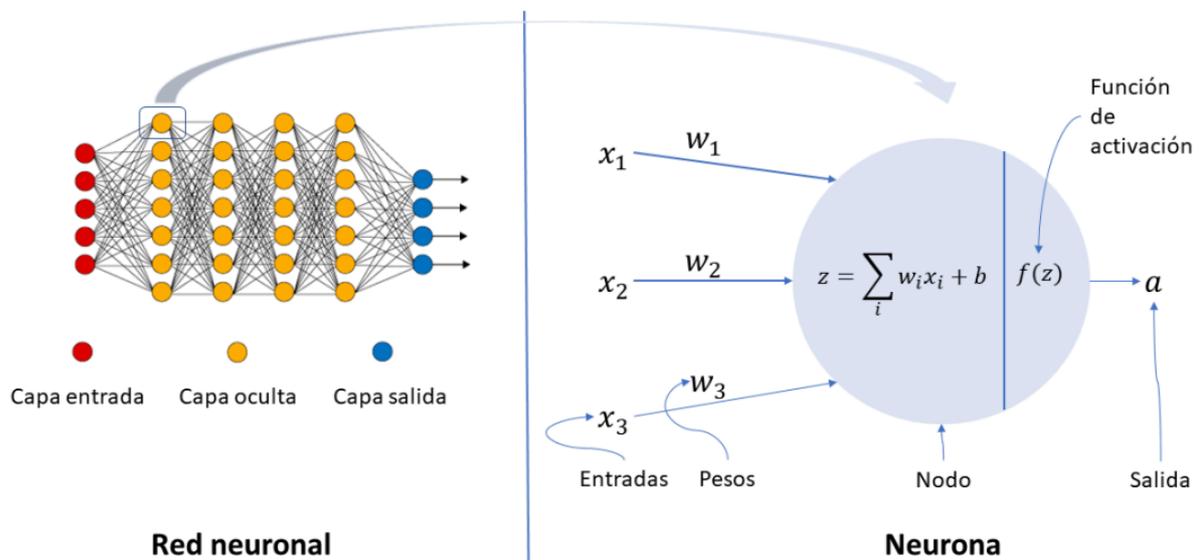


Ilustración 24. Ejemplo: red neuronal y neurona.

²⁵ Son partes de las redes neuronales que se encargan de analizar una imagen en busca de patrones específicos, como formas, bordes o texturas. Funcionan como una lupa que examina pequeñas partes de la imagen para encontrar detalles importantes.

²⁶ Es un proceso que ajusta los valores de los datos para que todos estén en el mismo rango. Esto ayuda a que la red aprenda más rápido y evite errores durante el entrenamiento. Es como nivelar el terreno antes de construir algo.

²⁷ Es la forma en que la red decide qué información es importante. Por ejemplo, una de las funciones más comunes, llamada *ReLU*, simplemente deja pasar los valores positivos y elimina los negativos, ayudando a que el sistema sea más eficiente.

5.4.1 Implementación de InceptionResNetV2 para Segmentación

La arquitectura **InceptionResNetV2** es una red profunda de convolución que incorpora módulos Inception²⁸ con bloques residuales²⁹, diseñada para segmentación de imágenes. Este diseño es ventajoso para el proyecto porque permite procesar múltiples escalas y captar patrones complejos, adaptando así la arquitectura a la segmentación de plantas solares.

Estructura del Modelo

1. Bloques de Convolución:

- Se emplean bloques de convolución con filtros de tamaño 3x3, seguidos de capas de normalización (*BatchNormalization*³⁰) y funciones de activación ReLU³¹. La elección de *BatchNormalization* mejora la estabilidad y rapidez del entrenamiento al reducir las covariables internas, ajustando dinámicamente la escala de las salidas de las convoluciones.
- La activación ReLU es fundamental porque introduce una no linealidad en el modelo, permitiendo la representación de patrones más complejos y mejorando la eficiencia computacional en comparación con funciones de activación como *sigmoid*³² o *tanh*³³.

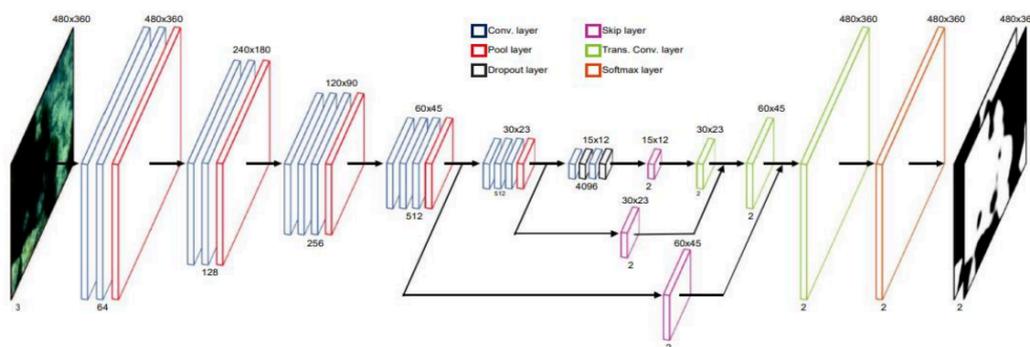


Ilustración 25. Capas convolucionales 1.

²⁸ Son partes avanzadas de una red neuronal que combinan diferentes formas de analizar las imágenes, como si usaran varias lentes al mismo tiempo para ver detalles grandes y pequeños.

²⁹ Ayudan a que la red aprenda más rápido al permitir que se salten pasos si ya se entienden ciertas partes de la imagen. Es como construir un edificio reutilizando partes que ya están completas, en lugar de empezar desde cero cada vez.

³⁰ Funciona ajustando y escalando los datos dentro de la red para que estén en un rango uniforme, evitando que valores extremos interfieran con el aprendizaje.

³¹ Es una función matemática simple que ayuda a las redes neuronales a decidir qué información es importante. Básicamente, si un número es negativo, lo convierte en cero, y si es positivo, lo deja igual.

³² Es una función matemática utilizada en redes neuronales para convertir números en valores entre 0 y 1. Ayuda a interpretar los resultados como probabilidades.

³³ función matemática que convierte números en un rango entre -1 y 1. Es útil para modelar relaciones en datos donde los valores negativos también tienen significado.

2. Decodificador:

- La estructura U-Net en el decodificador permite restaurar la resolución espacial de la imagen segmentada. Cada capa de decodificación está formada por:
 - Una capa de convolución transpuesta (*Conv2DTranspose*³⁴), que realiza una operación de *upsampling*³⁵ para restaurar la resolución.
 - Una capa de concatenación³⁶ (*Concatenate*), que combina características del decodificador con las del *encoder*³⁷ a través de conexiones *skip*³⁸, conservando detalles espaciales esenciales.
- Estas capas permiten al modelo retener tanto el contexto general de la imagen como los detalles específicos, asegurando una segmentación precisa de los paneles solares en un entorno variado.

3. Modelo Preentrenado:

- Se utiliza el modelo preentrenado InceptionResNetV2 en ImageNet como extractor de características en el encoder. Esta elección permite aprovechar representaciones ya aprendidas para la detección de formas y texturas, que son clave en la segmentación de infraestructuras como las plantas solares.
- La transferencia de aprendizaje facilita el entrenamiento del modelo en conjuntos de datos limitados, mejorando el rendimiento de segmentación sin requerir un conjunto de datos de gran tamaño.

4. Capa de Salida:

- La capa final es una convolución con activación sigmoid, adaptada para segmentación binaria. Esta capa convierte cada píxel en una probabilidad de pertenencia a la clase objetivo (presencia o ausencia de planta solar). La función sigmoid es ideal para esta tarea binaria, generando un rango continuo entre 0 y 1 que facilita la clasificación de cada píxel.

³⁴ A diferencia de la convolución normal (que reduce el tamaño de las imágenes), la convolución transpuesta aumenta el tamaño de las imágenes o mapas de características, actuando como una especie de "deshacer" la convolución.

³⁵ Es una técnica utilizada en redes neuronales para incrementar el tamaño espacial de los datos (como imágenes o mapas de características).

³⁶ En redes neuronales, una capa de concatenación combina la información de dos o más capas previas uniendo sus datos a lo largo de una dimensión específica, generalmente los canales de características. Este tipo de capa no realiza cálculos complejos; simplemente alinea las salidas de las capas que conecta.

³⁷ Es la parte de una red neuronal diseñada para convertir una entrada compleja, como una imagen, en una representación comprimida y simplificada que retiene las características esenciales.

³⁸ Son enlaces directos entre capas iniciales y avanzadas en una red neuronal que permiten transferir información, saltándose capas intermedias.



Configuración de Entrenamiento y Métricas

1. Función de Pérdida³⁹:

- Se emplea la **entropía cruzada binaria**⁴⁰ como función de pérdida, adecuada para segmentación binaria al penalizar discrepancias entre la predicción y la etiqueta real en cada píxel. La entropía cruzada asegura una optimización eficaz del modelo al maximizar la probabilidad de clasificación correcta.

2. Métrica IoU:

- Para medir la eficacia de la segmentación, se utiliza la métrica de **Intersección sobre Unión (IoU)**⁴¹. Esta métrica evalúa la precisión de la segmentación al comparar la superposición entre las áreas predichas y las áreas reales. Es particularmente útil en segmentación de imágenes, donde cuantifica la precisión en la detección de los paneles solares.

3. Callback de Guardado:

- Para asegurar la preservación de los mejores resultados, se emplea un *callback* personalizado que guarda el modelo cada 10 épocas⁴². Este método permite retomar el entrenamiento desde puntos intermedios y analizar el rendimiento a lo largo de las iteraciones.

³⁹ Es una métrica utilizada durante el entrenamiento de modelos de aprendizaje automático para medir la diferencia entre las predicciones del modelo y los valores reales. Su objetivo es guiar la optimización, ajustando los parámetros del modelo para minimizar este error.

⁴⁰ Es una función de pérdida utilizada para problemas de clasificación binaria. Mide la diferencia entre las probabilidades predichas por el modelo y las etiquetas reales, penalizando más los errores en las predicciones con alta confianza. Su objetivo es maximizar la probabilidad de predecir correctamente cada clase.

⁴¹ Es una métrica utilizada para evaluar la precisión de modelos de segmentación o detección de objetos. Calcula la superposición entre el área predicha por el modelo y el área real, dividiendo esta intersección entre la unión total de ambas áreas. Un valor más alto indica mayor precisión en la predicción.

⁴² Una *época* es una iteración completa sobre el conjunto de datos de entrenamiento, durante la cual el modelo ajusta sus parámetros para mejorar su precisión. El entrenamiento generalmente se realiza durante varias épocas para optimizar el rendimiento del modelo.

5.4.2 Modelo DeepLabV3+ para Segmentación de Plantas Solares

Para el desarrollo de un segundo enfoque en la segmentación de plantas solares, se utilizó el modelo **DeepLabV3+**. Esta arquitectura es reconocida en el campo de la visión por computadora por su capacidad de capturar contextos de diferentes escalas mediante el uso de **convoluciones dilatadas**⁴³ en combinación con redes neuronales profundas. El objetivo de este modelo es mejorar la segmentación semántica en escenarios complejos al aumentar el área de percepción de la red sin reducir la resolución espacial de las características extraídas.

En esta implementación, el modelo se basa en **TensorFlow** y **Keras**, que son librerías clave para la creación y entrenamiento de redes neuronales profundas en Python. A continuación, se explica cada sección del código, resaltando los aspectos técnicos y las decisiones de diseño.

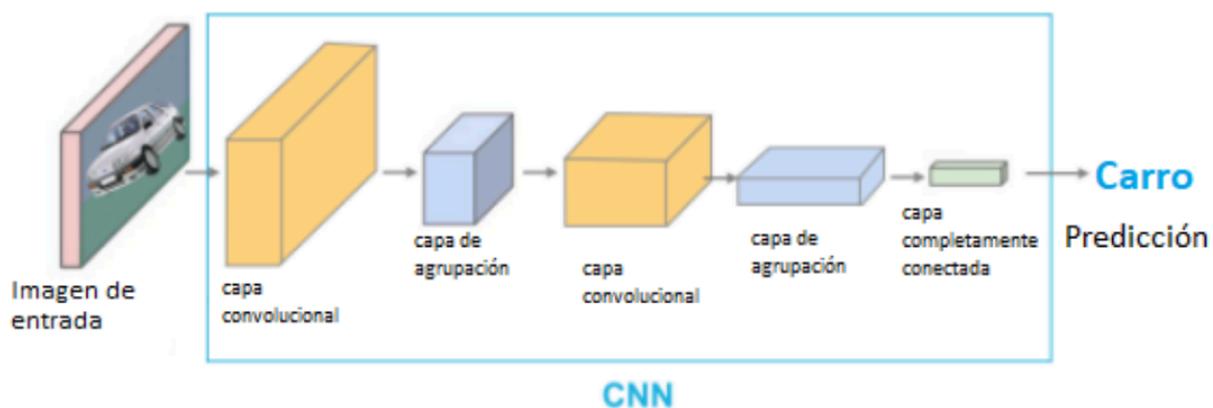


Ilustración 26. Capas convolucionales 2.

Arquitectura del Modelo DeepLabV3+

1. Bloque de Convolución:

- El bloque de convolución (*convolution_block*) es fundamental para aprender representaciones a nivel de características. Aquí, cada convolución está seguida de una **normalización por lotes**⁴⁴ (*batch normalization*) y una **función de activación ReLU** para mejorar la estabilidad del entrenamiento y la capacidad de representar no linealidades. Este bloque es reutilizado en distintas partes del modelo.

⁴³ Aumenta el área que una red puede observar en una imagen (campo de visión) al incluir "espacios" entre los valores del filtro que analiza los datos. Esto es útil para identificar patrones a diferentes escalas sin perder precisión.

⁴⁴ Es una técnica utilizada en redes neuronales para mejorar la estabilidad y velocidad del entrenamiento. Consiste en normalizar las salidas de una capa (ajustando su media y desviación estándar) dentro de un lote de datos antes de pasarlas a la siguiente capa. Esto ayuda a que la red sea menos sensible a los valores iniciales de los pesos y acelera la convergencia al entrenar el modelo.



2. Bloque de Agrupación Espacial Dilatada ⁴⁵(*Dilated Spatial Pyramid Pooling*):

- Este bloque, conocido como **ASPP** (*Atrous Spatial Pyramid Pooling*), permite al modelo capturar información de contexto a distintas escalas. Para ello, se emplean convoluciones dilatadas⁴⁶ con factores de dilatación de 1, 6, 12 y 18, lo que permite analizar regiones con diferentes niveles de detalle.
- Se utiliza una combinación de convoluciones dilatadas y una capa de *average pooling*⁴⁷, cuyos resultados son concatenados y luego procesados por una última convolución, generando así un conjunto de características ricas en contexto espacial.

3. Modelo Base (ResNet50):

- Se emplea la arquitectura **ResNet50**⁴⁸ preentrenada en ImageNet para extraer características profundas de la imagen de entrada. Esto permite aprovechar características de bajo nivel aprendidas en el dominio visual general y aplicarlas a la tarea de segmentación.
- Se seleccionan las salidas de distintas capas de ResNet50, lo cual facilita que el modelo combine información de diferentes profundidades en el decodificador.

4. Decodificador:

- El decodificador utiliza las características del ASPP y las fusiona con características intermedias de ResNet50 para recuperar detalles espaciales perdidos en el proceso de extracción de características.
- La salida se somete a varias capas de **convolución y de upsampling** para llevar la segmentación al tamaño de la imagen original. Finalmente, la capa de salida usa una función de activación **sigmoide** para producir un mapa binario que diferencia entre áreas de planta solar y fondo.

⁴⁵ Es una técnica en redes neuronales que permite analizar diferentes áreas de una imagen a la vez, desde detalles muy pequeños hasta estructuras más grandes. Esto se logra procesando partes de la imagen con distintos niveles de "salto" entre píxeles, sin reducir el tamaño de la imagen.

⁴⁶ Técnica utilizada para ampliar el alcance de la región que una convolución analiza sin aumentar el número de parámetros ni perder resolución.

⁴⁷ Es una operación en redes neuronales que reduce el tamaño de una matriz de características al calcular el promedio de los valores en regiones específicas (por ejemplo, un bloque de 2x2 píxeles) de la entrada. Esto ayuda a destacar las características más relevantes, reducir el ruido y disminuir la carga computacional.

⁴⁸ Es un modelo de red neuronal profunda compuesto por 50 capas, diseñado para resolver problemas de aprendizaje profundo mediante el uso de "conexiones residuales".



Funciones de Preprocesamiento y Configuración del Entrenamiento

1. Preprocesamiento de Etiquetas:

- La función *preprocess_labels* ajusta las etiquetas de segmentación para asegurar que estén en el formato adecuado, necesario para entrenar el modelo en Keras.

2. Cálculo de Pesos de Muestras:

- Dado que algunas clases de imágenes son más importantes o representativas que otras, se utiliza *compute_sample_weights* para asignar pesos adicionales a ciertos subconjuntos de datos. En este caso, las imágenes dentro de rangos específicos reciben un peso mayor para mejorar la precisión del modelo en esas categorías.

3. Parámetros de Entrenamiento:

- **Batch Size:** El tamaño de lote se establece en 16, lo que representa el número de muestras que se procesan en paralelo antes de actualizar los parámetros del modelo. Un tamaño de lote moderado permite balancear entre estabilidad en la convergencia y eficiencia computacional.
- **Número de Epochs:** Se especifica un máximo de 50 iteraciones, permitiendo que el modelo ajuste sus parámetros en múltiples iteraciones sobre el conjunto de datos de entrenamiento. La cantidad de iteraciones se ajusta con el fin de evitar tanto el *underfitting* (subajuste) como el *overfitting* (sobreajuste).
- **Optimizador y Función de Pérdida:** Se emplea el optimizador **Adam**⁴⁹ debido a su eficiencia y popularidad en tareas de segmentación de imágenes. La **pérdida de entropía cruzada binaria**⁵⁰ es apropiada para esta tarea binaria de segmentación, en la que se busca diferenciar entre plantas solares y fondo.

⁴⁹ Es un optimizador que ayuda a los modelos de aprendizaje profundo a aprender más rápido y de manera eficiente. Ajusta automáticamente cómo de grandes son los pasos que el modelo da en cada iteración para mejorar, usando información sobre los cambios previos.

⁵⁰ Es una forma de medir qué tan bien un modelo predice una clasificación binaria. Compara las predicciones del modelo con las respuestas correctas y asigna un puntaje; cuanto menor sea, mejor está funcionando el modelo.

4. Callbacks de Entrenamiento:

- **ModelCheckpoint:** Permite guardar el mejor modelo en función de la pérdida en el conjunto de validación, asegurando que se mantenga el mejor estado del modelo durante el entrenamiento.
- **EarlyStopping:** Detiene el entrenamiento si no hay mejora en la pérdida de validación después de 20 iteraciones, previniendo el sobreajuste y ahorrando tiempo de cómputo.
- **ReduceLROnPlateau:** Reduce la tasa de aprendizaje si no se detecta mejora en el rendimiento durante 5 iteraciones, permitiendo que el modelo realice ajustes más finos hacia el final del entrenamiento.

Entrenamiento del Modelo

El entrenamiento del modelo se realiza sobre un conjunto de datos preprocesado y dividido en subconjuntos de entrenamiento y validación. La red aprende a segmentar plantas solares a partir de la información contextual capturada en el ASPP y de las características proporcionadas por la ResNet50. Este proceso produce un modelo optimizado y preparado para evaluar su desempeño en tareas de segmentación de plantas solares en imágenes satelitales.

5.4.3 Selección de DeepLabV3+ como Modelo Final

Se elige **DeepLabV3+** sobre **Inception ResNet V2** para la segmentación de plantas solares por las siguientes razones:

1. **Capacidad de Captura de Contexto Multiescala:** El uso de ASPP en DeepLabV3+ permite capturar mejor el contexto a distintas escalas, esencial para identificar plantas solares en imágenes satelitales.
2. **Simplicidad y Eficiencia:** DeepLabV3+ ofrece una estructura menos compleja, lo que facilita su implementación y reduce el tiempo de entrenamiento sin sacrificar precisión.
3. **Optimización para Segmentación Semántica:** Diseñado específicamente para segmentación, DeepLabV3+ es más efectivo para recuperar detalles espaciales críticos en imágenes de baja resolución.

En resumen, **DeepLabV3+** proporciona un equilibrio óptimo entre precisión, eficiencia y adecuación a las necesidades específicas de segmentación en imágenes satelitales.



Capítulo 6 EVALUACIÓN DEL MODELO Y RESULTADOS



6.1 INTRODUCCIÓN

Antes de presentar los resultados, es crucial explicar el proceso de evaluación del modelo DeepLabV3+ y los parámetros utilizados para medir su rendimiento. La evaluación comenzó con la división del conjunto de datos en tres partes: entrenamiento, validación y prueba. El conjunto de entrenamiento se utilizó para ajustar los parámetros del modelo, mientras que el conjunto de validación ayudó a optimizar los hiperparámetros y prevenir sobreajuste. Por último, el conjunto de prueba, que no había sido visto por el modelo, permitió evaluar su capacidad de generalización en condiciones realistas.

Además, se incluyeron imágenes completamente nuevas, ajenas al *dataset* original, para añadir un nivel adicional de desafío. Estas imágenes, que no estaban relacionadas con las del entrenamiento ni se generaron mediante técnicas de aumento de datos, sirvieron para probar la capacidad del modelo en escenarios desconocidos, una prueba esencial para valorar su aplicabilidad en el mundo real.

Para medir el rendimiento, se utilizó la métrica Intersección sobre Unión (IoU), que compara las áreas predichas por el modelo con las áreas reales, ofreciendo una medida clara de la precisión en la segmentación. También se consideraron otras métricas como la precisión y la sensibilidad, que evalúan qué tan bien el modelo predice correctamente y detecta las plantas solares reales, respectivamente. Aunque el análisis principal fue visual, estas métricas brindaron un soporte cuantitativo valioso.

Este enfoque de evaluación, complementado por las imágenes de prueba adicionales, permitió analizar tanto la capacidad del modelo para generalizar como su desempeño en situaciones prácticas. Así, el modelo no solo demostró su eficacia en datos conocidos, sino también su potencial para enfrentar desafíos reales en escenarios diversos.

6.2 EVALUACIÓN DURANTE EL ENTRENAMIENTO

El modelo fue entrenado durante 50 épocas, utilizando la función de pérdida de *Binary Crossentropy*, que mide la discrepancia entre las predicciones del modelo y las etiquetas reales, y permite ajustar los pesos del modelo⁵¹ para mejorar su rendimiento. Como métrica principal para evaluar la segmentación se empleó la Intersección sobre Unión (IoU, por sus siglas en inglés), que proporciona una medida precisa de la superposición entre las áreas predichas y las reales, especialmente útil en tareas de segmentación de imágenes.

El proceso de entrenamiento fue monitorizado en cada época para evaluar la pérdida y la precisión, tanto en el conjunto de entrenamiento como en el de validación, permitiendo ajustar hiperparámetros⁵² y controlar posibles problemas de sobreajuste. Además, se utilizó un sistema de validación cruzada para garantizar la robustez del modelo y su capacidad para generalizar a datos no vistos.

A continuación, se detallan los resultados obtenidos, incluyendo gráficos que ilustran la evolución de las métricas clave a lo largo del entrenamiento.

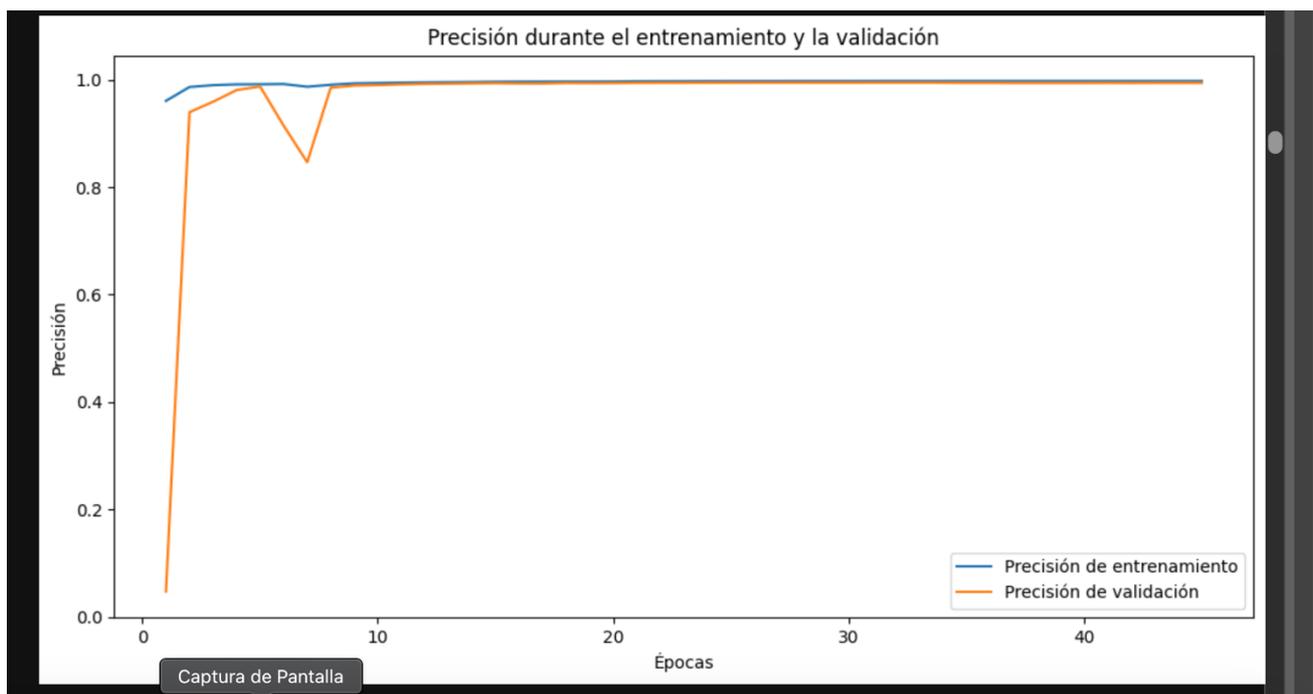


Ilustración 27. Imagen de la gráfica de precisión del modelo Deeplabv3+.

⁵¹ Valores internos que el modelo ajusta durante el entrenamiento para aprender patrones en los datos y generar predicciones.

⁵² Configuraciones ajustables como tasa de aprendizaje o tamaño de lotes que se establecen antes del entrenamiento y determinan el comportamiento del modelo.

6.2.1 Análisis de Resultados de Entrenamiento

Los gráficos de precisión y pérdida revelaron un posible sobreajuste del modelo, ya que la precisión de entrenamiento alcanzó valores cercanos al 99,8%, mientras que la precisión en el conjunto de validación mostró ligeras oscilaciones. Sin embargo, la verdadera capacidad de generalización se evaluó posteriormente con un conjunto de imágenes totalmente nuevas para verificar si el modelo podía mantener un rendimiento satisfactorio en datos no vistos.

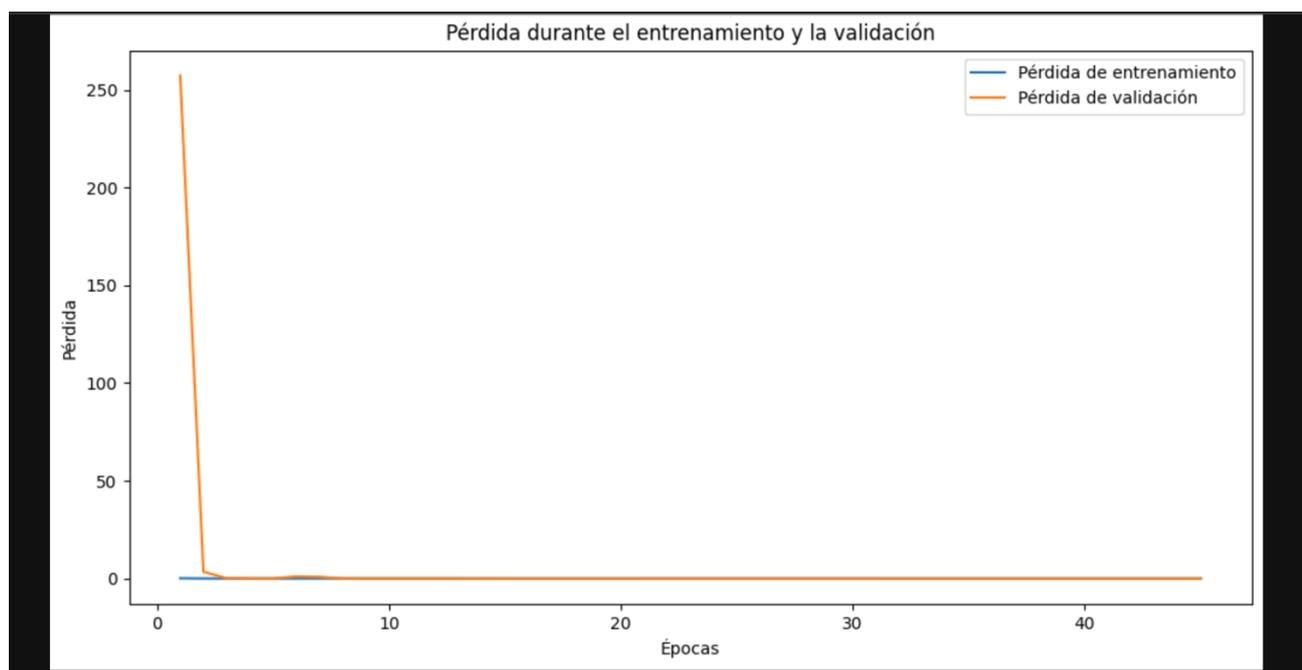


Ilustración 28. Imagen de la gráfica de la pérdida del modelo Deeplabv3+.

6.3 EVALUACIÓN EN CONJUNTO DE DATOS NO VISTO

Para evaluar la capacidad de generalización del modelo, se utilizó un conjunto de datos de validación independiente, compuesto por imágenes satelitales que el modelo no había visto anteriormente. En esta evaluación, se generaron predicciones de segmentación para estas imágenes, las cuales fueron comparadas visualmente con las máscaras de verdad de terreno (ground truth).

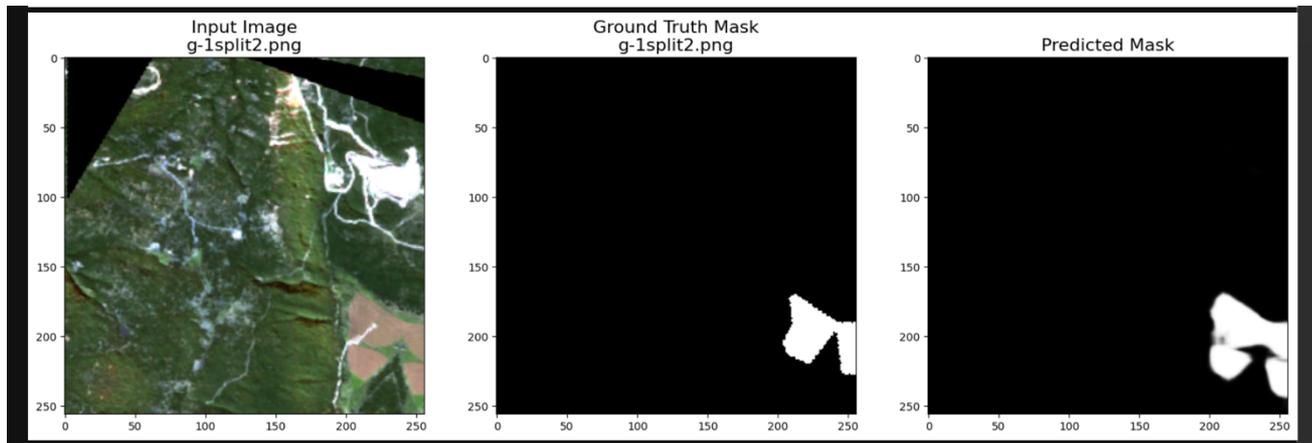


Ilustración 29. Resultado del modelo con dataset nuevo 1.

6.3.1 Ejemplos de Resultados en Imágenes Nuevas

Los resultados visuales demuestran que el modelo es capaz de identificar con precisión las áreas de plantas solares en las nuevas imágenes. Aunque los gráficos iniciales de pérdida y precisión sugerían un posible sobreajuste, la correcta identificación en estas imágenes no vistas confirma la capacidad del modelo para generalizar en nuevos conjuntos de datos, haciendo del modelo DeepLabV3+ una herramienta robusta para la segmentación de plantas solares.

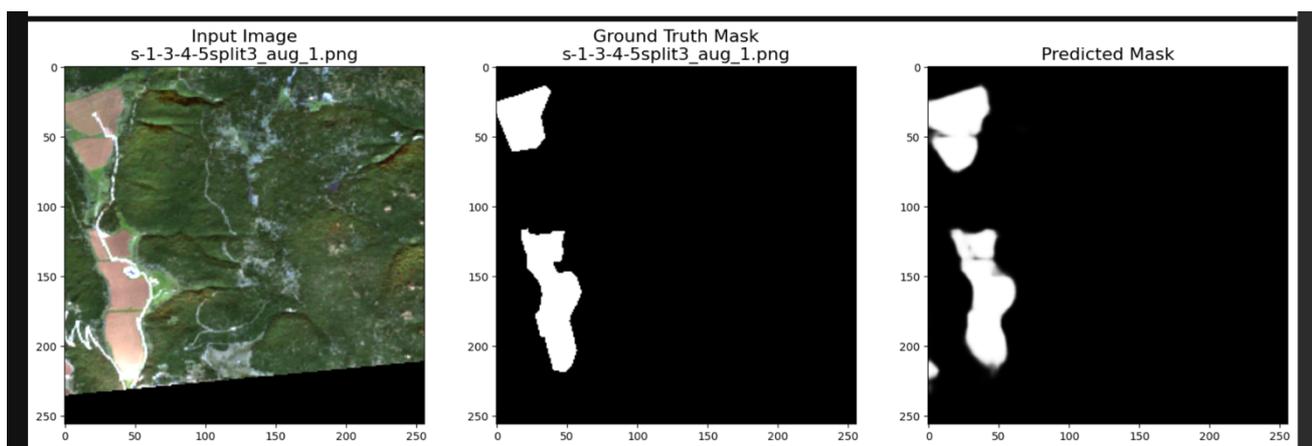


Ilustración 30. Resultado del modelo con dataset nuevo 2.

6.4 CONCLUSIÓN SOBRE EL RENDIMIENTO DEL MODELO

El modelo DeepLabV3+ ha mostrado un desempeño sólido en la segmentación de plantas solares, logrando resultados precisos y consistentes tanto en el conjunto de entrenamiento como en los datos no vistos. Los gráficos de precisión y pérdida iniciales plantearon preocupaciones sobre el sobreajuste, pero la validación en un conjunto de datos independiente validó su eficacia, sugiriendo que es capaz de generalizar adecuadamente a nuevos escenarios.

En conclusión, el modelo DeepLabV3+ proporciona una combinación óptima de precisión y robustez, siendo adecuado para su implementación en aplicaciones de segmentación de plantas solares basadas en imágenes satelitales.

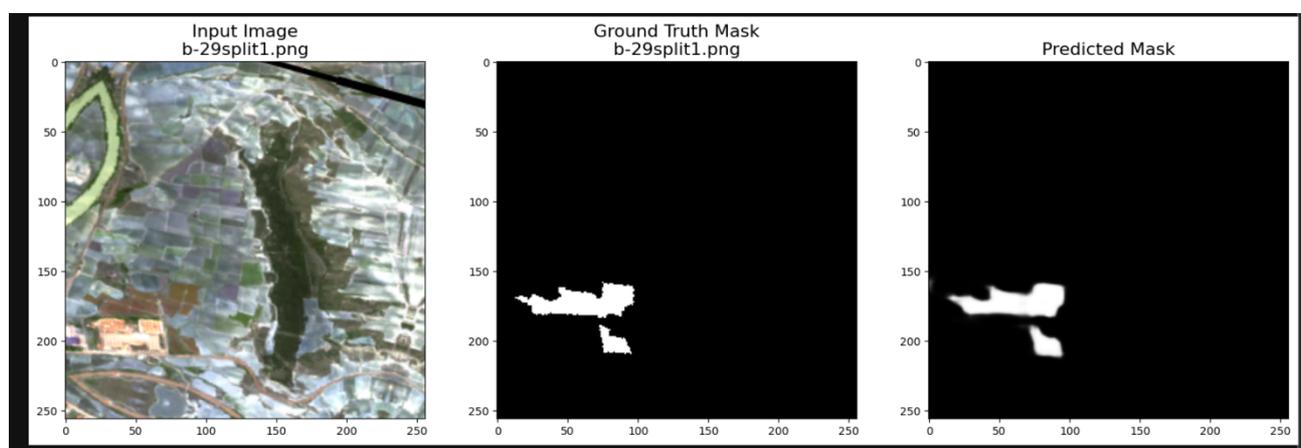


Ilustración 31. Resultado del modelo contra dataset de validación.



Capítulo 7 CONCLUSIONES Y TRABAJO FUTURO



7.1 CONCLUSIONES

Este proyecto se propuso desarrollar un modelo de segmentación capaz de identificar plantas solares en imágenes satelitales, lo que requirió un estudio profundo de técnicas de Visión por Computador y redes neuronales avanzadas. La creación del *dataset* fue un reto crucial, ya que implicó normalizar, segmentar y aplicar técnicas de preprocesamiento, como el recorte aleatorio y el aumento de datos, para mejorar la variedad y representatividad de los datos.

Tras probar varias arquitecturas, entre ellas InceptionResNetV2 y DeepLabV3+, seleccioné la última por su equilibrio entre precisión y eficiencia en la segmentación de objetos pequeños y dispersos, típicos de las plantas solares en imágenes satelitales. El modelo DeepLabV3+ demostró una gran capacidad de generalización, alcanzando una precisión alta en el conjunto de validación y logrando buenos resultados en imágenes nuevas y no vistas.

Este proyecto no solo logró los objetivos de segmentación propuestos, sino que también me permitió profundizar en el manejo de datos geoespaciales y en la importancia de un pipeline de preprocesamiento robusto para entrenar modelos eficaces. Los resultados obtenidos validan el enfoque y destacan el potencial de las redes neuronales para tareas de análisis de infraestructura en energía renovable a partir de imágenes satelitales.



7.2 TRABAJO FUTURO

Este proyecto abre la puerta a futuras mejoras que podrían enriquecer el modelo de segmentación de plantas solares y ampliar su capacidad en diversos contextos. Las posibles líneas de trabajo futuro incluyen:

- **Expansión del *dataset*:**

Ampliar el conjunto de datos con una mayor variedad de plantas solares, tanto en términos de ubicación geográfica como de características específicas. Esto permitiría al modelo adaptarse mejor a diferentes entornos y generalizar con mayor robustez en áreas nuevas.

- **Uso de imágenes monobanda en el Infrarrojo Cercano (NIR):**

Incorporar imágenes monobanda de infrarrojo cercano o NIR, podría mejorar la detección de plantas solares al diferenciar de manera más clara las superficies artificiales y la vegetación. Este enfoque es particularmente útil en paisajes complejos y puede aumentar la precisión del modelo.

- **Optimización del modelo:**

Explorar arquitecturas de redes neuronales más avanzadas y técnicas de ensamble para mejorar la precisión de los resultados. Además, se podrían integrar enfoques de aprendizaje semi-supervisado para aprovechar grandes volúmenes de datos no etiquetados, reduciendo la dependencia de datos etiquetados.

- **Automatización del pipeline:**

Desarrollar un pipeline más automatizado para las etapas de preprocesamiento y etiquetado de datos. Esto simplificaría la incorporación de nuevos datos, acelerando las actualizaciones del modelo y reduciendo el esfuerzo manual.

Estas mejoras no solo fortalecerían el modelo actual, sino que también ampliarían sus aplicaciones en el análisis y seguimiento de las plantas solares mediante imágenes satelitales, facilitando su implementación en una variedad de entornos y escalas.



Capítulo 8 MANUAL DE REPRODUCCIÓN DEL PROYECTO



8.1 INTRODUCCIÓN

Este manual describe los pasos necesarios para reproducir el proyecto, desde la configuración del entorno hasta la ejecución de los notebooks desarrollados en el lenguaje de programación python. Se incluyen detalles sobre las versiones de Python y las librerías utilizadas, además de recomendaciones para utilizar servidores en la nube.

8.2 REQUISITOS DEL PROYECTO

1. Versión de Python

Python: 3.10.13 (recomendada para compatibilidad con las librerías utilizadas).

2. Librerías Necesarias:

Se incluye un fichero requirements.txt con todas las librerías que tengo en mi entorno de desarrollo.

3. Hardware Recomendado

Tarjeta gráfica: NVIDIA T4 o equivalente, compatible con CUDA para entrenamiento acelerado.

Servidor recomendado: Google Compute Engine.

- Tipo de máquina: n1-highmem-8.
- GPU: 1 x NVIDIA T4.
- Disco: 80 GB SSD.



8.3 CONFIGURACIÓN DEL SERVIDOR

8.3.1 Google Compute Engine

Para reproducir el proyecto en un entorno controlado y optimizado, se recomienda utilizar Google Compute Engine debido a su facilidad de configuración y acceso a hardware avanzado.

El primer paso consiste en crear una instancia en Google Cloud con las siguientes especificaciones recomendadas:

Tipo de máquina: n1-highmem-8, que proporciona un equilibrio adecuado entre memoria y procesamiento para manejar grandes *datasets* y realizar operaciones complejas.

GPU: Una tarjeta NVIDIA T4, que es altamente eficiente para tareas de entrenamiento de modelos de aprendizaje profundo, especialmente con bibliotecas como TensorFlow y PyTorch.

Disco de almacenamiento: Un SSD de 100 GB para garantizar una lectura y escritura rápida de datos, indispensable al trabajar con grandes volúmenes de imágenes satelitales.

Una vez configurada la instancia, es necesario instalar las herramientas y librerías necesarias para el proyecto.

Un requisito fundamental es contar con Python en este caso yo he usado la **versión 3.10.13** y **JupyterLab**, que facilita la ejecución y edición de los notebooks del proyecto en un entorno interactivo.

Para ello vamos a hacer lo siguiente, primero de todo instalaremos la versión de Python:

Instalación de Python y creación de entorno virtual

```
sudo apt install python3.10
```

A continuación debemos de chequear si tenemos pip instalado usando el comando:

```
pip3 --version
```

en caso negativo deberíamos de hacer lo siguiente:

```
sudo apt install python3.10-distutils python3.10-venv
```

```
curl -sS https://bootstrap.pypa.io/get-pip.py | sudo python3.10
```

Ahora al consultar la versión de pip3 debería ya de aparecer, es recomendable usar un entorno virtual de Python, podemos crear uno de la siguiente manera:

```
python3.10 -m venv jupyter_env
```

```
source jupyter_env/bin/activate
```



Instalación y configuración de JupyterLab

Debemos de llevar a cabo en estos momentos la instalación de jupyterlab:

Pip install jupyterlab o pip3 install jupyterlab

Continuamos con la configuración como un servicio dentro de nuestro sistema, para ello debemos de ejecutar los siguientes comandos:

jupyter lab --generate-config → generamos un archivo de configuración en
~/jupyter/jupyter_lab_config.py

jupyter notebook password → para establecer una contraseña (seguir los pasos que indica).

nano ~/jupyter/jupyter_lab_config.py → hay que editar el fichero de configuración para habilitar el acceso remoto, en caso de no tener nano instalado debemos de ejecutar:

apt install nano

Y ahora hay que descomentar o añadir las siguientes líneas:

c.ServerApp.ip = '0.0.0.0'

c.ServerApp.port = 8888

c.ServerApp.open_browser = False

c.ServerApp.allow_remote_access = True

Permitir el tráfico en el puerto 8888

Debemos de acceder a JupyterLab desde el navegador por ello tenemos que permitir el tráfico en el puerto 8888 desde Google Cloud:

Ve a la consola de Google Cloud.

Ve a VPC network > Firewall rules.

Crea una nueva regla de firewall:

Nombre: algo como allow-jupyter.

Objetivo: Todas las instancias en la red.

Filtros de IP: 0.0.0.0/0 (para permitir acceso global, o restringe a tu IP específica).

Protocolo y puerto: TCP:8888.



Iniciar Servicio de JupyterLab

Tocaría iniciar el servicio anteriormente configurado en nuestro sistema, para ello simplemente tenemos que ejecutar

```
jupyter-lab --ip 0.0.0.0 --port 8888 --no-browser
```

Esto haría que si introducimos “http://<IP-de-tu-servidor>:8888”, en la barra de nuestro navegador, si todo ha ido bien nos abra el servicio. (Es posible que la primera vez solicite una contraseña que se debe de configurar, es seguir los pasos que se muestran en la terminal).

Tras instalar JupyterLab, se deben copiar al servidor todos los archivos necesarios, incluyendo los notebooks y el archivo requirements.txt, que contiene las librerías necesarias para la ejecución del proyecto. Estas librerías incluyen geopandas, rasterio, torch, tensorflow, entre otras, asegurando que todas las dependencias estén correctamente instaladas.

pip install -r requirements.txt → para instalar todas las librerías con las que he trabajado.

8.3.2 Google Colab

Google Colab se presenta como una solución accesible y gratuita para ejecutar los notebooks del proyecto sin necesidad de instalar y configurar un servidor dedicado. Esta herramienta, basada en la nube, resulta ideal para desarrolladores que buscan una alternativa rápida y sencilla para trabajar con entornos de aprendizaje profundo.

Para comenzar, los notebooks del proyecto se pueden cargar directamente en la plataforma desde tu ordenador, Google Drive o un repositorio en GitHub. Una vez cargados, Colab ofrece la posibilidad de habilitar una GPU para aprovechar la aceleración de hardware necesaria para tareas intensivas como el entrenamiento de redes neuronales. Esta configuración es muy sencilla: basta con ir al menú "Entorno de ejecución", seleccionar "Cambiar tipo de entorno de ejecución" y activar el uso de GPU.

En cuanto a las dependencias del proyecto, Colab permite instalar todas las librerías requeridas mediante el archivo requirements.txt. Este archivo contiene un listado de todas las versiones específicas de las bibliotecas necesarias, garantizando la compatibilidad del entorno. La instalación se realiza directamente desde una celda del notebook con un simple comando:

```
!pip install -r requirements.txt
```



8.4 NOTAS FINALES

Aunque el diseño original del proyecto contempla su ejecución en un servidor especializado, la flexibilidad de las herramientas y metodologías utilizadas permite adaptarlo para su implementación en entornos locales. En particular, aquellos usuarios que cuenten con tarjetas gráficas compatibles y recursos computacionales suficientes podrán reproducir el proyecto en sus propios equipos. Sin embargo, es importante tener en cuenta que esto podría requerir una configuración más personalizada y ajustes específicos en las librerías y dependencias del entorno.

Para aquellos que no disponen de acceso a servidores o hardware dedicado, Google Colab se posiciona como una excelente alternativa. Esta plataforma basada en la nube permite ejecutar los notebooks del proyecto de manera gratuita, proporcionando acceso a GPUs que facilitan el entrenamiento y la ejecución de modelos de aprendizaje profundo.

Este proyecto contiene todos los recursos necesarios para reproducirse, incluyendo el archivo `requirements.txt` que contiene un listado completo de las dependencias utilizadas, con sus versiones correspondientes. Este archivo, junto con los notebooks del proyecto, asegura la replicabilidad del entorno y facilita el proceso de instalación, ya sea en un servidor, equipo local o a través de plataformas como Colab.

En resumen, este proyecto ha sido diseñado para ser versátil y adaptable, permitiendo a cualquier usuario, independientemente de sus recursos o experiencia, explorar y aprovechar las técnicas desarrolladas para la segmentación de imágenes satelitales.

Capítulo 9 BIBLIOGRAFÍA





[

- [1] Glassdoor, «Glassdoor,» 1 11 2024. [En línea]. Available: <https://www.glassdoor.es/>.
- [2] H. A. Claudio Sebastian, «UNIVERSIDAD TECNICA FEDERICO SANTA MARIA,» 08 2022. [En línea]. Available: <https://repositorio.usm.cl/server/api/core/bitstreams/966e0ebb-000f-401f-8672-10bfc4ecb03e/content>.
- [3] H. d. V. C. D. ,. G. B. Cristian Pacheco, «Universidad Nacional del Sur (UNS),» 2 2014. [En línea]. Available: https://sedici.unlp.edu.ar/bitstream/handle/10915/41255/Documento_completo.pdf?sequence=1.
- [4] L. Kahana, «pv magazine,» 4 octubre 2023. [En línea]. Available: <https://www.pv-magazine.es/2023/10/04/imagenes-por-satelite-de-alta-resolucion-para-detectar-sistemas-fotovoltaicos-residenciales/>.
- [5] N. M. R. Quintana, «Universidad Nacional de Colombia Facultad de Ciencias Agrarias,» 2024. [En línea]. Available: <https://repositorio.unal.edu.co/bitstream/handle/unal/86362/1057579801.2024.pdf?sequence=2>.
- [6] L. Kahana, «pv magazine,» 14 agosto 2023. [En línea]. Available: <https://www.pv-magazine-latam.com/2023/08/14/una-tecnologia-basada-en-ia-para-identificar-sistemas-solares-en-tejados-a-partir-de-imagenes-aereas/>.
- [7] O. Contributors, «Overpass-Turbo,» 2024. [En línea]. Available: <https://overpass-turbo.eu/>.
- [8] g. Developers, «geojson.io,» 2024. [En línea]. Available: <https://geojson.io/#map=2/0/20>.
- [9] J. M. Redondo, «Documentos-modelo para Trabajos de Fin de Grado/Master de la Escuela de Informática de Oviedo,» 17 6 2019. [En línea]. Available: https://www.researchgate.net/publication/327882831_Plantilla_de_Proyectos_de_Fin_de_Carrera_de_la_Escuela_de_Informatica_de_Oviedo.
- [10] R. C. & W. R. E. Gonzalez, «Digital Image Processing (4th Edition),» de *Digital Image Processing (4th Edition)*, Pearson, 2018, p. Capítulo 10.5.
- [11] J. Canny, «IEEEExplore,» 27 1 2009. [En línea]. Available: <https://ieeexplore.ieee.org/document/4767851>.
- [12] L. G. S. Robert M. Haralick, «sciencedirect.com,» 26 5 1982. [En línea]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0734189X85901537?via%3Dihub>.



- [13] P. F. & T. B. Olaf Ronneberger, «Springer Nature Link,» 18 11 2015. [En línea]. Available: https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28.
- [14] V. Badrinarayanan, A. Kendall y R. Cipolla, «IEEEExplore,» 2 1 2017. [En línea]. Available: <https://ieeexplore.ieee.org/document/7803544>.
- [15] Y. Z. G. P. F. S. & H. A. Liang-Chieh Chen, «Springer Nature Link,» 6 10 2018. [En línea]. Available: https://link.springer.com/chapter/10.1007/978-3-030-01234-2_49.
- [16] S. J. Pan y Q. Yang, «IEEEExplore,» 16 10 2009. [En línea]. Available: <https://ieeexplore.ieee.org/document/5288526>.



Capítulo 10 ANEXOS

10.1 PLAN DE GESTIÓN DE RIESGOS

Para este proyecto, se ha decidido adoptar el **Plan de Gestión de Riesgos** basado en la **metodología de Boehm**, que se centra en un enfoque estructurado para identificar, analizar, planificar y monitorear los riesgos del proyecto. Esta metodología es particularmente efectiva para proyectos de desarrollo, como el presente trabajo, al proporcionar una base sólida para la gestión proactiva de los riesgos potenciales que puedan surgir.

10.1.1 Metodología de Gestión de Riesgos de Boehm

1. Identificación de Riesgos

En primer lugar, se han identificado los principales riesgos asociados con el desarrollo de este proyecto, los cuales han sido documentados y detallados previamente en [Identificación de riesgos](#). Cada riesgo se relaciona con aspectos específicos del proyecto, como la obtención de datos, la capacidad de procesamiento o la precisión de los modelos desarrollados. Esta fase es fundamental, ya que permite reconocer de forma temprana los posibles problemas que podrían afectar al desarrollo y al resultado del proyecto.

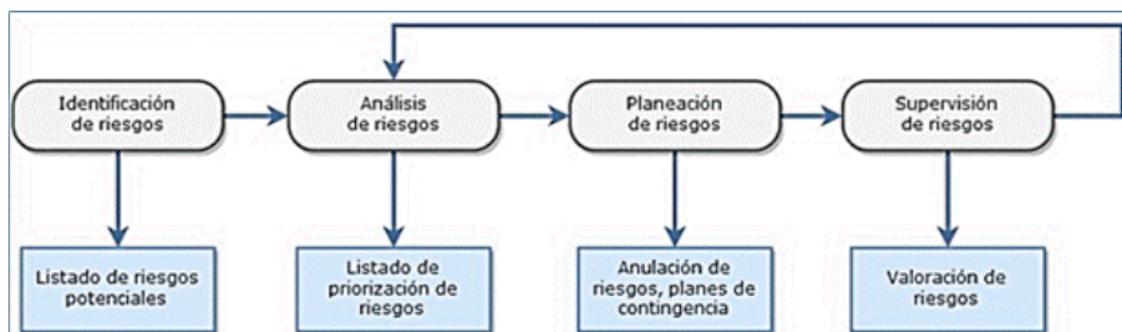


Ilustración 32. Metodología Boehm.



2. Análisis de Riesgos

Una vez identificados los riesgos, se realiza un análisis exhaustivo para evaluar la **probabilidad de ocurrencia** y el **impacto potencial** de cada uno. La evaluación sigue una escala cualitativa en la que se clasifican los riesgos según su nivel de criticidad:

- **Alto:** Riesgos con alta probabilidad de comprometer significativamente el desarrollo del proyecto o la calidad de los resultados.
- **Medio:** Riesgos con un impacto moderado que pueden afectar al cronograma o al presupuesto.
- **Bajo:** Riesgos menores que, aunque posibles, no representan una amenaza significativa para el proyecto.

Este análisis permite establecer una priorización de riesgos, de forma que se destinen mayores esfuerzos y recursos a aquellos riesgos que presentan una mayor amenaza.

3. Planificación de Respuestas a los Riesgos

Con la lista priorizada de riesgos, se planifican estrategias de respuesta específicas para mitigar su impacto. Entre las estrategias seleccionadas se encuentran:

- **Evitar:** En algunos casos, se optará por modificar ciertos aspectos del proyecto para eliminar el riesgo.
- **Mitigar:** En la mayoría de los riesgos críticos, se han planteado acciones preventivas para reducir su impacto o probabilidad.
- **Aceptar:** Para ciertos riesgos menores, se reconoce su posibilidad, pero no se toman medidas adicionales, ya que el impacto es limitado y no justifica una intervención.
- **Transferir:** Cuando sea viable, algunos riesgos se delegan a terceros, especialmente aquellos relacionados con la infraestructura en la nube o recursos de datos externos.

Cada riesgo cuenta con un responsable específico dentro del equipo que se encargará de supervisar y ejecutar la estrategia de respuesta planificada.

4. Seguimiento y Control de Riesgos

La metodología de Boehm establece que los riesgos deben ser revisados de forma continua a lo largo del proyecto. Por ello, se ha implementado un sistema de **revisión periódica**, con evaluaciones semanales para:

- Comprobar el estado actual de cada riesgo.
- Evaluar la efectividad de las respuestas implementadas.
- Realizar ajustes en las estrategias según sea necesario.

Además, se prevé realizar una **reevaluación** de los riesgos en puntos clave del proyecto para adaptar la gestión a las circunstancias cambiantes, y se documentará cualquier cambio en la estrategia o el estado de los riesgos en el **registro de riesgos** del proyecto.

Ejemplo de Gestión de Riesgos

La siguiente tabla presenta ejemplos específicos de riesgos y las estrategias de respuesta desarrolladas para cada uno:

Riesgo	Probabilidad	Impacto	Estrategia	Responsable	Acción
Retrasos en la obtención de <i>datasets</i>	Alta	Alta	Mitigar	Investigador	Disponer de varias fuentes de <i>datasets</i> alternativas con licencias adecuadas
Calidad insuficiente del <i>dataset</i>	Media	Alta	Mitigar	Validador de <i>Dataset</i>	Aplicar un filtrado y limpieza inicial del <i>dataset</i> para asegurar su calidad
Capacidad de procesamiento insuficiente	Baja	Alta	Evitar	Administrador de Servidores	Usar infraestructura en la nube que permita escalabilidad en procesamiento
Baja precisión del modelo	Media	Media	Mitigar	Programador	Ajustar hiperparámetros y realizar pruebas adicionales de validación

Tabla 32. Ejemplo: Gestión de Riesgos.

Esta implementación del Plan de Gestión de Riesgos, basada en la metodología de Boehm, asegura un enfoque ordenado y eficaz para minimizar el impacto de los riesgos en el desarrollo del proyecto, mejorando la capacidad de respuesta ante posibles dificultades y optimizando el flujo de trabajo del equipo.