



UNIVERSIDAD DE OVIEDO
ESCUELA POLITÉCNICA DE MIERES

MÁSTER UNIVERSITARIO EN GEOTECNOLOGÍA Y DESARROLLO DE PROYECTOS SIG

TRABAJO FIN DE MÁSTER

Proceso para la Detección Automática de
Cortaderia selloana usando Imágenes
Sentinel-2 y Redes Neuronales Artificiales

AUTOR:

José María Uribarri Landa

TUTORAS:

Ana Belén Ramos Guajardo

Pilar García Manteca

Gijón
14 de julio de 2022

AGRADECIMIENTOS

Ana y Pilar, mis tutoras, gracias por guiarme y por dejarme hacer con esta idea de utilidad incierta. Gracias por arriesgar vuestro tiempo sin tener la seguridad de saber que iba a resultar de provecho. Espero haber cumplido con la confianza que me habéis brindado.

Grazas, Xesús Pablo, por teres crido en min antes do que eu mesmo. Sempre obrigado por me empurrar neste camiño.

Olaya, vuestro esfuerzo diario dándome ese espacio que es tan difícil sacar en un piso tan pequeño y por poner el tiempo que no he podido estar yo este año. Grazas a ti tamén, Gaizka, meu pequecho, imos recuperar o tempo que non tivemos estes meses. Quérovos.

Ama, Aita, vuestra paciencia, amor y enseñanzas son el barco en el que navego. No podría sentirme más en deuda con nadie, porque, al fin y al cabo, nadie me ha regalado nunca un barco.

A ti, que tanto me faltas, que sólo me oyes dentro de mi cabeza y mientras escribo esto me dices: "¿qué haces?". Sin ti esto nunca habría sucedido. Te prometo disfrutar de la ornitología todo lo que te habría gustado hacer a ti.

Resumen

El presente trabajo desarrolla una metodología completa para obtener una clasificación de lugares con una alta probabilidad de presencia de *Cortaderia selloana*, desde la manipulación de las imágenes obtenidas del satélite Sentinel-2 hasta la conversión de las salidas en archivos ráster válidos para su carga en un software GIS. Para lograrlo, se recurrirá al Deep Learning, configurando y entrenando una red neuronal con capas densamente conectadas, que se entrenará de forma iterativa, hasta conseguir unos resultados que mejoran los obtenidos hasta ahora por medio de la teledetección clásica, empleando muchos menos recursos. La zona de estudio está situada en el norte de la región central de Asturias, donde se han alcanzado precisiones por encima del 80 % de los píxeles que han podido ser comprobados para zonas con una superficie identificada de 1.000 m^2 en adelante.

Abstract

The present work develops a complete methodology to obtain a classification of places with a high probability of having presence of *Cortaderia selloana*, starting with the manipulation of the images obtained from the Sentinel-2 satellite, to the conversion of the outputs into valid raster files for loading into GIS software. To achieve this, Deep Learning will be used, configuring and training a neural network of fully connected layers, which will be trained iteratively, until obtaining results that improve those obtained so far through classical remote sensing, using much fewer resources. The study area is located in the north of the central region of Asturias, where precision has been achieved above 80% of the pixels that have been verified for areas with an identified area of 1,000 m^2 and more.

Índice

1	Introducción	4
2	Materiales y trabajo previo	7
2.1	Área de estudio y elección de sensor	7
2.2	Adquisición de Imágenes	8
2.3	Preparación del entorno	10
3	Metodología y procedimientos	11
3.1	Pre-procesado de imágenes y creación de los conjuntos de datos	11
3.2	Configuración de la Red Neuronal Artificial	13
3.3	Optimización de los parámetros de la red	15
3.4	Resultados preliminares y proceso iterativo de clasificación	16
3.5	Validación de resultados	16
3.6	Obtención de las salidas gráficas y comprobación visual de resultados	17
4	Resultados	19
4.1	Resultados numéricos	19
4.1.1	Estadísticas propias del modelo	19
4.1.2	Estadísticas elaboradas tras la comprobación visual de resultados	20
4.2	Detalles de salidas gráficas	22
5	Conclusiones	25
6	Referencias	27
A	Anexo I	
B	Anexo II	

Índice de figuras

1	Comparativa de mapa de densidad de Cortaderia seollanda (arriba, Autor: LIFE Stop Cortaderia) con otro de densidad de población (abajo, Autor: Alter Geosistemas)	4
2	Distribución de modelos de Deep Learning usados en estudios Autor: Ma y col. 2019	5
3	Distribución irregular de <i>C. selloana</i> (izquierda) frente a la distribución regular de los cultivos (derecha). Fuente: IGN	6
4	Área de Estudio. Fuente: IGN. Autor: Elaboración propia.	7
5	Zona de estudio (naranja), encuadrada dentro de la zona de solape de ambos satélites (verde más oscuro). Fuente: ESA. Autor: Elaboración propia.	8
6	Fenología de <i>C. selloana</i> en Bizkaia. G: Germinación; CV: Crecimiento vegetativo; FL: producción de flores; FR: producción de frutos; SD: dispersión de semillas; SN: senescencia de hojas. Las líneas de puntos delimitan el periodo de actividad vegetal. Fuente: Herrera y Campos 2006. Autor: Ministerio para la Transición Ecológica.	9
7	Banda 2 (Azul) del Sentinel-2 en la imagen de 13/10/2021 (Escena completa). Fuente: E.S.A.	9
8	Detalle de píxeles identificados como libres de Cortaderia (verde) y otros con presencia por la planta (rojo). Autor: Elaboración propia	12
9	Diagrama de capas de la red neuronal densamente conectada. Autor: Elaboración propia .	13
10	Función PReLU. Autor: Programador de Click.	14
11	Función sigmoide. Autor: Programador de Click.	14
12	Flujo de una neurona. Autor: Rodríguez 2018	15
13	A la izquierda: clasificación inicial incorrecta; en el centro: clasificación manual correcta de la zona; a la derecha: identificación correcta realizada por la red tras un nuevo entrenamiento. 16	16
14	10-fold cross validation, donde cada subconjunto contiene aproximadamente el 10 % de los datos. Autor: Berrar 2018	17
15	Gráfico que representa el valor de la estadística de precisión (izquierda) y de coste (derecha) en el eje Y en cada repetición representada en el eje X. Autor: Elaboración propia.	19
16	A la izquierda la zona con más píxeles identificados como <i>C. selloana</i> en acantilados (probablemente incorrectos); a la derecha gran extensión de costa donde aparece la única zona identificada sobre un acantilado (probablemente correcta). Autor: Elaboración propia. 22	22
17	Detalle de la clasificación donde se ve correctamente clasificada una zona donde se encuentra <i>C. selloana</i> en sitintos contextos, mezclada con <i>Ulex europaeus</i> , otras con flores y zonas de plantas jóvenes que no han florecido todavía.	23
18	A la izquierda se puede ver una zona salpicada con píxeles sueltos (probablemente incorrectos). En el centro, tras aplicar el filtro, sólo queda una gran zona (correcta), ampliada en la imagen de la derecha. Autor: Elaboración propia.	23
19	Detalle de zona industrial con un grupo de plantas identificado en campo, que obtiene un resultado que para un umbral de 0.9 o sin aplicar el filtro, sería correctamente idetificada, pero con esta doble restricción, es incorrectamente descartada. Autor: Elaboración propia. 24	24
20	Detalle de zona industrial intensamente afectada. A la izquierda: clasificación para niveles entre 0.09-0.95 (amarillo), 0.95-0.99 (naranja) y >0.99 (rojo). A la derecha: zonas seleccionadas tras aplicar sólo nivel >0.99 y el filtro para zonas de 500 m ² o más. Autor: Elaboración propia.	24
21	Problema con la identificación de los acantilados. Autor: INVASAT	26

Índice de tablas

1	Número de píxeles identificados y asignados a los conjuntos conjuntos de entrenamiento y test. Fuente: GIS&Beers	8
2	Número de píxeles identificados y asignados a los conjuntos conjuntos de entrenamiento y test.	12
3	Precisión de las mejores combinaciones de parámetros.	16
4	Precisión media para cada fold.	17
5	Principales estadísticas extraídas del proceso de validación cruzada.	17
6	Matriz de confusión que muestra en columnas el valor	20
7	Conteo y porcentaje de píxeles según el resultado de la comprobación visual para la muestra aleatoria extraída de los grupos de positivos con área inferior a $500m^2$	20
8	Conteo y porcentaje de píxeles según el resultado de la comprobación visual para los grupos de positivos con área superior a $500m^2$	20
9	Sumario estadístico de las entidades y su área según el resultado de la comprobación visual para polígonos con área superior a $500 m^2$	21
10	Sumario estadístico de las entidades y su área según el resultado de la comprobación visual para polígonos con área comprendida entre $500 m^2$ y $1000 m^2$	21
11	Sumario estadístico de las entidades y su área según el resultado de la comprobación visual para polígonos con área superior a $1000 m^2$	22

1. Introducción

Las especies exóticas invasoras se han convertido en una de las principales causas de pérdida de biodiversidad en el mundo (Ministerio de Agricultura, Alimentación y Medio Ambiente 2013) y en los últimos años se ha visto cómo especies tanto animales (*Vespa velutina*, *Ostrya japónica*, *Aedes albopictus*,...) como vegetales (*Acacia*, *Baccharis halimifolia*, *Cortaderia selloana*,...) están ganando terreno y desplazando a flora y fauna local, produciendo daños económicos, especialmente a la producción agrícola, ganadera y forestal, e incluso a la salud pública (Ministerio de Agricultura, Alimentación y Medio Ambiente 2013).

La actividad humana alrededor del mundo ha sido el agente de mayor importancia en la reciente introducción de un largo número de especies que han derivado más tarde en invasiones biológicas (Saurama y Lloret 2005). La especie *Cortaderia selloana* (*C. selloana* de ahora en adelante), también conocida como Plumero de la Pampa, es una de las que más he aumentado su expansión en todo el norte peninsular, especialmente en las zonas cercanas a las costas cantábrica y atlántica.

La *C. selloana*, como otras especies exóticas invasoras, demuestra gran capacidad para adaptarse a entornos alterados por la actividad humana (ver Figura 1), lo que significa que la construcción civil de zonas de edificios, los alrededores de las carreteras y otros corredores artificiales son zonas óptimas para el establecimiento de estas especies (Rentch y col. 2005). Aunque tiene facilidad para invadir terrenos intervenidos como taludes, cunetas o parcelas libres en polígonos industriales, también ocupa espacios naturales como bordes de marismas, dunas o bordes de bosques y se alerta de su avance por áreas hasta ahora consideradas inadecuadas para esta especie, como zonas de montaña, territorios al sur de la Cordillera Cantábrica e incluso por el centro peninsular (Ministerio para la Transición Ecológica 2018).

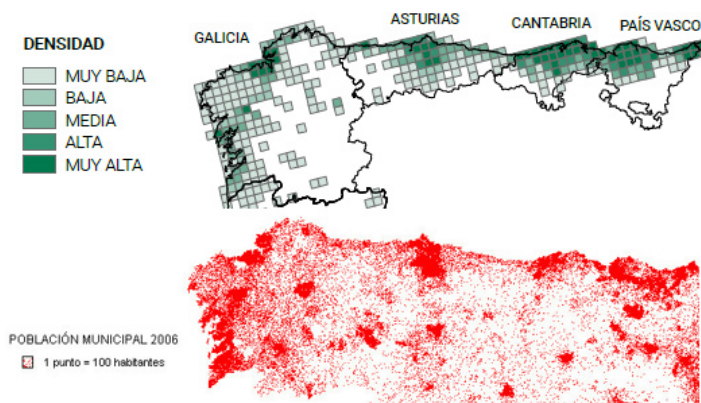


Figura 1: Comparativa de mapa de densidad de *Cortaderia seollanda* (arriba, Autor: LIFE Stop Cortaderia) con otro de densidad de población (abajo, Autor: Alter Geosistemas)

La expansión de la *C. selloana* despierta una gran preocupación en las administraciones tanto a nivel estatal como europeo, lo que se traduce en la financiación de proyectos tales como el Proyecto LIFE Stop Cortaderia (proyecto Life+ de la UE con base en Cantabria, aunque ya se extiende con colaboraciones entidades de Galicia y Portugal) o la publicación en 2018 de la Estrategia de gestión, control y posible erradicación del plumero de la Pampa y otras especies de Cortaderia por parte del Ministerio para la Transición Ecológica.

Son una constante en los proyectos y planes para el control de la Cortaderia las referencias a la detección temprana como herramienta fundamental del control de la expansión (habiendo desarrollado LIFE Stop Cortaderia una herramienta de ciencia ciudadana para su identificación) y, tanto en el Proyecto Life Stop Cortaderia, como en la estrategia publicada por el Ministerio para la Transición Ecológica, se incide en la importancia de contar con una cartografía actualizada de la distribución de la planta, pero

a pesar de ello en esta última se destaca que esto es algo que no ocurre en todas las Comunidades Autónomas.

En los últimos años empiezan a estar a disposición pública imágenes de satélite con resoluciones espectrales, espaciales y temporales muy interesantes que podrían ayudar a resolver el problema de la dificultad y el coste de obtener y actualizar una cartografía de este tipo de eventos. Es por ello que se ha multiplicado su uso al ser una fuente de datos relativamente rápida y eficiente de adquisición e interpretación de firmas espectrales en zonas de interés relativamente grandes (Adiri y col. 2020). Poder llevar a cabo una identificación automática de la presencia de la planta en cada pixel, repercutiría notablemente en la capacidad de tener una base de datos cartográfica actualizada anualmente, reduciendo sensiblemente la inversión necesaria para ello.

Existen muchas plataformas de satélites que recogen imágenes digitales en diferentes resoluciones, en variados espectros lumínicos, separadas por intervalos de tiempo distintos y con variedad de precios. Las imágenes de dos de estas plataformas han sido valoradas para ser utilizadas en este trabajo: las de la misión Sentinel-2 de la Agencia Espacial Europea (E.S.A. según sus siglas en inglés) y la de la National Aeronautics and Space Administration (N.A.S.A.). Sentinel-2 ofrece imágenes con una resolución espectral de 12 bandas con longitudes de onda que oscilan entre los 443 nm de la banda B01 de aerosol costero hasta los 2190 nm de la B12 de onda corta infrarrojo 2 y todo ello con un tiempo de revisita de 5 días, que permite aumentar la probabilidad de obtener escenas libres de nubes en un lugar tan nuboso como es Asturias. Si se compara con Landsat 8, se puede observar que este cuenta con una resolución espectral que llega a longitudes de onda mucho mayores en sus bandas 10 y 11 (hasta 12510 nm en la 11), pero la resolución espacial de estas bandas sería de 100 metros. Además, la resolución del resto de bandas es de “tan solo” 30 metros, salvo para el pancromático que sería de 15 metros. La mayor resolución espacial de Sentinel-2 ofrece la capacidad de poder identificar píxeles con una superficie menor ocupada por el elemento a identificar y su resolución temporal ayudará a obtener imágenes sin nubes de más fechas que los 16 días que pasarán entre cada vuelo de Landsat 8 sobre nuestra zona de estudio.

El desarrollo informático de los últimos tiempos ha posibilitado que se hayan puesto al alcance de la comunidad capacidades de computación que permiten la implementación de cálculos complejos en ordenadores de uso doméstico y esto ha permitido que los métodos de Machine Learning hayan experimentado un gran impulso en su uso y su desarrollo. Habiendo sido usados con éxito en la clasificación de cultivos, los algoritmos convencionales de Machine Learning tales como support vector machine (SVM), random forest (RF), K-nearest neighbours (KNN), y árboles de decisión (DT) (Singha y col. 2016, Guo y col. 2017) no son tan efectivos en teledetección en comparación con las técnicas de Deep Learning (Rauf y col. 2022).

La mayor parte de estudios y aproximaciones a la clasificación de píxeles en imágenes satelitales por medio de técnicas de Deep Learning (ver Figura 2)) se han centrado en el uso de convolutional neural networks (CNN) (Ma y col. 2019), que son redes especializadas en la detección de objetos dadas relaciones entre los píxeles.

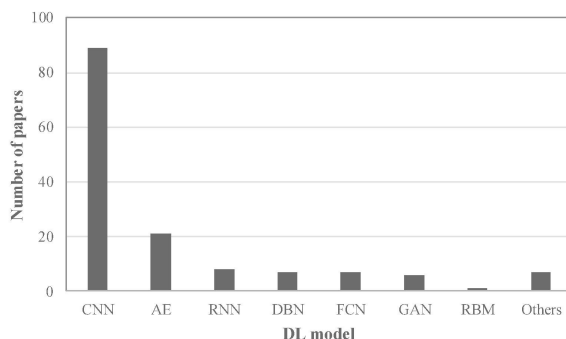


Figura 2: Distribución de modelos de Deep Learning usados en estudios Autor: Ma y col. 2019

Se propone que una clasificación píxel a píxel que tenga sólo en cuenta las reflectividades sobre cada píxel y su variación a lo largo del año, sin importar los valores de aquellos colindantes, puede ser adecuada para la detección de pequeños grupos o hasta de individuos de gran tamaño que hayan crecido de forma aislada. Recientes trabajos han concluido que se pueden obtener resultados ligeramente mejores (al menos un 3.5 %) con este tipo de técnicas frente a otros métodos de Machine Learning en la clasificación de cultivos (Trujillo-Jiménez y col. 2022). Teniendo en cuenta que los cultivos se suelen producir en parcelas uniformes, surge el pensamiento de que este tipo de técnicas pueden mejorar todavía más lo logrado por las técnicas de Machine Learning en identificación de fenómenos de naturaleza irregular y que puedan surgir de forma aislada, como pueden ser las agrupaciones de plantas invasoras de *C. selloana* (ver Figura 3).

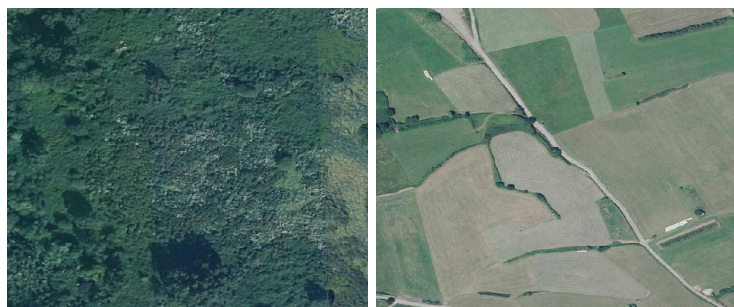


Figura 3: Distribución irregular de *C. selloana* (izquierda) frente a la distribución regular de los cultivos (derecha). Fuente: IGN

Para el desarrollo de este trabajo se parte de otra experiencia previa donde se planteó la utilidad de este tipo de redes para la clasificación basada en el píxel para la detección automática de asfalto con imágenes del Sentinel-2 (Uribarri 2020). Este tipo de redes es utilizado para realizar predicciones sobre el comportamiento de individuos, como la posibilidad de que se cancele una reserva en un restaurante partiendo de parámetros de la reserva y del cliente (Huang y col. 2013). En este trabajo se plantea que si se pueden hacer predicciones con éxito sobre el comportamiento futuro de personas, partiendo de datos de comportamiento pasado parametrizando atributos de personas, debería ser posible usar los atributos de un píxel para predecir su pertenencia o no a cierta clase.

El proceso consistirá en crear una matriz de características para todos los individuos (píxeles), que están identificados por las coordenadas X e Y, separando estas de sus atributos y analizando un subconjunto de datos, que han sido identificados como aquellos con presencia o no de *C. selloana*, y al que se añadirá dicha clasificación como un vector de clasificación con 0 y 1 como valores. Tras el aprendizaje de la red neuronal esta creará un vector de clasificación para todos los píxeles, que permitirá crear una nueva banda que formará la imagen clasificada de la zona de estudio.

El presente trabajo tiene como objetivo inicial definir una metodología que permita manipular ágilmente la gran cantidad de datos necesarios para el análisis multi-espectral y multi-temporal de grandes extensiones con técnicas de Machine Learning. El segundo objetivo es el de sondear la potencialidad de la técnica de Deep Learning de Redes Neuronales Densamente Conectadas para observar fenómenos que solamente ocupan una porción de un determinado píxel, pero provocan pequeñas variaciones en la reflectividad a lo largo del tiempo y que son difíciles de detectar por medio de las técnicas clásicas de teledetección.

En las siguientes secciones se describirán todos los procedimientos y tareas que se han realizado con el objetivo de obtener la herramienta y evaluar la eficacia de la misma en la tarea de identificación de la especie exótica invasora *C. seollana* en Asturias. Desde la selección de la zona a estudiar y las fuentes de los datos, siguiendo por la adquisición de las imágenes y la preparación del software, como trabajos preparatorios. También se describirán los métodos y procedimientos que se han seguido para la configuración de la red, la manipulación de los datos de las imágenes, la evaluación de la red y su posterior uso para lograr unas salidas que deben ser evaluadas e interpretadas.

2. Materiales y trabajo previo

Durante esta sección se expondrán las tareas realizadas con anterioridad al trabajo que será realizado mediante la manipulación de datos. Dichas tareas incluyen la selección de la zona a estudiar, la selección del satélite del que se obtendrán las imágenes y la preparación de los equipos en los que se acondicionarán los entornos de desarrollo y se instalará el software necesario para poder llevar a cabo las tareas pertinentes.

2.1. Área de estudio y elección de sensor

El área de estudio elegida se encuentra en las proximidades franja costera de la zona central de Asturias y en ella se encuadran total o parcialmente los municipios de Avilés, Gozón, Carreño, Corvera, Llanera y Gijón (ver Figura 4). Esta zona fue elegida por la facilidad que le suponía al autor para identificar in situ zonas invadidas y realizar algunas comprobaciones en campo, además de presentar una variedad de superficies con zonas rurales, industriales, acantilados, playas, cultivos, monte bajo, etc, y todo ello salpicado por zonas invadidas por *C. selloana*, que se distribuye de forma heterogénea, a veces en grandes extensiones creando cubiertas regulares y otras salpicado con otros elementos o especies vegetales.



Figura 4: Área de Estudio. Fuente: IGN. Autor: Elaboración propia.

Si bien se trataron de localizar un buen número de plantas sobre el terreno, para obtener un conjunto de datos lo suficientemente grande que forme un buen conjunto de entrenamiento hubo que recurrir a otras fuentes de datos. Se pudieron consultar archivos vectoriales del plan de control del Principado de Asturias en colaboración con Tragsa y la Universidad de Oviedo, aunque estos datos tampoco sirvieron para georreferenciar fehacientemente lugares invadidos o ejemplares individuales, ya que muchos puntos contaban con una localización que no era lo suficientemente precisa como para poder mantener el punto que representa la planta dentro del píxel en el que ésta se ve encuadrada en la imagen Sentinel-2. Finalmente, la mayor parte de las plantas fueron localizadas por medio de su identificación directa sobre ortofotos del Plan Nacional de Ortofografía Aérea (PNOA) en su vuelo de 2021, cargadas en el software QGIS 3.22 a través del servicio WMTS del Instituto Geográfico Nacional.

El sensor elegido es el que va equipado en los satélites Sentinel-2 de la misión Copernicus de la Agencia Espacial Europea (E.S.A.). La decisión se tomó teniendo en cuenta cuatro pilares fundamentales, que son: las resoluciones espacial, radiométrica y temporal, así como el acceso libre a las imágenes publicadas.

Sentinel-2 ofrece 12 bandas (ver Tabla 1) con resoluciones espaciales que oscilan entre los 10 y los 60 metros, teniendo tres bandas específicas para propósitos de análisis de vegetación (Red Edge Infra-Red).

Bandas	Longitud de onda (μm)	Resolución (m)
Banda 1 - Aerosol	0.43 - 0.45	60
Banda 2 - Blue	0.45 - 0.52	10
Banda 3 - Green	0.54 - 0.57	10
Banda 4 - Red	0.65 - 0.68	10
Banda 5 - Red edge 2	0.69 - 0.71	20
Banda 6 - Red edge 3	0.73 - 0.74	20
Banda 7 - Red edge 3	0.77 - 0.79	20
Banda 8 - Near Infrared (NIR) 1	0.78 - 0.90	10
Banda 8A - Near Infrared (NIR) 2	0.85 - 0.87	20
Banda 9 - Water vapour	0.93 - 0.95	60
Banda 10 - Cirrus	1.36 - 1.39	60
Banda 11 - SWIR 1	1.56 - 1.65	20
Banda 12 - SWIR 2	2.10 - 2.28	20

Tabla 1: Número de píxeles identificados y asignados a los conjuntos conjuntos de entrenamiento y test. Fuente: GIS&Beers

Estas dos resoluciones combinadas ya convierten a Sentinel-2 en una de las mejores opciones a la hora de elegir imágenes de satélite de acceso libre para trabajar con procesos vegetales, pero además cuenta con un tiempo de revisita de 5 días, lo que es muy importante para lograr escenas libres de cubierta nubosa en una zona que sólo suele contar con entre 30 y 50 días despejados al año (Linés Escardó, 1970). La zona elegida, además, se encuentra en una zona de solape entre los dos satélites de la misión (Sentinel-2A y Sentinel-2B), lo que arrojará una imagen cada 2 o 3 días y permitirá duplicar la posibilidad de lograr escenas útiles (ver Figura 5).

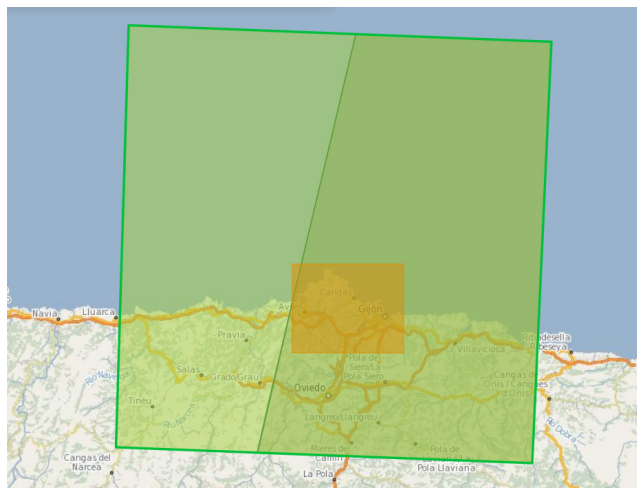


Figura 5: Zona de estudio (naranja), encuadrada dentro de la zona de solape de ambos satélites (verde más oscuro). Fuente: ESA. Autor: Elaboración propia.

2.2. Adquisición de Imágenes

Para la descarga de imágenes se ha recurrido al servicio Copernicus Acces Hub del programa Copernicus de la Agencia Espacial Europea (E.S.A.) (<https://scihub.copernicus.eu/dhus>), que permite descarga de imágenes de las distintas misiones de Sentinel, filtrando la búsqueda por localización, fechas, porcentaje de cubierta nubosa, sensor, nivel de corrección, etc.

La E.S.A. ofrece imágenes ya corregidas atmosféricamente, de forma que se eliminan los efectos de dispersión y absorción de la atmósfera sobre la luz cuando es atravesada por ella, para obtener la reflectancia que se registraría a nivel de superficie, en lugar de la recibida por el sensor en el satélite. Este nivel de corrección es llamado nivel 2A y es en el que se descargan todas las imágenes a usar en este trabajo.

Al descargar las primeras imágenes se estimaba que la forma más precisa de identificar parcelas afectadas por la presencia de *C. selloana* sería la identificación en campo, aunque pronto se hizo evidente que la cantidad de horas requeridas para lograr identificar un volumen suficiente de localizaciones para el modelo excedía la capacidad de recursos que se podían destinar al presente trabajo. A pesar de ello, con este primer enfoque, se realizó la descarga de las tres primeras imágenes con las que se realizaron los primeros test en los scripts para recortes y configuración de la red. Para asegurar que las imágenes contuvieran las plantas identificadas en campo en el segundo trimestre de 2022, se descargaron imágenes con fechas 01/11/2021, 24/01/2022 y 06/04/2022. Se supuso que, como la *C. selloana* florece en los meses de agosto y septiembre (ver Figura 6) y mantiene el característico penacho de sus inflorescencias durante todo el año, las plantas que poseían flores en su identificación en Junio, también contarían con flores en las fotos desde septiembre del año anterior.

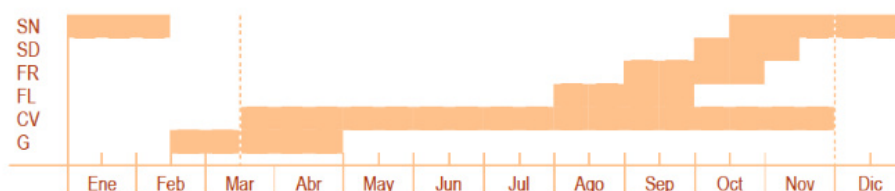


Figura 6: Fenología de *C. selloana* en Bizkaia. G: Germinación; CV: Crecimiento vegetativo; FL: producción de flores; FR: producción de frutos; SD: dispersión de semillas; SN: senescencia de hojas. Las líneas de puntos delimitan el periodo de actividad vegetal. Fuente: Herrera y Campos 2006. Autor: Ministerio para la Transición Ecológica.

Con este planteamiento se descargaron y se trabajó con las siguientes escenas (ver Figura 7):



Figura 7: Banda 2 (Azul) del Sentinel-2 en la imagen de 13/10/2021 (Escena completa). Fuente: E.S.A.

- S2B_MSIL2A_20211013T110919_N0301_R137_T29TQJ_20211013T125854
- S2B_MSIL2A_20220124T112249_N0301_R037_T29TQJ_20220124T130231
- S2A_MSIL2A_20220406T110621_N0400_R137_T29TQJ_20220406T172813

Para poder cambiar metodología de identificación de las plantas, sustituyendo el trabajo de campo por la localización a través de imágenes del PNOA, había que tener en cuenta la fecha en la que fueron tomadas las imágenes. Las imágenes del PNOA disponibles más recientes para Asturias a la fecha de elaboración de este trabajo, fueron capturadas en vuelos realizados entre el 3 al 29 de septiembre de 2020 (consultar en <https://pnoa.ign.es/productos-disponibles>). Como las plantas no producen flores antes del primer año, las plantas identificadas por este método estarán, también, en las imágenes tomadas desde al menos un año antes.

Se descargaron, finalmente, 5 imágenes tomadas en las fechas 23/08/2019, 07/09/2019, 26/12/2019, 14/02/2020 y 23/06/2020, que se descargan con los siguientes nombres codificados:

- S2A_MSIL2A_20190823T112121_N0213_R037_T29TQJ_20190823T123320
- S2B_MSIL2A_20190907T112119_N0213_R037_T29TQJ_20190907T144322
- S2B_MSIL2A_20191226T112359_N0213_R037_T29TQJ_20191226T122946
- S2B_MSIL2A_20200214T112109_N0214_R037_T29TQJ_20200214T122355
- S2B_MSIL2A_20200623T112119_N0214_R037_T29TQJ_20200623T143258

Al no ser imágenes recientes, no se encuentran para la descarga directa y hay que solicitar que se pongan en línea, lo que en este caso tardó unas 12 horas.

2.3. Preparación del entorno

El trabajo se desarrolló en dos equipos, en los que se procuró preparar el mismo entorno de trabajo e instalando en ambos las mismas versiones de los mismos programas, lenguaje de programación y librerías. Sobre Windows 10 Pro se trabajó con la distribución Anaconda de Python, que permite una fácil creación de entornos de programación, instalación de paquetes de dependencias y aplicaciones de programación como son los IDEs.

El primer equipo de trabajo contó con un procesador Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz 3.19 GHz, con 8 gigas de memoria RAM y una GPU AMD Radeon R7 430 y el segundo dispuso de una CPU AMD Ryzen 9 3900X 3.8 GHz, con 32 GB de memoria RAM y una tarjeta gráfica Geforce GTX 970. Sobre este último recayeron las tareas de entrenamiento y predicción de la RNA, dadas sus mayores capacidades.

En ambos se preparó un entorno en Anaconda con Python 3.8, el IDE SPyder 5.1.5 y se instalaron los paquetes de Pandas 1.4.1, GDAL 3.0.2, Tensorflow 2.3.0 y Keras 2.4.3. Se instaló también el software de GIS QGIS 3.22.4-Białowieża y dentro de éste se instaló el Plugin Serval 3.10.2.

3. Metodología y procedimientos

La presente sección se dedica a describir el conjunto de técnicas y los procedimientos empleados para transformar, haciendo uso de los materiales descritos anteriormente, los datos adquiridos, en los resultados que se pretenden obtener.

3.1. Pre-procesado de imágenes y creación de los conjuntos de datos

Las imágenes se descargan comprimidas en una carpeta con múltiples subcarpetas, con variedad de datos y metadatos sobre las mismas y se encuentran en formato *jp2*, en un archivo con dicha extensión para cada banda. Una vez descomprimidas las imágenes, se deben buscar los archivos por el entramado de carpetas, cargarlas en un software de GIS para realizar los recortes y transformaciones necesarias para poder crear los conjuntos de datos necesarios. Éste es un proceso lo suficientemente largo como para entorpecer la tarea si se decide cambiar las imágenes a utilizar o el número de las mismas, con el fin de obtener mejores resultados.

Para agilizar todo el proceso y permitir una mayor flexibilidad a la hora de variar las imágenes a utilizar, se escribió un script que permite descomprimir todas las imágenes que se desee en una carpeta de manera que éste las lee y busca las escenas en su interior. Una vez localizadas las imágenes, realiza los recortes y crea un conjunto de datos en formato *.xyz* que incluye en sus columnas las coordenadas centrales de cada pixel, las bandas elegidas y el Índice de Vegetación de Diferencia Normalizada (NDVI) definido en la Ecuación (1), calculado durante la ejecución del programa. Dicho índice emplea la diferencia entre las reflectividades en las longitudes de onda del infra rojo cercano (NIR) y el rojo, ofreciendo un resultado normalizado, que arroja resultados entre -1 y 1, donde para los valores por debajo de 0 se interpreta como objeto inanimado o planta muerta y por encima las cubiertas vegetales teniendo valores cercanos a 0 las plantas enfermas y creciendo hacia 1 al ganar vigorosidad. Se estima que la inclusión de este índice puede resultar útil al proceso de aprendizaje de la red neuronal.

$$NDVI = \frac{NIR - RED}{NIR + RED} = \frac{B8 - B4}{B8 + B4} \quad (1)$$

Durante todo el procedimiento se ha buscado trabajar siempre con el mismo sistema de coordenadas y proyección cartográfica, usando para ello el EPSG:32629 (elipsoide de referencia WGS 84, proyectado en UTM para el huso 29), manteniendo así los parámetros con los que son provistas las imágenes, lo que permite mantener la ubicación y orientación de los píxeles en todo momento.

QGIS cuenta con la posibilidad de instalar múltiples plugins o complementos, que permiten extender las capacidades originales del software GIS. Entre los mencionados plugins se encuentra Serval, que permite cambiar el valor de píxeles de capas ráster por superposición de capas vectoriales y que sirvió para formar los conjuntos de entrenamiento y test, creando una capa clasificada. Para ello se trazan en QGIS capas de polígonos que circunscriben zonas en las que, sobre todos los píxeles, ha sido identificada la presencia o la ausencia de *C. selloana*. Estas capas se superponen con otra creada para este propósito, que cuenta con una sola banda donde todos los píxeles se inician con valor 0 y que ocupa la misma extensión y tamaño de píxel que los recortes. La superposición del nuevo ráster con la capa vectorial clasificada como "sí" (tiene *C. selloana*) cambiará los píxeles afectados de valor 0 a 1, mientras los clasificados como "no", adquirirán valor 2. Tras terminar el proceso esta capa es guardada en formato *.xyz* (ver Figura 8).

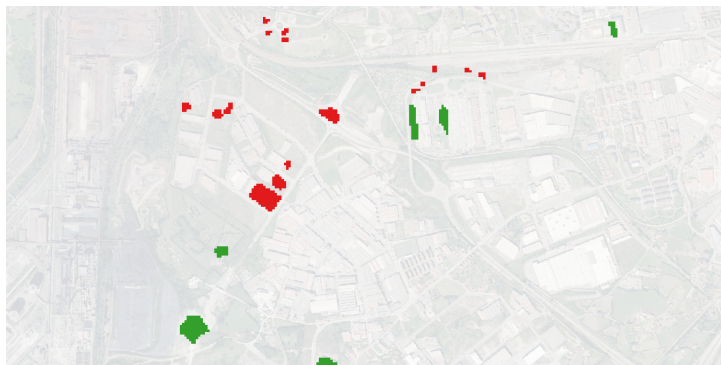


Figura 8: Detalle de píxeles identificados como libres de Cortaderia (verde) y otros con presencia por la planta (rojo). Autor: Elaboración propia

En la Tabla 2 se observa la cantidad de píxeles clasificados como 'sí' y aquellos que han sido catalogados como 'no', el total de píxeles del cada conjunto (entrenamiento o test), el total clasificado y el total de la escena sobre la que se realizará la predicción. También se muestra el porcentaje que estos píxeles representan sobre el total de clasificados y sobre toda de la escena.

	Sí	No	Entrenamiento	Test	Clasificados	Total Escena
Píxeles	2444	19384	17462	4366	21828	6555600
% Clasificados	11.20	88.08	80.00	20.00	—	—
% Total Escena	0.04	0.30	0.27	0.07	0.33	—

Tabla 2: Número de píxeles identificados y asignados a los conjuntos conjuntos de entrenamiento y test.

Para tratar de reducir la mala asignación de zonas clasificadas con una categoría a un píxel contiguo que no pertenecería a dicha clase, se ha tratado de no identificar las zonas hasta muy cerca de su perímetro exterior, dejando un cierto margen y así minimizar la posibilidad de que suceda dicha transferencia.

Como último paso para su posterior introducción en la red neuronal y clasificación final, se realiza otro script encargado de leer los archivos creados con anterioridad: tanto los de recortes como los de la clasificación. De la capa clasificada se seleccionan aquellos catalogados como 'sí' o 'no' y se añaden a sus filas los valores de las reflectividades de las bandas seleccionadas y el NDVI para cada imagen, lo que formará la matriz de características. Además, se crea una matriz que contendrá los valores de las reflectividades de las bandas y el NDVI de todos los píxeles de la zona de estudio, pero que no tendrá vector de clasificación, puesto que la obtención del mismo es el objetivo final de la red neuronal.

En el presente caso se ha obtenido un conjunto de datos para entrenamiento y test con 19384 píxeles que se consideran correctamente clasificados y de los que se han seleccionado todas las bandas de 10 y 20 metros de resolución, además del NDVI para cada uno, resultando un total de 10 columnas por foto y, dado que se seleccionaron 3 imágenes a lo largo del año, se obtiene un conjunto de datos de 19384 filas x 30 columnas.

3.2. Configuración de la Red Neuronal Artificial

Tras la obtención de los conjuntos de entrenamiento y test, se procede a la configuración de una red neuronal con capas densamente conectadas. En estas redes la totalidad las neuronas de cada capa están conectadas con todas las neuronas de la capa inmediatamente anterior o posterior (ver Figura 9). Siempre habrá una capa de entrada que recibe los datos de la matriz de características, una serie de capas ocultas y una capa de salida con tantas neuronas como posibilidades de clasificación tenga su vector de clasificación, que como se quiere determinar la posibilidad de presencia de *C. selloana* en cada píxel, será sólo una.

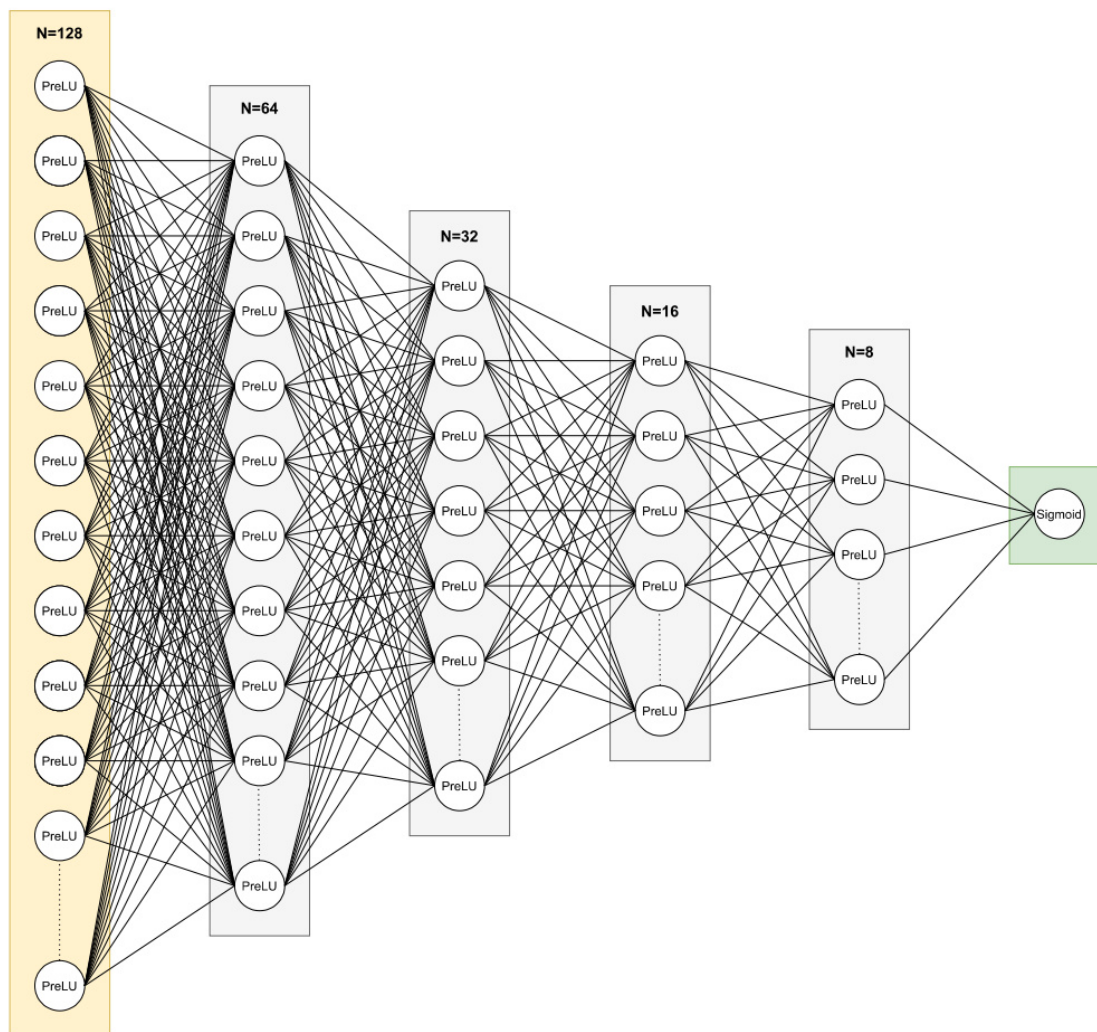
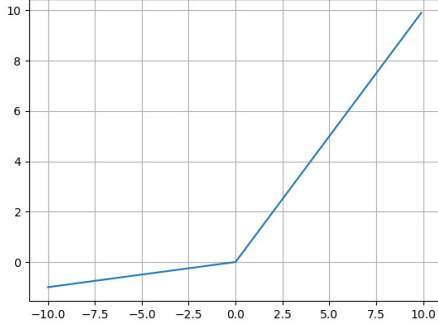


Figura 9: Diagrama de capas de la red neuronal densamente conectada. Autor: Elaboración propia

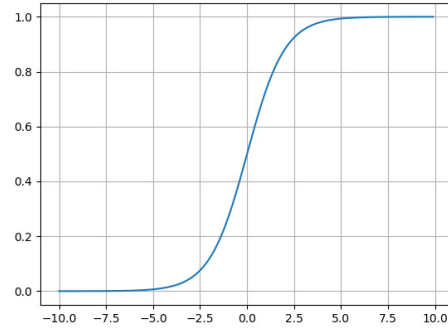
Para crear la red neuronal se ha usado Keras, una librería de alto nivel que provee una interfaz sencilla en Python para TensorFlow.

Se ha optado por una red con una capa de entrada con 128 neuronas, cuatro capas ocultas con 64, 32, 16 y 8 neuronas, todas estas capas activan sus neuronas con la función de activación Parametric Rectified Linear Unit (PReLU), definida en la ecuación 2 y la capa de salida emplea la función sigmoide, en la Ecuación 3 (ver Figuras 10 y 11).



$$PReLU(x_i) = \begin{cases} x_i & \text{si } x_i > 0 \\ a_i x_i & \text{si } x_i \leq 0 \end{cases} \quad (2)$$

Figura 10: Función PReLU. Autor: Programador de Click.



$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (3)$$

Figura 11: Función sigmoide. Autor: Programador de Click.

Con el objetivo de evitar problemas de sobreajuste (overfitting), con los que el modelo no sería capaz de generalizar y funcionar correctamente para todos los datos de entrada, se añade a todas las capas un Dropout del 10 %, lo que quiere decir que una de cada 10 neuronas por capa serán 'apagadas' en cada repetición y la red tendrá que realizar el ajuste sin ellas, lo que dificulta que el modelo alcance una excesiva particularización para el conjunto de datos con los que se ha entrenado (Antona 2020).

Para la compilación se ha usado el optimizador de Adam, que combina unos buenos resultados en la detección de la dirección del descenso gradiente estocástico (la dirección en la que desciende el error en las predicciones) con un bajo consumo de recursos de computación (Alom 2021). Al realizar una clasificación binaria, será empleada la función de coste de entropía cruzada binaria que se define a continuación.

$$Loss = -\frac{1}{outputsize} \sum_{i=1}^{outputsize} y_i \cdot \log \hat{y}_i + (1 - y_i) \cdot \log(1 - \hat{y}_i) \quad (4)$$

La red neuronal se inicia comenzando el proceso de propagación hacia adelante (forward propagation) (ver Figura 12), en el que cada capa tendrá un parámetro \mathbf{b} que será igual para todas las neuronas de la capa y recibirá un parámetro \mathbf{X} . Este valor es extraído de los datos de la matriz de características (el valor normalizado de la reflectividad en una banda) en la capa de entrada o la salida de otra neurona de una capa previa en las demás capas. La neurona transformará el valor recibido con su función de activación y transmitirá el resultado a la capa siguiente multiplicado por un peso \mathbf{W} , que en la primera repetición será asignado al azar (Rodríguez 2018). Para la capa de entrada se genera un peso por columna de la matriz de características para cada neurona y en todas las capas generará un peso la unión de cada neurona con todas las de la capa siguiente. Irá transitando por cada capa hasta llegar a la de salida, donde contrasta el resultado obtenido con el esperado en el vector de clasificación del conjunto de entrenamiento y se inicia la propagación hacia atrás (back propagation). Durante la propagación hacia atrás irá corrigiendo el error ajustando cada peso en sentido contrario al gradiente que ha sido determinado por el optimizador, ajustando todos los pesos hasta llegar a los de la capa de entrada, comenzando otra vez la propagación hacia adelante, ahora con los pesos que se acaban de corregir. Cuando el desempeño es correcto, con las sucesivas repeticiones de la operación, se logra la progresiva reducción del error, realizando lo que se llama convergencia del modelo, en el que la función de coste arroja un valor más pequeño, mientras el de la precisión será más alto con la sucesión de iteraciones, hasta que se alcanza un punto en el que no se mejora o se hace muy lentamente y con un coste computacional muy elevado. Este proceso llamado 'entrenamiento' permite obtener un modelo capaz de generar predicciones de cierta precisión, pudiendo

pasar a realizar dichas predicciones sobre el conjunto de datos que no han sido utilizados durante la fase de entrenamiento.

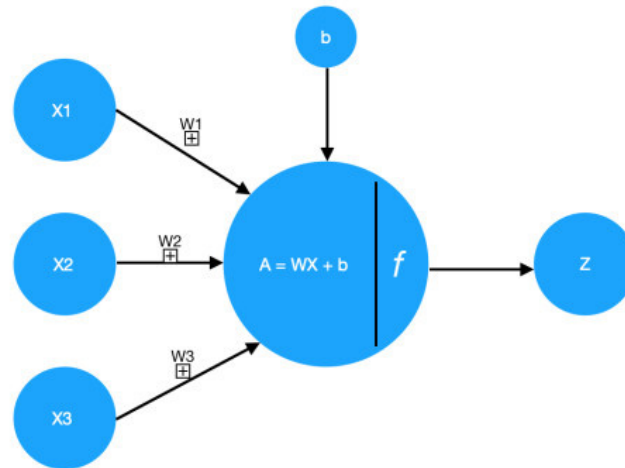


Figura 12: Flujo de una neurona. Autor: Rodríguez 2018

3.3. Optimización de los parámetros de la red

Se ha optado por una configuración fija en el número de capas y la cantidad de neuronas de las mismas, además de optar por una función de activación de reconocida solvencia, como es PReLU. Se ha decidido así tras considerar que la realización de las pruebas necesarias para lograr una buena optimización de estos parámetros que permiten gran cantidad combinaciones alargaría demasiado el trabajo, dado el tiempo que tendría que invertirse en el entrenamiento y supervisión de las distintas configuraciones.

Para la configuración de otros parámetros importantes se ha recurrido al método de Grid Search Cross Validation (GridSearchCV) para su estudio, éste permite realizar un análisis estadístico de los resultados obtenidos por la red siendo entrenada probando todas las combinaciones de los mismos.

Con el fin de mejorar el desempeño de la red se realizará el Grid Search con varias combinaciones de:

- Épocas (epochs): el número de ciclos completos de forward y back propagation que completa la red hasta finalizar el entrenamiento.
- Tamaño de lote (batch size): el número de datos que se introducen en cada iteración posibilitando que, en vez de calcular los errores y compensaciones de peso dato a dato, se agrupen para reducir la carga de operaciones a realizar por el equipo.
- Optimizadores: algoritmos encargados de detectar la dirección en la que se desplaza el gradiente, lo que permite corregir el error en la dirección adecuada.

Para el número de épocas se probará con 500, 1000 y 2000 repeticiones, en el caso del batch size se realizará el test con lotes de 4, 8 y 16 datos. Como salida es obtenida la precisión para la mejor combinación de parámetros (ver Tabla 3).

En este caso la mejor combinación fue la de 2000 épocas y batch de 16 datos, pero con escasa diferencia sobre la combinación de 1000 épocas y batch de 8 datos. Como esta última media tiene un mayor efecto del error de las primeras repeticiones y un tiempo de cálculo menor, se estimo la opción más interesante para el desarrollo de la red.

Combinación	Precisión media
<i>batch: 8, epochs: 1000</i>	0.984022
<i>batch: 16, epochs: 2000</i>	0.985110

Tabla 3: Precisión de las mejores combinaciones de parámetros.

3.4. Resultados preliminares y proceso iterativo de clasificación

Una vez creada y configurada la red neuronal, ésta se puede empezar a entrenar para obtener resultados preliminares en los que se aprecian zonas con problemas de identificación provocados porque los píxeles clasificados en primera instancia no recogían una suficiente variabilidad de los datos. Esto provoca un problema llamado *underfitting* (ajuste insuficiente), ya que en las primeras fases no se ha conseguido dar a la red una clasificación que le permita distinguir correctamente los fenómenos. Un ejemplo análogo podría ser el siguiente: si se quisiesen identificar píxeles que contengan hierba, sin ofrecer a la red zonas de césped artificial y otro tipo de cubiertas artificiales pintadas de verde, clasificadas como zonas sin hierba, sería fácil que una red neuronal las clasificase erróneamente, puesto que son verdes.

Para reducir este problema, se identificaron cubiertas que generaban confusión a la red y procedió a su correcta clasificación, tanto en zonas donde se predijo de forma errónea la presencia de *C. selloana* (falso positivo) como en otras que provocaban el efecto contrario (falsos negativos). Dado que la red fue sufriendo modificaciones al avanzar el trabajo, se observaron diversas zonas que provocaban errores y que también cambiaban al cambiar los parámetros de la red. Estas zonas se consideraron de interés para el entrenamiento y se procuró su identificación visual y su incorporación manual a la capa que sirve como fuente para los datos de entrada a la red. Tras un nuevo entrenamiento, la red aprende a identificar correctamente estas zonas, mejorando sensiblemente sus predicciones (ver Figura 13).

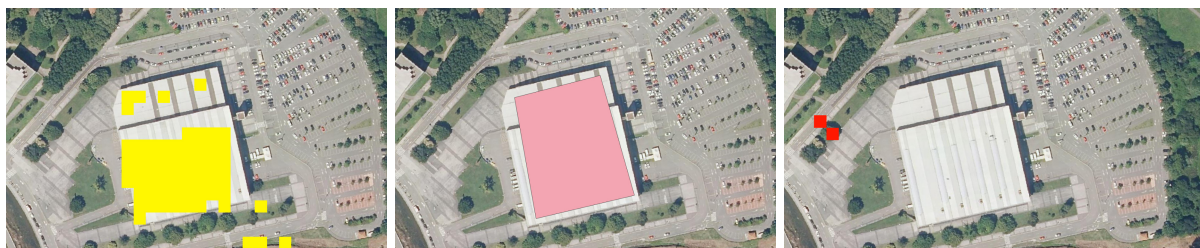


Figura 13: A la izquierda: clasificación inicial incorrecta; en el centro: clasificación manual correcta de la zona; a la derecha: identificación correcta realizada por la red tras un nuevo entrenamiento.

3.5. Validación de resultados

Con el fin de comprobar la calidad de los resultados obtenidos se realizará una validación cruzada *k*-fold (*k*-fold cross validation). Este tipo de validación permite evaluar la capacidad de generalización de los modelos y evitar el sobreajuste (Berrar 2018). Para ello divide aleatoriamente el conjunto de datos en *k* partes aproximadamente iguales de manera que una de ellas se usa para la validación del entrenamiento que se realiza sobre los demás. En cada iteración posterior el conjunto usado para la validación cambia, de tal forma que cuando termina el proceso, todos los datos han sido utilizados como entrenamiento y como test (ver Figura 14).

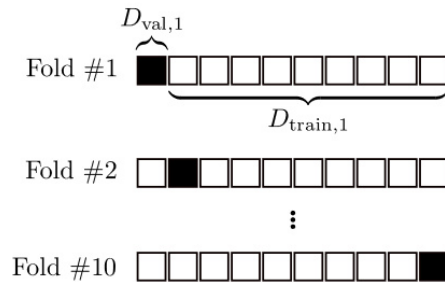


Figura 14: 10-fold cross validation, donde cada subconjunto contiene aproximadamente el 10 % de los datos. Autor: Berrar 2018

Esta técnica ya fue implementada en el paso anterior, ya que GridSearchCV permite introducir un parámetro de cross validation, lo que mejora la calidad de la validación. En ese caso se configura la red con la combinación elegida tras el GridSearchCV para obtener, con el proceso de k -fold cross validation, las medias de precisión para cada fold (ver Tabla 4) y algunas estadísticas que ayuden a detectar la existencia de un alto sesgo o grandes desviaciones entre folds (ver Tabla 5).

Fold n ^o	Precisión media	Fold n ^o	Precisión media
1	0.986835	6	0.976518
2	0.986262	7	0.982245
3	0.981672	8	0.983391
4	0.989691	9	0.983391
5	0.985109	10	0.985109

Tabla 4: Precisión media para cada fold.

media	0.984022
std	0.003547
min	0.976518
mediana	0.984250
max	0.989691

Tabla 5: Principales estadísticas extraídas del proceso de validación cruzada.

Se observa que la distancia entre media (0.984022) y mediana (0.984250) es menor que un 0.02 %, que la diferencia entre el valor más alejado de la media (0.976518) y esta no llega a un 0.8 % o que la desviación típica alcanza un valor bajo (0.3547 %). Estas medidas indican que los datos de entrada carecen de grandes problemas de sesgo o desviación y el modelo tendrá capacidad de crear predicciones consistentes para nuevos conjuntos de datos distintos a los empleados en su entrenamiento.

3.6. Obtención de las salidas gráficas y comprobación visual de resultados

Tras completar los pasos anteriores la red estará lista para realizar el entrenamiento definitivo y llevar a cabo la lectura de la matriz del recorte completo. Es en este momento cuando realiza una predicción para cada píxel, con lo que crea un vector de clasificación cuyos elementos toman valores entre 0 y 1, que es posteriormente añadido como última columna a la matriz de características. Finalmente se eliminan todas las columnas salvo las coordenadas y el vector que se acaba de obtener, se eliminan las cabeceras y se almacena en formato *.xyz*, con lo que adquiere un formato adecuado ser cargada directamente en QGIS.

Una vez cargada la salida como capa ráster, ésta se superpone a la imagen del PNOA. La única banda de esta capa tiene un valor que indica el grado de pertenencia a la clase a la que pertenecen los píxeles que contiene *C. selloana*, lo que servirá para aplicar una transparencia a los niveles que queden por debajo de cierta magnitud en dicha banda. El nivel hasta el que se aplica la transparencia se va modificando y se coteja cuidadosamente sobre el mapa buscando zonas de posibles fallos tanto por exceso y como por defecto de clasificación, hasta que se obtiene el mejor resultado posible y se asume ese valor como el que delimitará las dos clases. En este caso ha sido el valor de 0.99, ya que se ha visto que por debajo de él aumentaban los píxeles sueltos que podrían ser considerados ruido, sin lograr con ello un aumento en las clasificaciones aparentemente correctas que lo justifique.

Para evitar realizar una comprobación de cada píxel se ha planteado una clasificación con dos métodos distintos. Primero se aplica un filtro en el que se extraen las zonas con 5 o más elementos vecinos en los que se ha detectado la presencia de *C. selloana*, para después convertirlos en polígonos, lo que simplifica la comprobación de las entidades al reducir enormemente el número de las mismas. Tras esto se realiza la misma operación de poligonización sobre todos los píxeles clasificados, sin filtro, y se sustraen las capas, dejando sólo aquellos polígonos que contienen entre 4 y 1 elementos clasificados. Sobre esta nueva capa se hará una extracción aleatoria del 5 % de todas las entidades de la misma y es sobre dicha muestra sobre la que se procederá a la identificación visual.

Superponiendo estas capas a la ortofoto del PNOA que sirvió inicialmente para identificar las zonas se hará una comprobación entidad a entidad. Creándose en las capas una nueva columna que será empleada para asignar a cada polígono los siguientes valores según se estime su clasificación en la comprobación visual:

- 0: incorrectamente clasificadas.
- 1: correctamente clasificadas.
- 2: no apreciable.

De esta comprobación se extraerán finalmente estadísticas para estimar la bondad del ajuste realizado.

4. Resultados

Tras describir los medios, métodos y procedimientos empleados para lograr una predicción con la mayor precisión posible, se llevará a cabo el entrenamiento y se obtendrán los resultados definitivos. En esta sección serán presentados y analizados tanto estadística como gráficamente.

Los resultados expuestos son los aquellos que se han extraído de entrenar la red con las 3 imágenes de 01/11/2021, 24/01/2022 y 06/04/2022, puesto es la combinación que se consideró que ofrecía una mejor predicción.

4.1. Resultados numéricos

A continuación se exponen las estadísticas obtenidas por dos vías diferentes. Primero se expondrá la información que las librerías TensorFlow y scikit-learn, con las que se ha implementado la red neuronal, obtienen durante el proceso de ajuste del modelo. Después se tratarán aquellas obtenidas tras contrastar visualmente las salidas gráficas con la realidad observada.

4.1.1. Estadísticas propias del modelo

Al realizar el ajuste sobre el conjunto de entrenamiento se generan las estadísticas de precisión (accuracy) y coste (loss) (ver Figura 15). Durante el proceso de entrenamiento de la red, ésta ofrece estos datos en el momento en el que son obtenidos y los almacena para su posterior consulta. Con estos datos se pueden generar gráficos que permiten comprobar de forma visual cómo va evolucionando el ajuste de dichos parámetros a lo largo de los sucesivos ciclos de front y back propagation.

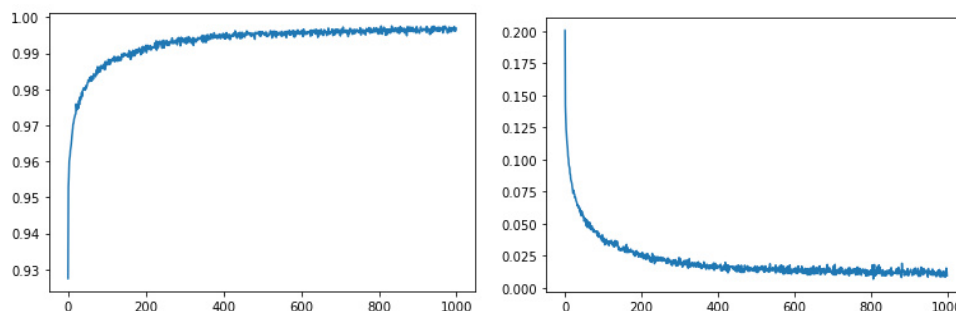


Figura 15: Gráfico que representa el valor de la estadística de precisión (izquierda) y de coste (derecha) en el eje Y en cada repetición representada en el eje X. Autor: Elaboración propia.

Otro elemento importante para observar cómo se ajusta el modelo a la entrada de datos clasificados que se le ha ofrecido, es el de la matriz de confusión (ver Tabla 6). Ésta se genera al realizar la predicción sobre el conjunto de test para el que se reservó el 20% de los datos que forman parte de la matriz de características clasificada y no han participado en el ajuste, contrastando las predicciones realizadas para cada píxel con los valores que se habían asignado previamente a los mismos. En esta matriz de confusión se sitúan en el eje vertical los valores reales, mientras que en el eje horizontal los valores predichos como salida de la red. Esto hace que arriba a la izquierda, en la posición (0,0), se encuentren los 3899 píxeles que sin contener *C. selloana*, han obtenido el mismo valor como salida de la red (verdaderos negativos). En la posición (0,1) se hallan 78 píxeles que habiendo sido identificada presencia de la planta en su interior, han sido clasificados como libres de la misma por la red (falsos negativos). En la posición (1,0) se sitúan los 4 píxeles que siendo realmente negativos, la red ha determinado que contienen *C. selloana* (falsos positivos). Por último, en la posición (1,1) se muestran los 385 píxeles del conjunto de test en los que la predicción de la red ha coincidido con la clasificación realizada de forma manual. En resumen, la matriz de confusión muestra un total de 4366 píxeles, 4284 de los cuales han sido correctamente clasificados por la red, mientras ha fallado únicamente en 82 casos.

	0	1
0	3899	4
1	78	385

Tabla 6: Matriz de confusión que muestra en columnas el valor

4.1.2. Estadísticas elaboradas tras la comprobación visual de resultados

Siguiendo el procedimiento descrito en la Sección 3.7 se consiguen los datos que permiten extraer estadísticas para evaluar la calidad de las predicciones.

Para los píxeles que no cumplen la condición de estar agrupados en conjuntos de 5 o más (ver Tabla 7), se extrae con QGIS una muestra del 5 % de ellos y se procede a la comprobación visual de todos los elementos de la muestra. Al realizar esta comprobación individual, el 51.8 % de los comprobados no han podido ser identificados con claridad como miembros de una clase concreta. Entre los elementos restantes, una gran mayoría de estos (37.05 %) está mal identificada, mientras solamente un 11.15 % ha sido identificados correctamente, por lo que se considera que la clasificación bajo ese umbral de $500m^2$ debe ser descartada.

	mal clasificados	bien clasificados	sin definir
Nº Píxeles	113	34	158
%	37.05	11.15	51.80

Tabla 7: Conteo y porcentaje de píxeles según el resultado de la comprobación visual para la muestra aleatoria extraída de los grupos de positivos con área inferior a $500m^2$.

La comprobación sobre aquellos grupos que sí pasaron el filtro que exigía que se agrupasen en conjuntos de 5 o más píxeles contiguos (ver Tabla 8), se realizó visualizando cada entidad para su evaluación. La mejora de este grupo, con respecto al de conjuntos menores de 5, es muy significativa, llegando a identificar correctamente la presencia de *C. selloana* para más de 10 píxeles por cada uno que falla. Si, además, se divide este grupo en menores y mayores 10 píxeles, se aprecia que entre los de menor área, los píxeles correctamente clasificados (207), apenas supera el de aquellos que se han considerado incorrectamente clasificados (185), mientras que de un 60 % del total de este grupo no ha podido ser comprobada su pertenencia a una clase. Se ha decidido mantener este grupo para su cartografiado, pues podría ser interesante conocer su ubicación para una identificación en campo, ya que un 50 % de posibilidades de contener una planta invasora es un riesgo a tener en cuenta y que podría justificar salidas a campo para comprobar todos esos lugares.

Tamaño de zonas	mal clasificados	bien clasificados	sin definir
Nº Píxeles >5	546	5529	1787
% >5	6.94	70.33	22.73
Nº Píxeles 5-10	185	207	590
% 5-10	18.84	21.08	60.08
Nº Píxeles >10	371	5352	1237
% >10	5.33	76.90	17.77

Tabla 8: Conteo y porcentaje de píxeles según el resultado de la comprobación visual para los grupos de positivos con área superior a $500m^2$.

En aquellos grupos que contienen 10 píxeles o más es donde se produce una identificación que da un nivel alto de confianza. Habiendo podido comprobar la pertenencia a una clase de un 82.23 % del total de píxeles en los que se ha predicho la presencia de *C. selloana*, se han identificados como correctamente clasificados 5352 píxeles, mientras solamente en 371 casos se ha considerado errónea su clasificación. Esto

significa que han clasificado bien el 93.51 % de los píxeles que se han podido comprobar, logrando casi 15 clasificaciones correctas por cada una incorrecta.

Se han identificado un total de 275 zonas que contienen 5 píxeles ($500m^2$) o más (ver Tabla 9) y se puede apreciar cómo el tamaño medio de las zonas bien clasificadas ($5882m^2$) es muy superior a la superficie de aquellas que se han clasificado de forma incorrecta ($1213m^2$) y también a la de aquellas que no han podido ser verificadas ($1314m^2$). Como muestran los datos, las zonas identificadas llegan a tener 8510 píxeles, mientras el tamaño máximo mal identificado es de 780 y la mayor superficie que no se ha podido comprobar no supera los 990. Esto se debe al método de comprobación visual desde ortofotos, donde se necesita poder apreciar las estructuras y texturas típicas que forman las colonias de *C. selloana*, siendo más sencillo de identificar la presencia de plantas en un lugar cuanto más grande es la zona a evaluar.

	count	mean	std	min	25 %	50 %	75 %	max
Mal clasificadas	45.0	1213.33	1260.52	500.0	600.0	800.0	1200.0	7800.0
Bien clasificadas	94.0	5881.91	11996.26	500.0	800.0	2250.0	6125.0	85100.0
No identificadas	136.0	1313.97	1637.94	500.0	600.0	700.0	1325.0	9900.0

Tabla 9: Sumario estadístico de las entidades y su área según el resultado de la comprobación visual para polígonos con área superior a $500 m^2$.

En el rango de entre $500 m^2$ y $1000 m^2$ (ver Tabla 10), las 28 zonas mal clasificadas tienen una media de $660m^2$, muy similar a los $690m^2$ que alcanzan las que han sido clasificadas incorrectamente y los $634m^2$ de los lugares cuya comprobación no ha resultado definitiva. Poniendo el foco sobre los percentiles, se puede apreciar que en el del 50 %, el valor que deja por encima la mitad de los resultados, y el del 75 %, el tamaño de la zona es mayor para los bien clasificados. Esto indica que se encuentran más grupos bien clasificados según se van estudiando superficies más extensas.

	count	mean	std	min	25 %	50 %	75 %	max
Mal clasificadas	28.0	660.71	147.421	500.0	500.0	600.0	725.0	1000.0
Bien clasificadas	30.0	690.00	168.87	500.0	500.0	700.0	800.0	1000.0
No identificadas	93.0	634.41	141.03	500.0	500.0	600.0	700.0	1000.0

Tabla 10: Sumario estadístico de las entidades y su área según el resultado de la comprobación visual para polígonos con área comprendida entre $500 m^2$ y $1000 m^2$.

Se puede observar nuevamente cómo es al considerar conjuntos con más de 1000 píxeles (ver Tabla 11) cuando se produce un gran aumento la calidad de la predicción. Tan solo se encontraron 18 zonas que se evaluaron como incorrectas, con un tamaño medio de $2061m^2$, mientras otras 68 zonas que ocupan una superficie promedio de $7988m^2$ han sido correctamente catalogadas. En esta categoría se acentúan las diferencias en cuanto a las medidas de las áreas, siendo la media de las superficies bien clasificadas ($7988m^2$) casi cuatro veces superior al tamaño medio de las zonas incorrectamente identificadas ($2061m^2$) y más de tres veces el de las no identificadas ($2361m^2$). Los percentiles también indican un aumento de las zonas bien clasificadas según se va aumentando el tamaño de las mismas. Se observa que para las agrupaciones de píxeles bien clasificadas una parcela de $390m^2$ tendría todavía un 50 % de todas las zonas bien clasificadas, pero la superficie que cumple esa propiedad en las mal clasificadas se queda en $1400m^2$. Mientras para la mediana la superficie es 2.8 veces más grande para los recintos correctos, que para los incorrectos, esta diferencia aumenta hasta 3.5 veces en el percentil 75. Estas diferencias también se observan en la superficie más grande de cada categoría, siendo de $7800m^2$ la mayor superficie mal clasificada y de $9900m^2$ la más grande cuya evaluación no ha resultado definitiva, pero llega hasta $851000m^2$ en el caso de las zonas que han sido catalogadas correctamente.

	count	mean	std	min	25 %	50 %	75 %	max
Mal clasificadas	18.0	2061.11	1678.28	1000.0	1200.0	1400.0	2050.0	7800.0
Bien clasificadas	67.0	7988.061	13678.37	1000.0	2000.0	3900.0	7150.0	85100.0
No identificadas	47.0	2631.91	2265.82	1000.0	1250.0	1900.0	2800.0	9900.0

Tabla 11: Sumario estadístico de las entidades y su área según el resultado de la comprobación visual para polígonos con área superior a $1000 m^2$.

4.2. Detalles de salidas gráficas

A continuación se mostrarán detalles de la representación gráfica de la predicción de la red neuronal a modo de ilustración sobre la calidad de los resultados obtenidos en puntos que podrían ser especialmente conflictivos. Con ello se logrará tener una idea más concreta sobre la calidad del desempeño del procedimiento realizado.

En las primeras pruebas de la red neuronal se apreció un problema que también se había visto en otros intentos de localización de *C. selloana* usando técnicas de teledetección: la sobreestimación de la probabilidad de presencia de la planta en las zonas de acantilados. Este problema se pudo corregir seleccionando píxeles no sospechosos de contener *Cortaderia* en estas zonas y clasificándolos correctamente para los sucesivos entrenamientos. Tras ello, la red muestra sólo algunos píxeles diseminados por las zonas de acantilados y por el litoral. De todos ellos sólo un conjunto pasa el filtro de los 5 píxeles agrupados, no pudiendo descartar, además, que se trate de una clasificación correcta (ver Figura 16).

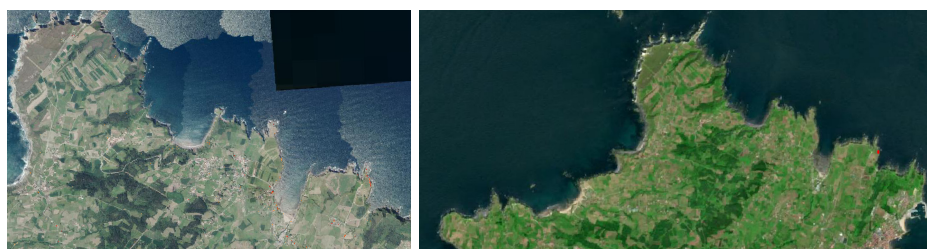


Figura 16: A la izquierda la zona con más píxeles identificados como *C. selloana* en acantilados (probablemente incorrectos); a la derecha gran extensión de costa donde aparece la única zona identificada sobre un acantilado (probablemente correcta). Autor: Elaboración propia.

Otro punto que podría añadir dificultad a la clasificación es el de que la *C. selloana* crece en distintos contextos. Puede invadir zonas de suelo casi desnudo, pero se puede encontrar frecuentemente mezclado con otras plantas como es el *Ulex europaeus* (ver Figura 17), muy presente en la zona de estudio. Además, la planta tarda entre 1 y 3 años en alcanzar el estado de madurez suficiente en el que adquiere su forma adulta con sus características inflorescencias. En las salidas gráficas se ven zonas clasificadas correctamente en varias de estas situaciones.

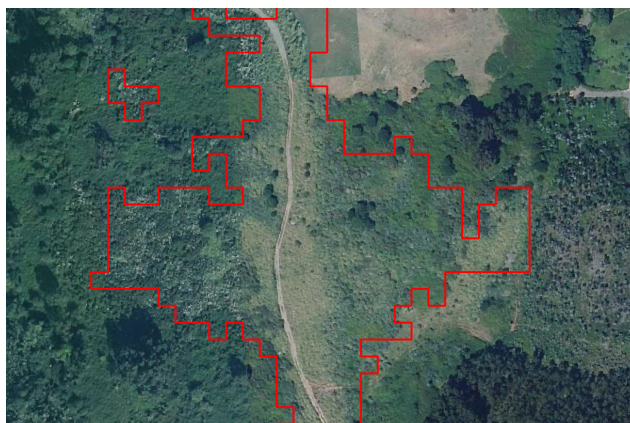


Figura 17: Detalle de la clasificación donde se ve correctamente clasificada una zona donde se encuentra *C. selloana* en sitintos contextos, mezclada con *Ulex europaeus*, otras con flores y zonas de plantas jóvenes que no han florecido todavía.

El proceso de filtrado para los píxeles que pasan el filtro de 5 píxeles tiene un efecto visual muy significativo en el mapa, eliminando gran cantidad de elementos que, como se ha comprobado anteriormente, son en su mayor parte falsos positivos. Como se puede ver en la Figura 18, se eliminan dichos píxeles mal clasificados, pero se conservan zonas que han sido correctamente identificadas.



Figura 18: A la izquierda se puede ver una zona salpicada con píxeles sueltos (probablemente incorrectos). En el centro, tras aplicar el filtro, sólo queda una gran zona (correcta), ampliada en la imagen de la derecha. Autor: Elaboración propia.

La necesidad de aplicar un umbral alto al nivel obtenido en la clasificación (0.99) y un filtro para eliminar la gran cantidad de falsos positivos que salpican toda la zona de estudio, tiene también como efecto indeseado la pérdida de clasificaciones correctas. Esta pérdida de información hará que algunas zonas que contienen *C. selloana* y habían sido identificadas con relativa corrección, se vean excluidas de la clasificación definitiva (ver Figuras 19 y 20).



Figura 19: Detalle de zona industrial con un grupo de plantas identificado en campo, que obtiene un resultado que para un umbral de 0.9 o sin aplicar el filtro, sería correctamente idetificada, pero con esta doble restricción, es incorrectamente descartada. Autor: Elaboración propia.



Figura 20: Detalle de zona industrial intensamente afectada. A la izquierda: clasificación para niveles entre 0.09-0.95 (amarillo), 0.95-0.99 (naranja) y >0.99 (rojo). A la derecha: zonas seleccionadas tras aplicar sólo nivel >0.99 y el filtro para zonas de 500 m^2 o más. Autor: Elaboración propia.

5. Conclusiones

En el desarrollo del presente trabajo se ha abordado la elaboración de un método que, partiendo de imágenes de satélite de acceso público como son aquellas que provienen de los satélites Sentinel-2, permitiese identificar la presencia de objetos contenidos en el espacio de cada píxel. Se ha tratado el doble desafío de identificar plantas que crecen en variedad de contextos configurados distintas superficies, entre diferentes plantas, solas o en grandes colonia, y de hacerlo con unas imágenes con una resolución máxima de 10x10 metros. Esto también brindó la posibilidad de poner a prueba las capacidades de las redes neuronales para reconocer patrones.

Además de las dificultades que había que enfrentar derivados de la naturaleza del problema y de las fuentes de datos disponibles, también había que hacerlo con una escasez de medios que prácticamente imposibilitaron las salidas de campo. No disponer de identificación sobre el terreno y teniendo que recurrir a la localización de especímenes desde imágenes de ortofotografía aérea supone ya un factor limitante para las predicciones, pues obliga a asumir que un número indeterminado de elementos habrán sido asignados a un grupo al que no pertenecían. Además, se consideró que esta forma de identificar píxeles sólo permitía una clasificación binaria para una realidad continua en la que un píxel, con la mitad de su área ocupada por la planta a identificar, tendrá valor 1, cuando podría ser interesante que tuviese valor 0.5. Como las imágenes usadas para crear esta primera clasificación son de vuelos con fecha de septiembre 2020 se daban dos escenarios. Por un lado, en uno se podrían emplear imágenes de satélite de fechas anteriores a las ortofotos, lo que aleja la predicción del estado actual de la realidad. En cambio, realizarlo con imágenes más recientes y posteriores a los vuelos del PNOA más actual podría suponer que se diesen cambios en el estado del terreno de algunas zonas que afectasen a la clasificación. Asumiendo que ambos enfoques tenían sus fortalezas y debilidades, se optó por suministrar ambos a la red y quedarse con el que mejor resultado ofreciese, que se consideró que fue la combinación de las 3 imágenes de octubre 2021, enero 2022 y abril 2022.

Las imágenes de Sentinel-2, sin tener una enorme resolución espacial, cuentan con una variedad de bandas, algunas de ellas, como las del Red Edge, de especial interés para el estudio de elementos vegetales. Además, los 5 días de resolución temporal que brinda Sentinel-2, que se ven reducidos a 2-3 días en la zona de estudio suponen una enorme ventaja en una zona tan nubosa como la cornisa cantábrica donde se encuentra la misma. Con esta realidad se abordó un trabajo que logró que para grupos de menos de 5 píxeles se necesiten algo menos de 5 intentos fallidos para encontrar una píxel con plantas, pero que por encima de ese límite empieza a mejorar rápidamente los resultados, alcanzando cerca del 50 % de acierto para áreas de entre $500m^2$ y $1000m^2$, y superando las 14 clasificaciones correctas por cada una fallida para zonas mayores de $1000m^2$.

En trabajos previos consultados en los que se habían hecho intentos de localización satelital de *C. selloana* como el de INVASAT (ver Figura 21), se encontraron dificultades en zonas de acantilado y ribera, o zonas de otras plantas como el *Phragmites australis*. Estas dificultades no se han detectado o se han podido superar con el entrenamiento adecuado.

El procedimiento descrito ha logrado resultados que permiten ampliar la información disponible, superando algunos problemas de otras metodologías, aunque con sus propias limitaciones. Parece probable que una ampliación de los recursos podrían permitir mejoras metodológicas que permitirían reducir el tamaño de la superficie que se podría identificar correctamente. Se proponen las siguientes opciones que podrían ser de interés para futuras investigaciones:

- Establecimiento de áreas y puntos controlados identificados sobre el terreno.

Sustituyendo el método de identificación desde imágenes de vuelos fotogramétricos por la identificación precisa sobre el terreno de parcelas invadidas, se podría lograr una mayor fidelidad de la identificación inicial que facilitaría la tarea de aprendizaje de la red.

- Visitas y comprobaciones en zonas conflictivas para agregarlas a futuros entrenamientos.

En la misma línea del punto anterior, poder visitar zonas donde la corrección de la identificación por la red resulta dudosa, podría ser de gran utilidad para que esta perfeccione su capacidad para distinguir superficies que en principio pueden resultar confusas.

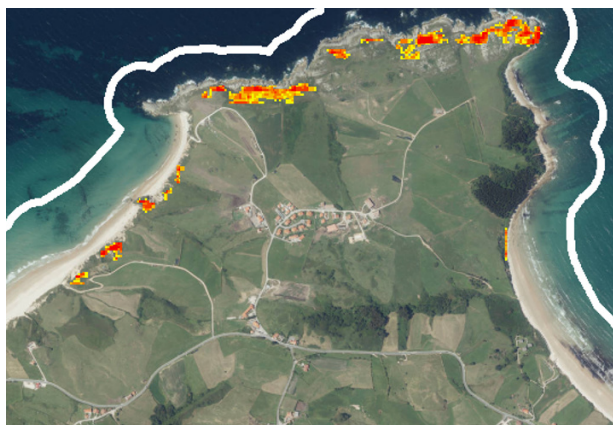


Figura 21: Problema con la identificación de los acantilados. Autor: INVASAT

- Estudiar la posibilidad de agregar medidas de textura a la matriz de características.
Realizar un análisis textural sobre imágenes de mayor resolución espacial, incorporando sus medidas como columnas a los píxeles, podría aportar información de utilidad para su correcta clasificación.
- Realizar pruebas con otros índices que pudiesen aportar información de interés.
Además de NDVI existen otros muchos índices que podría ayudar a realzar características de los terrenos invadidos por *C. selloana* mejorando el proceso de aprendizaje de la red neuronal.
- Plantear pruebas con más configuraciones y tamaños de la red.
Debido a la cantidad de parámetros disponibles, así como la infinidad de combinaciones de número de capas y cantidad de neuronas en las mismas, es probable que existan combinaciones que ofrezcan mejores resultados que la aquí empleada. Por ello podría ser recomendable dedicar tiempo a encontrar las configuraciones más adecuadas de la red.

Los resultados obtenidos para superficies mayores de $1000 m^2$, con más de un 82 % de clasificaciones correctas entre las comprobadas, están en consonancia con otras investigaciones similares como SatRed (Trujillo-Jiménez y col. 2022, que logró precisiones de entre el 76 % y el 92 % para usos del suelo y cubiertas vegetales. Se ha comprobado el enorme potencial de las redes neuronales para la identificación de este tipo de fenómenos, consiguiendo una clasificación bastante correcta y que es superior a la que se ha podido consultar previamente a la elaboración del presente trabajo, a pesar de tener que recurrir a métodos de filtrado similares. Todo esto se ha logrado con información disponible públicamente y usando equipos informáticos domésticos, sin apenas trabajo de campo y en un tiempo relativamente corto.

Tal vez, en su estado actual, la red no emita unas salidas que puedan ser usadas de forma directa como fuente cartográfica y estas no se deban considerar como ciertas por su elevada incertidumbre pero, a pesar de las limitaciones para la identificación de plantas sueltas, se ha mostrado como una herramienta muy potente para revelar zonas a partir de los $1000 m^2$. El método, además, admite mejoras, que con ciertos recursos, es previsible que permita un aumento de la calidad de los resultados reduciendo el límite a partir del que se considera efectiva.

El desarrollo del presente trabajo ha supuesto un desafío, pero, sobre todo, la oportunidad de poner en práctica conocimientos adquiridos durante el *Máster en Geotecnología y Desarrollo de Proyectos SIG*, en combinación con otras capacidades adquiridas con anterioridad a cursar el mismo. Se trabajó combinando de las herramientas de SIG y la programación informática, junto con los datos de imágenes de satélite, creando un método propio para su uso análisis con técnicas de Deep Learning que ha arrojado resultados que se consideran satisfactorios con unos recursos limitados en medios y tiempo. Además, al nacer de la preocupación personal que despierta en el autor la proliferación de especies exóticas invasoras, resulta emocionante la posibilidad de realizar un trabajo que pueda contribuir, aunque sea mínimamente, a su control, aportando una herramienta útil para la localización de forma masiva y de bajo coste.

6. Referencias

- [1] Ministerio de Agricultura, Alimentación y Medio Ambiente. *Real Decreto 630/2013, de 2 de agosto, por el que se regula el Catálogo español de especies exóticas invasoras*. <https://www.boe.es/eli/es/rd/2013/08/02/630/con>. 2013.
- [2] S. Saura-Mas y F. Lloret. “Wind effects on dispersal patterns of the invasive alien *Cortaderia selloana* in Mediterranean wetlands”. En: *Acta Oecologica* 27.2 (2005), págs. 129-133. ISSN: 1146-609X. DOI: <https://doi.org/10.1016/j.actao.2004.12.001>. URL: <https://www.sciencedirect.com/science/article/pii/S1146609X04001195>.
- [3] J. S. Rentch, R. H. Fortney, S. L. Stephenson, H. S. Adams, W. N. Grafton y J. T. Anderson. “Vegetation–site relationships of roadside plant communities in West Virginia, USA”. En: *Journal of Applied Ecology* 42.1 (2005), págs. 129-138. DOI: <https://doi.org/10.1111/j.1365-2664.2004.00993.x>. eprint: <https://besjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1365-2664.2004.00993.x>. URL: <https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2664.2004.00993.x>.
- [4] Ministerio para la Transición Ecológica. *Estrategia de gestión, control y posible erradicación del plumero de la Pampa (Cortaderia selloana) y otras especies de Cortaderia*. https://www.miteco.gob.es/es/biodiversidad/publicaciones/estategia_cortaderia_tcm30-478427.pdf. 2018.
- [5] Zakaria Adiri, Rachid Lhissou, Abderrazak El Harti, Amine Jellouli y Mohcine Chakouri. “Recent advances in the use of public domain satellite imagery for mineral exploration: A review of Landsat-8 and Sentinel-2 applications”. En: *Ore Geology Reviews* 117 (2020), pág. 103332. ISSN: 0169-1368. DOI: <https://doi.org/10.1016/j.oregeorev.2020.103332>. URL: <https://www.sciencedirect.com/science/article/pii/S0169136819305517>.
- [6] Irom Singha, Yelena Kakoty, B. Unni, Jaysree Das y Mohan Kalita. “Identification and characterization of Fusarium sp. using ITS and RAPD causing fusarium wilt of tomato isolated from Assam, North East India”. En: *Journal of Genetic Engineering and Biotechnology* 14 (jul. de 2016). DOI: 10.1016/j.jgeb.2016.07.001.
- [7] Chuan Guo, Geoff Pleiss, Yu Sun y Kilian Weinberger. “On Calibration of Modern Neural Networks”. En: (jun. de 2017).
- [8] Usman Rauf, Waqar S. Qureshi, Hamid Jabbar, Ayesha Zeb, Alina Mirza, Eisa Alanazi, Umar S. Khan y Nasir Rashid. “A new method for pixel classification for rice variety identification using spectral and time series data from Sentinel-2 satellite imagery”. En: *Computers and Electronics in Agriculture* 193 (2022), pág. 106731. ISSN: 0168-1699. DOI: <https://doi.org/10.1016/j.compag.2022.106731>. URL: <https://www.sciencedirect.com/science/article/pii/S0168169922000485>.
- [9] Lei Ma, Yu Liu, Xueliang Zhang, Yuanxin Ye, Gaofei Yin y Brian Alan Johnson. “Deep learning in remote sensing applications: A meta-analysis and review”. En: *ISPRS Journal of Photogrammetry and Remote Sensing* 152 (2019), págs. 166-177. ISSN: 0924-2716. DOI: <https://doi.org/10.1016/j.isprsjprs.2019.04.015>. URL: <https://www.sciencedirect.com/science/article/pii/S0924271619301108>.
- [10] Magda Alexandra Trujillo-Jiménez, Ana Laura Liberoff, Natalia Pessacg, Cristian Pacheco, Lucas Díaz y Silvia Flaherty. “SatRed: New classification land use/land cover model based on multi-spectral satellite images and neural networks applied to a semiarid valley of Patagonia”. En: *Remote Sensing Applications: Society and Environment* 26 (2022), pág. 100703. ISSN: 2352-9385. DOI: <https://doi.org/10.1016/j.rsase.2022.100703>. URL: <https://www.sciencedirect.com/science/article/pii/S2352938522000118>.
- [11] J.M. Uribarri. “RNA-Estradas - Procesado de datos e creación dunha Rede Neuronal Artificial para a detección automática de asfalto con imaxes do Sentinel 2.” <https://github.com/Uribarri/RNA-Estradas>. 2020.

- [12] Han-Chen Huang, Allen Chang y Chih-Chung Ho. “Using Artificial Neural Networks to Establish a Customer-cancellation Prediction Model”. En: *Przeglad Elektrotechniczny* 89 (ene. de 2013), págs. 178-180.
- [13] M Herrera y J.A.l Campos. *El carrizo de la pampa (Cortaderia selloana) en Bizkaia. Guía práctica para su control*. Universidad del País Vasco, Diputación de Bizkaia. 2006.
- [14] A. Antona. “Aplicación del Dropout a la Cuantificación de la Incertidumbre de Redes Neuronales.” https://oa.upm.es/57875/1/TFM_ALEJANDRO_ANTONA_CASTANARES.pdf. 2020.
- [15] Mohammed Alom. “Adam Optimization Algorithm”. En: (jun. de 2021).
- [16] V. Rodríguez. “Conceptos básicos sobre redes neuronales.” <https://vincentblog.xyz/posts/conceptos-basicos-sobre-redes-neuronales>. 2018.
- [17] Daniel Berrar. “Cross-Validation”. En: ene. de 2018. ISBN: 9780128096338. DOI: 10.1016/B978-0-12-809633-8.20349-X.

A. ANEXO 1: SCRIPTS

B. ANEXO 2: MAPAS

A. Anexo I

A continuación se adjuntan los scripts elaborados en lenguaje Python para la automatización de los geoprocursos, la extracción y manipulación de los datos de las imágenes, la creación de la red neuronal y la obtención de estadísticas de los resultados arrojados por esta.

Listing 1: Script para crear recortes y calcular NDVI

```
import os, gdal, pandas

# introducir la carpeta donde se guardan las imágenes
dir_imagenes = 'G:/TFM/datos/raster/Sentinel-2/'
contenido = os.listdir(dir_imagenes)

# Crear la lista de las ubicaciones de las bandas a recortar

listaEscenas = []
imagenes = []
incluidas = ["B02_10m", "B03_10m", "B04_10m", "B05_20m", "B06_20m", "B07_20m", "B08_10m",
            "B11_20m", "B12_20m"] # Bandas a recortar

for directorio in contenido:
    if "L2A_" in directorio:
        ruta = dir_imagenes + directorio + "/GRANULE/"
        dir1 = os.listdir(ruta)

        dir10m = ruta + dir1[0] + "/IMG_DATA/R10m/"
        dir20m = ruta + dir1[0] + "/IMG_DATA/R20m/"
        dir60m = ruta + dir1[0] + "/IMG_DATA/R60m/"
        dirs = [dir10m, dir20m, dir60m]
        for i in dirs:
            listaImágenes = os.listdir(i)
            for imagen in listaImágenes:
                for banda in incluidas:
                    if banda in imagen:
                        listaEscenas.append(i+imagen)

for banda in listaEscenas:
    carpeta = dir_imagenes + 'recortes/' + banda[-27:-12] + '/'
    try:
        os.stat(carpeta)
    except:
        os.mkdir(carpeta)
    if '10m' in banda:
        recortado= gdal.Warp(carpeta + banda[-34:-4]+ '_recortado.tif',
            banda,
            outputBounds=(748360,4817680,775360,4841960),
            dstNodata=(0))
        gdal.Translate(carpeta + banda[-34:-4]+ '_recortado.xyz',
            recortado,
            format='xyz')
    else:
        recortado= gdal.Warp(carpeta + banda[-34:-4]+ '_recortado.tif',
            banda,
            outputBounds=(748360,4817680,775360,4841960),
            xRes= 10,
            yRes= 10,
            dstNodata=(0))
        gdal.Translate(carpeta + banda[-34:-4]+ '_recortado.xyz',
            recortado,
            format='xyz')

carpetas = os.listdir(dir_imagenes + 'recortes/')

ruta=dir_imagenes + 'recortes/' + carpetas[0] + '/'
coords = pandas.read_csv(ruta + os.listdir(ruta)[1], sep = ' ', header = None)
```

```

coords = coords.iloc[:, [0,1]]
ndvi_xyz = coords

for escena in carpetas:
    dir_recorte = dir_imagenes + 'recortes/' + escena + '/'
    recortes = os.listdir(dir_recorte)
    for banda in recortes:
        if 'xyz' in banda:
            if 'B04' in banda:
                b4 = pandas.read_csv(dir_recorte + banda, sep=' ', header=None).iloc[:, 2].
                    values

                elif 'B08' in banda:
                    b8 = pandas.read_csv(dir_recorte + banda, sep=' ', header=None).iloc[:, 2].
                        values
            ndvi = (b4-b8)/(b4+b8)
            ndvi_xyz.loc[:, 2] = ndvi
            ndvi_xyz.to_csv(dir_recorte + banda[:-21] + 'idx_NDVI_recortado.xyz', sep=' ', header
                = False, index = False)

```

Listing 2: Script para crear el Dataframe y entrenar la red neuronal.

```

import pandas
import os
import numpy as np
import tensorflow as tf

path = 'G:/TFM/datos/raster/Sentinel-2/'
clasificacion = 'clasificacion/'
recortes = 'recortes/'

# Creamos el data frame de toda la imagen con las coordenadas de los centros de los píxeles y su clasificación
todo = pandas.read_csv(path + clasificacion + 'SI_NO_PNOA_EPSG32629_4.xyz', sep=' ', header
    =None, names=["X", "Y", "clase"])#.sort_values(["X", "Y"])

# Creamos una lista con los nombres de los directorios de nuestros recortes (resulta una lista con los nombres de cada imagen)
lista_escenas = os.listdir(path+recortes)

# Se leen los xyz del recorte de cada banda y se agregamos la columna con el valor de la reflectividad de cada banda.
for i in lista_escenas:
    contenido = os.listdir(path+recortes+i+'/')
    for x in contenido:
        if 'xyz' in x:
            csv = pandas.read_csv(path+recortes+i+'/' + x, sep=' ', header=None)#.sort_values
                (by=[0,1])
            todo.loc[:, x[7:30]] = csv.iloc[:, 2]

# Seleccionamos sólo las filas de los píxeles clasificados como si o no y cambiamos los no de valor 2 a valor 0
si = todo.loc[todo['clase'].isin([1])]
no = todo.loc[todo['clase'].isin([2])].loc[:, 2].assign(clase=0) # Elegir 1 de cada 4 filas de no

# Juntamos ambos dataframes
sino = si.append(no).sort_values(by=["X", "Y"])

# Creamos la matriz de características y el vector de clasificación
X = sino.iloc[:, 3:].values
y = sino.iloc[:, 2].values

# Matriz de características de toda la imagen

```

```
X2 = todo.iloc[:, 3:].values

# Dividimos el data set en conjuntos de entrenamiento y test
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state
= 0)

# Escalamos las variables para que mayores valores no distorsionen los resultados
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)

X2_scaled = sc_X.transform(X2)

# ELABORAR RED NEURONAL

from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout

# Inicializar a RNA
classifier = Sequential()

# Agregar la capa de entrada y las capas ocultas con un Dropout del 10%.
classifier.add(Dense(128, kernel_initializer = 'uniform',
activation = tf.keras.layers.PReLU(), input_dim = np.prod(X_test.shape[1])))

classifier.add(Dropout(rate=0.1))

classifier.add(Dense(64, kernel_initializer = 'uniform',
activation = tf.keras.layers.PReLU()))
classifier.add(Dropout(rate=0.1))

classifier.add(Dense(32, kernel_initializer = 'uniform',
activation = tf.keras.layers.PReLU()))

classifier.add(Dropout(rate=0.1))

classifier.add(Dense(units = 16, kernel_initializer = 'uniform',
activation = tf.keras.layers.PReLU()))
classifier.add(Dropout(rate=0.1))

classifier.add(Dense(units = 8, kernel_initializer = 'uniform',
activation = tf.keras.layers.PReLU()))
classifier.add(Dropout(rate=0.1))

# Crear capa de salida
classifier.add(Dense(units = 1, kernel_initializer = 'uniform',
activation = 'sigmoid'))

# Compilar la RNA
classifier.compile(optimizer = "adam", loss = "binary_crossentropy",
metrics = ["accuracy"])

# Devolver el clasificador
history = classifier.fit(X_train, y_train, epochs = 1000, batch_size = 8)
```

```

# Elaborar predicción del conjunto de test.
y_pred = classifier.predict(X_test)
y_pred = (y_pred > 0.99)

# Generar la matriz de confusión.
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)

# Hacer una predicción sobre el el conjunto de datos a clasificar (clasificación)
imaxe_clasificada = classifier.predict(X2_scaled)

# Transformamos el numpy array en un dataframe de una sola columna.
imaxe_clasificada = pandas.DataFrame(imaxe_clasificada, columns = ['clase'])

# Concatenamos el vector de clasificación al final de la matriz de la imagen clasificada
imaxe_completa = pandas.concat([todo, imaxe_clasificada], axis = 1)

# Seleccionamos las columnas que formarán la capa clasificada (coordenadas y clasificaci
ón)
resultado_final = imaxe_completa.iloc[:, [0, 1, len(si.columns)]]

# Para cargar el .csv en QGIS no debe tener cabecera ni columna de índices.
resultado_final.to_csv(path+clasificacion+'
    capa_clasificada_DEFINITIVA_prelu_sigmoid_1000_3img.xyz', sep=' ', header = False,
    index = False)

# Se guarda el modelo
classifier.save(path+clasificacion+'modelo_202207004.h5')

# Imprimimos los plots generados con los datos de precisión y coste.
from matplotlib import pyplot as plt
print(history.history.keys())
plt.plot(history.history['accuracy'])
print('algo para separar los plots')
plt.plot(history.history['loss'])

```

Listing 3: GridSearchCV.

```

from sklearn.model_selection import GridSearchCV

classifier = KerasClassifier(build_fn = build_classifier)

parameters = {
    'batch_size' : [4, 8, 16],
    'epochs' : [500, 1000, 2000],
    'optimizer' : ['adam']
}

grid_search = GridSearchCV(estimator = classifier,
    param_grid = parameters,
    scoring = 'accuracy')

grid_search = grid_search.fit(X_train, y_train)

best_parameters = grid_search.best_params_
best_accuracy = grid_search.best_score_

```

Listing 4: k-fold Cross Validation.

```

classifier = KerasClassifier(build_fn = clasificador, batch_size = 8, epochs = 1000)
accuracies = cross_val_score(estimator = classifier, X = X_train, y = y_train, cv = 10)

```

```
summary = pandas.DataFrame(accuracies)
summary.describe()
```

Listing 5: Extracción de estadísticas adaptadas para tablas de latex.

```
import pandas

path = 'G:/TFM/datos/raster/Sentinel-2/clasificacion/'

file1 = 'poligonos_filtro_mas_5px_tabla.csv'
file2 = 'poligonos_filtro_menos_5px_tabla.csv'

tabla1 = pandas.read_csv(path+file1, sep=',')
tabla2 = pandas.read_csv(path+file2, sep=',')

tabla1 = tabla1.rename(columns={'sup_pnoa': 'clase'})
tabla1 = tabla1[['clase', 'area']]

tabla2 = tabla2[['clase', 'area']]

def listAreas(tabla):
    summary = tabla.groupby('clase').describe()
    area = tabla['area'].sum()
    listArea = []
    listPercent = []
    for i in range(3):
        sub_i = (tabla[tabla['clase']==i])
        area_i = sub_i['area'].sum()
        listArea.append(area_i)
        listPercent.append(area_i/area*100)
    listAreas = [listArea, listPercent]

return listAreas, summary

(menos500, summary500) = listAreas(tabla2)
(mas500menos1000, summary500y1000) = listAreas(tabla1[tabla1['area'] <= 1000])
(mas1000, summary1000) = listAreas(tabla1[tabla1['area'] >= 1000])
(mas500, summary_mas500) = listAreas(tabla1)

df = pandas.DataFrame(columns= ['mal clasificados', 'bien clasificados', 'sin definir'])
lista_df = []
dic={'<500':menos500, '500-100':mas500menos1000, '>10000':mas1000, '>500':mas500}
for grupo in dic:
    df_i = pandas.DataFrame(dic[grupo], columns= ['mal clasificados', 'bien clasificados',
        'sin definir'], index=['superficie '+grupo, 'porcentaje '+grupo])
    data = [df, df_i]
df = pandas.concat(data)

print(summary500.to_latex(index=[]))
print(summary500y1000.to_latex(index=[]))
print(summary1000.to_latex(index=[]))
print(summary_mas500.to_latex(index=[]))
```

B. Anexo II

A continuación se incorpora la cartografía a escala 1/25.000 en la que se muestran los resultados de la red para las agrupaciones de entre 5 y 10 píxeles (naranja) y más de 10 de píxeles (rojo), representados sobre las ortofotos del PNOA del Instituto Geográfico Nacional (IGN). También se muestra un plano de situación donde se marca en rojo la hoja representada en cada página y en amarillo las restantes. Este mapa se encuentra a escala 1/500.000 y como base cuenta con una combinación del mapa LiDAR y el mapa de relieve del IGN.

HOJA - A1

753000.000E

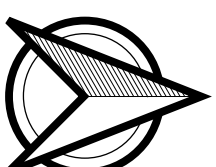
754000.000E

755000.000E

756000.000E

757000.000E

758000.000E



Presencia de Cortaderia

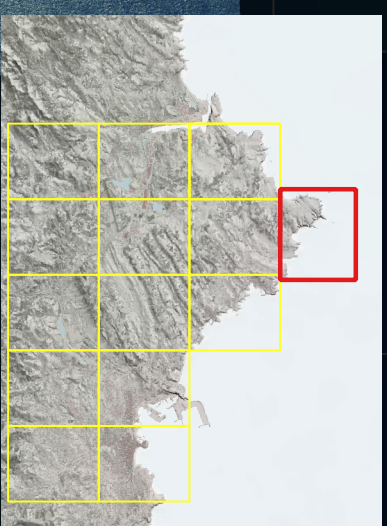
Filtro 1000 m²

> 90% de probabilidad



Filtro 500-1000 m²

50% aprox. de probabilidad



1:25.000

0 250 500 750 1.000 m

WGS 84 / UTM zona 29N

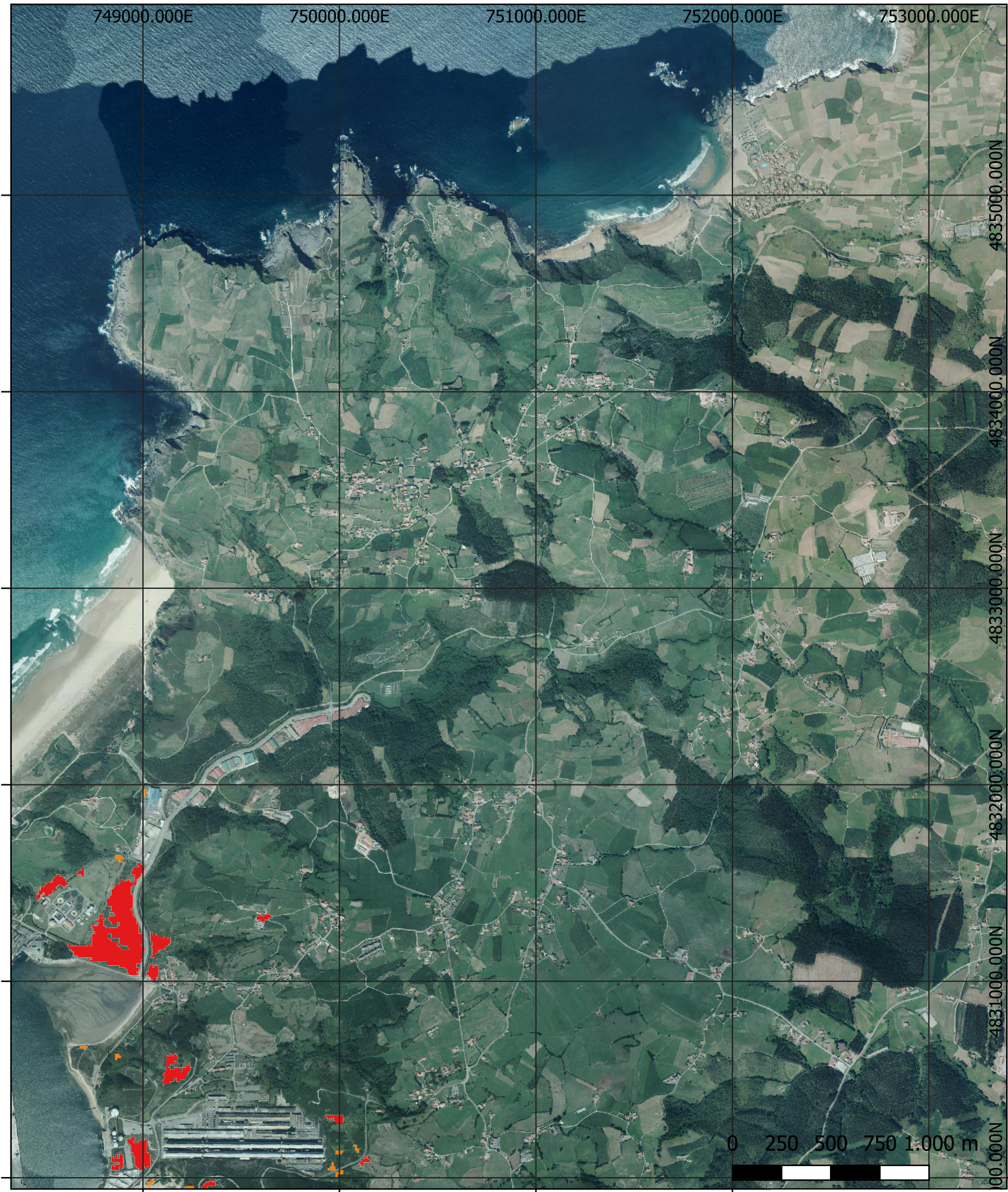
4837000.000N

4838000.000N

4839000.000N

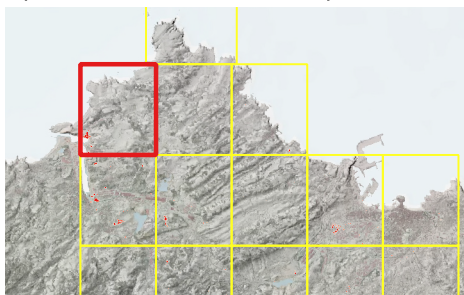
4840000.000N

HOJA - B1



1:25.000

WGS 84 / UTM zona 29N




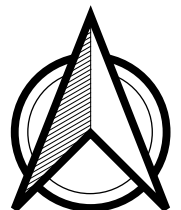
Presencia de Cortaderia

Filtro 1000 m²

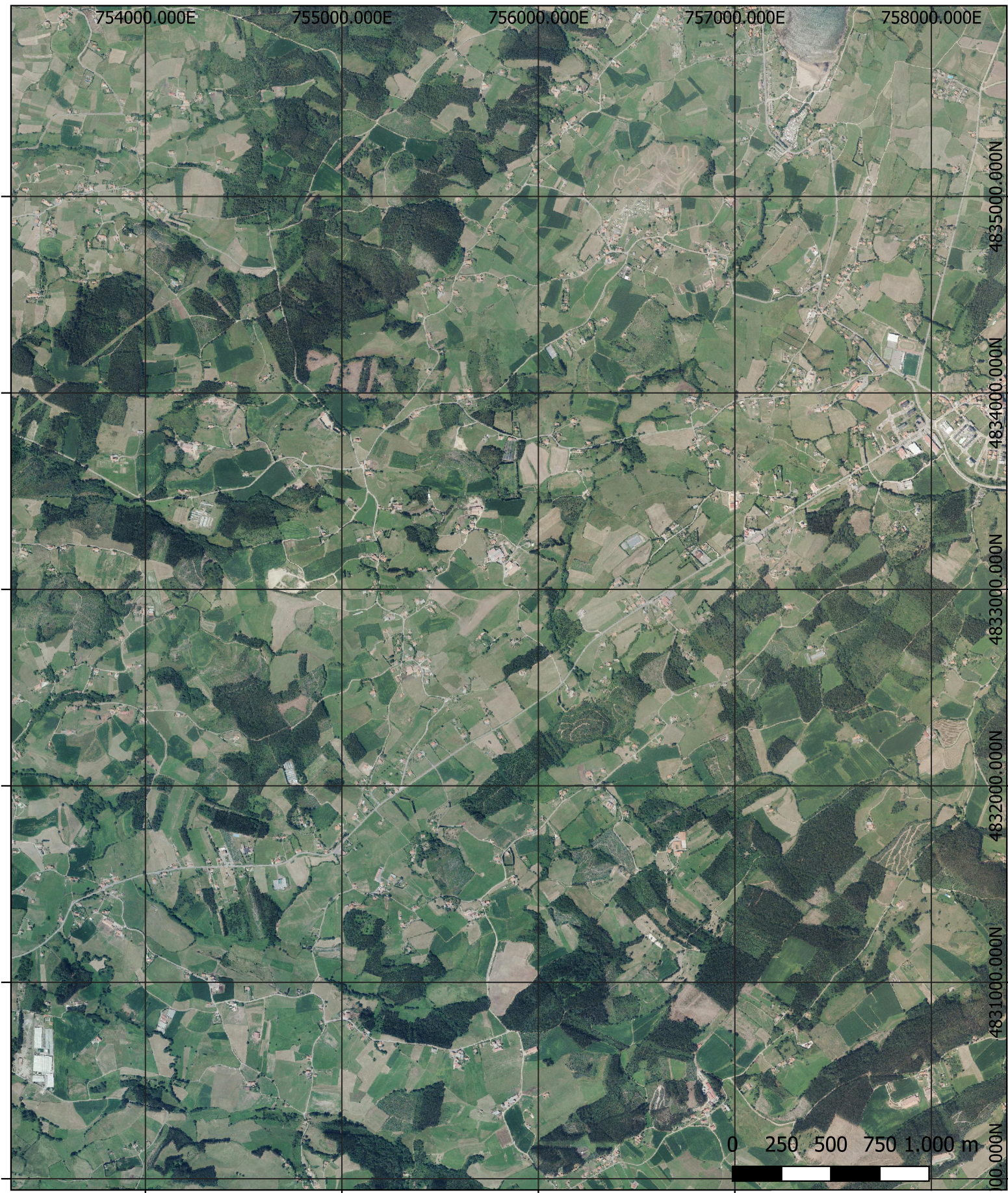
 > 90% de probabilidad

Filtro 500-1000 m²

 50% aprox. de probabilidad

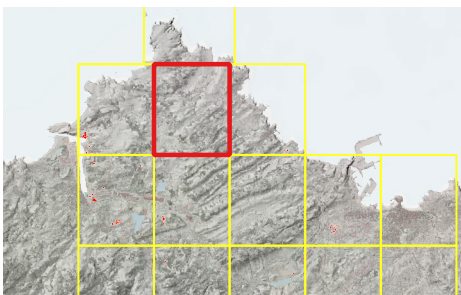


HOJA - B2



1:25.000

WGS 84 / UTM zona 29N




Presencia de Cortaderia

Filtro 1000 m²

 > 90% de probabilidad

Filtro 500-1000 m²

 50% aprox. de probabilidad

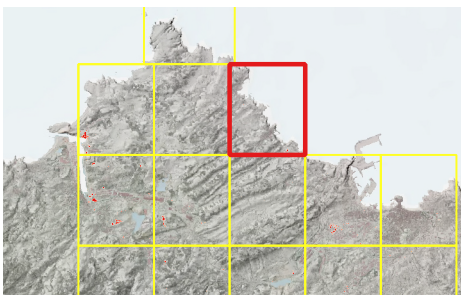


HOJA - B3



1:25.000

WGS 84 / UTM zona 29N




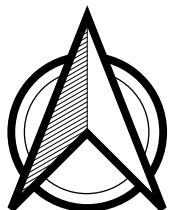
Presencia de Cortaderia

Filtro 1000 m²

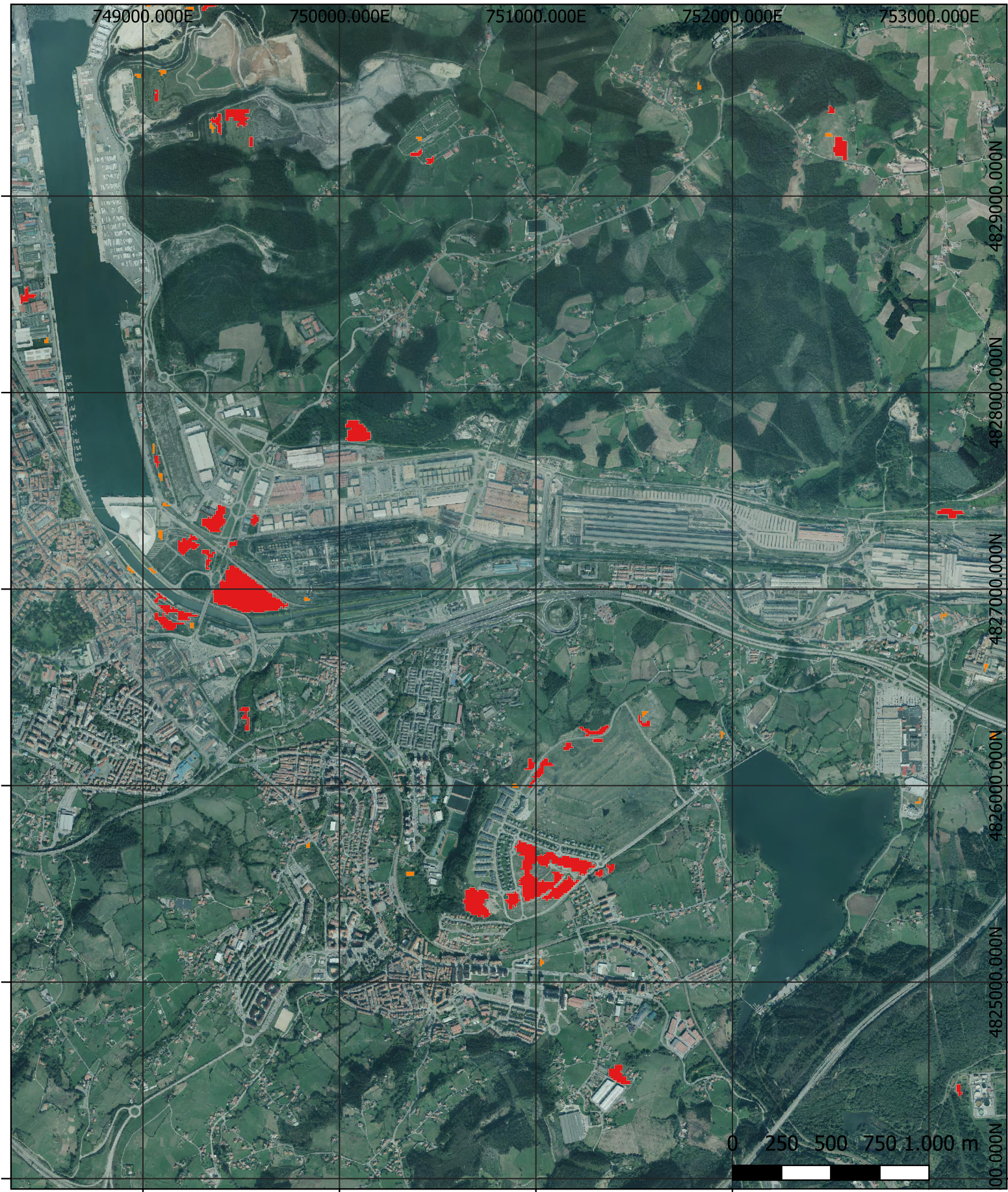
 > 90% de probabilidad

Filtro 500-1000 m²

 50% aprox. de probabilidad

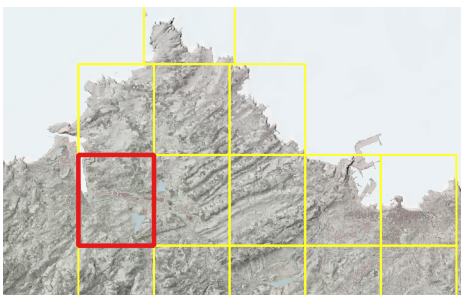


HOJA - C1



1:25.000

WGS 84 / UTM zona 29N




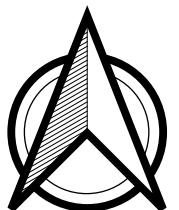
Presencia de Cortaderia

Filtro 1000 m²

 > 90% de probabilidad

Filtro 500-1000 m²

 50% aprox. de probabilidad

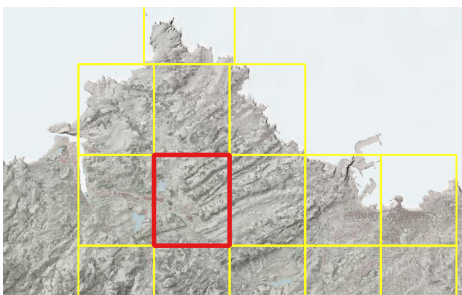


HOJA - C2



1:25.000

WGS 84 / UTM zona 29N




Presencia de Cortaderia

Filtro 1000 m²

 > 90% de probabilidad

Filtro 500-1000 m²

 50% aprox. de probabilidad

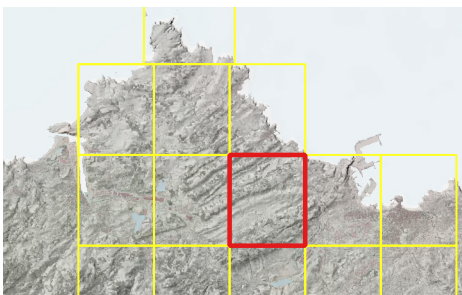


HOJA - C3



1:25.000

WGS 84 / UTM zona 29N




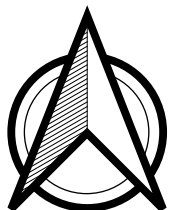
Presencia de Cortaderia

Filtro 1000 m²

 > 90% de probabilidad

Filtro 500-1000 m²

 50% aprox. de probabilidad

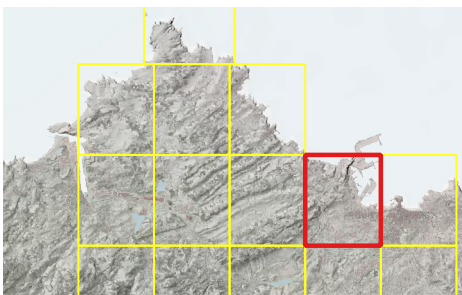


HOJA - C4



1:25.000

WGS 84 / UTM zona 29N




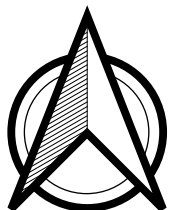
Presencia de Cortaderia

Filtro 1000 m²

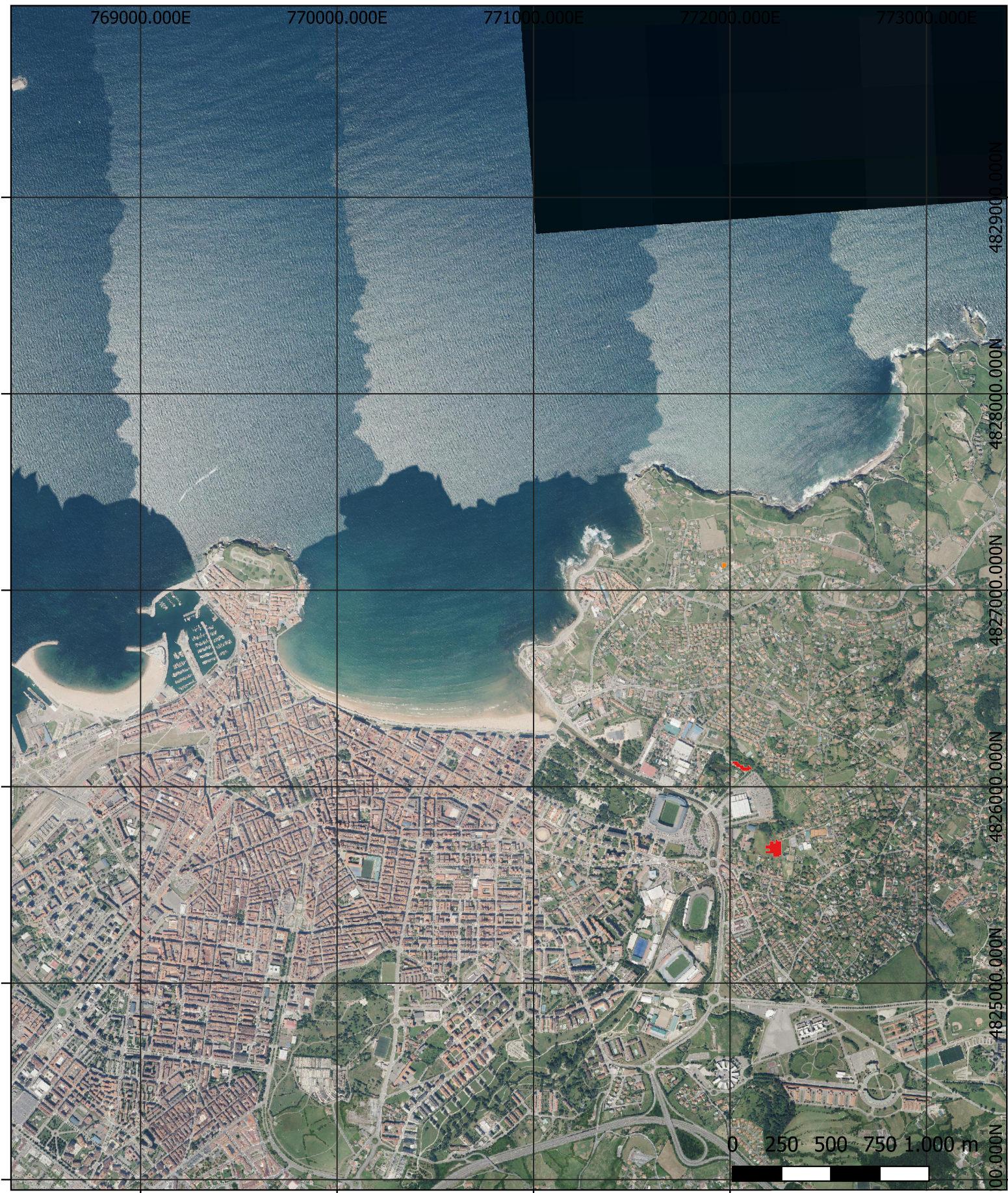
 > 90% de probabilidad

Filtro 500-1000 m²

 50% aprox. de probabilidad

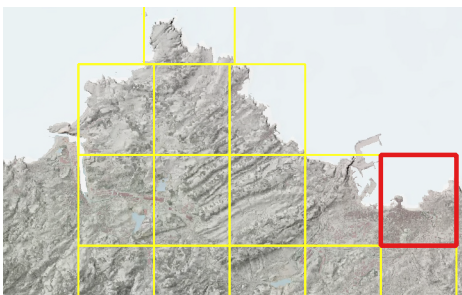


HOJA - C5



1:25.000

WGS 84 / UTM zona 29N




Presencia de Cortaderia

Filtro 1000 m²

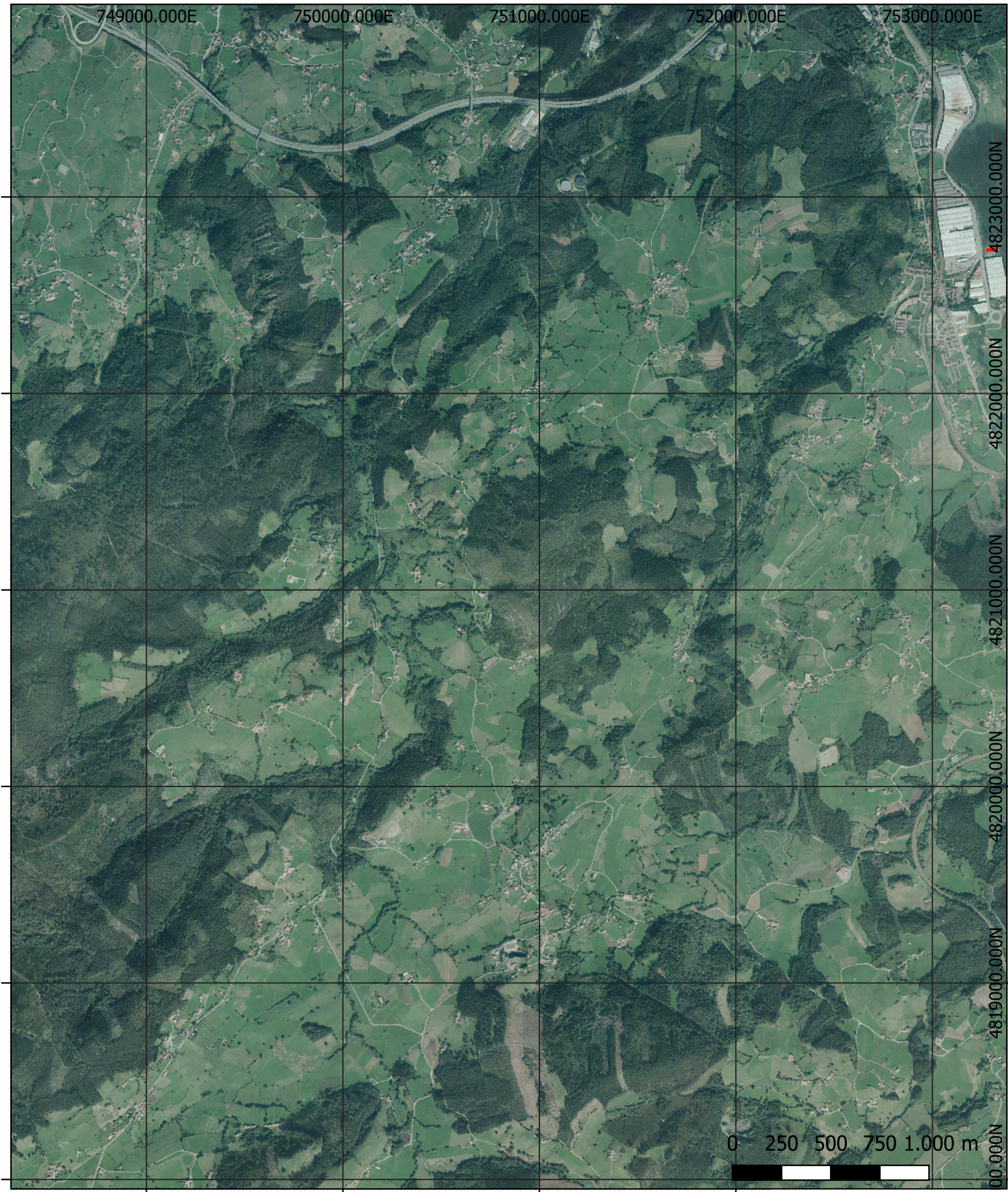
 > 90% de probabilidad

Filtro 500-1000 m²

 50% aprox. de probabilidad



HOJA - D1



1:25.000

WGS 84 / UTM zona 29N




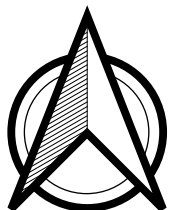
Presencia de Cortaderia

Filtro 1000 m²

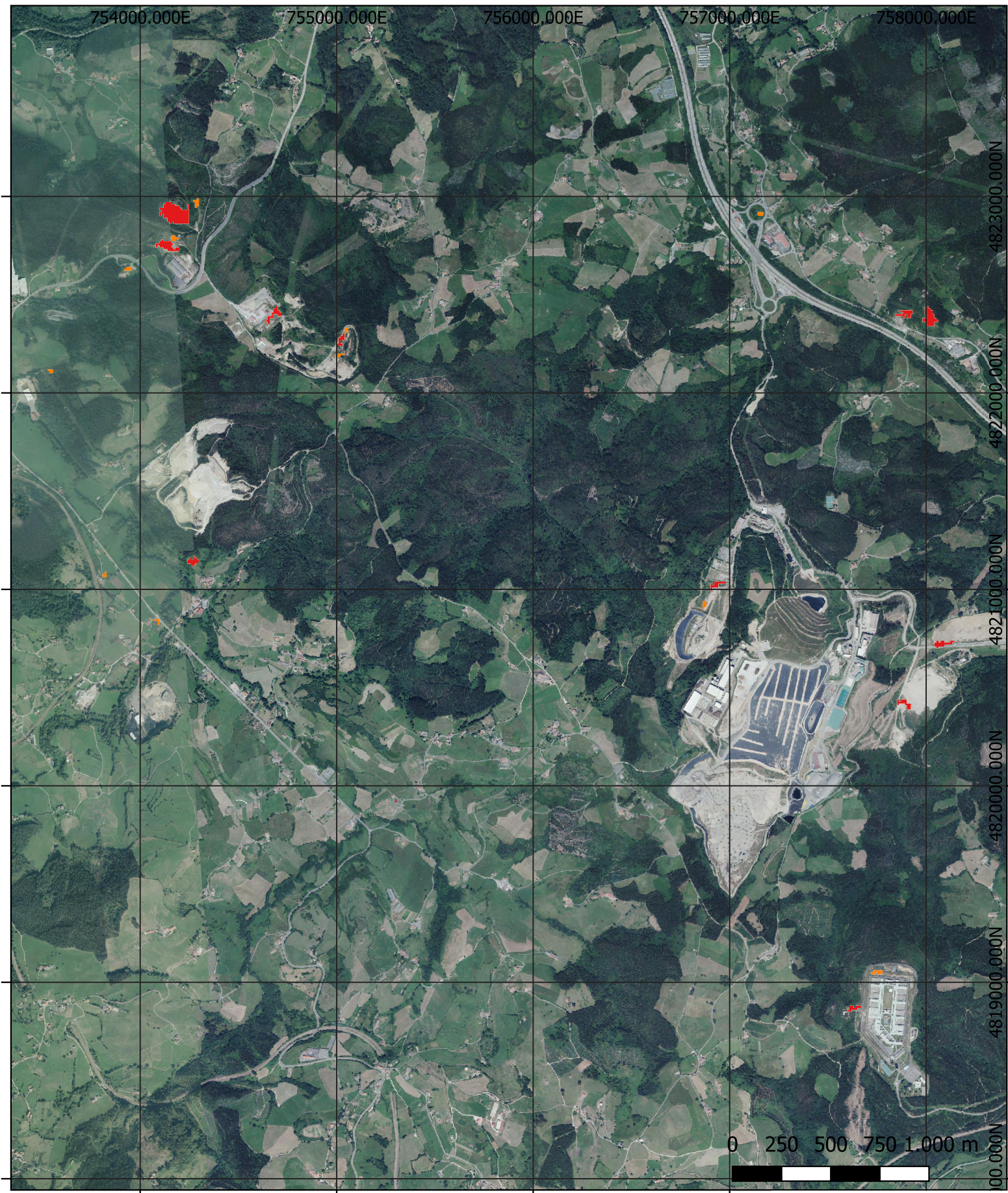
 > 90% de probabilidad

Filtro 500-1000 m²

 50% aprox. de probabilidad



HOJA - D2



1:25.000

WGS 84 / UTM zona 29N




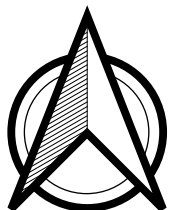
Presencia de Cortaderia

Filtro 1000 m²

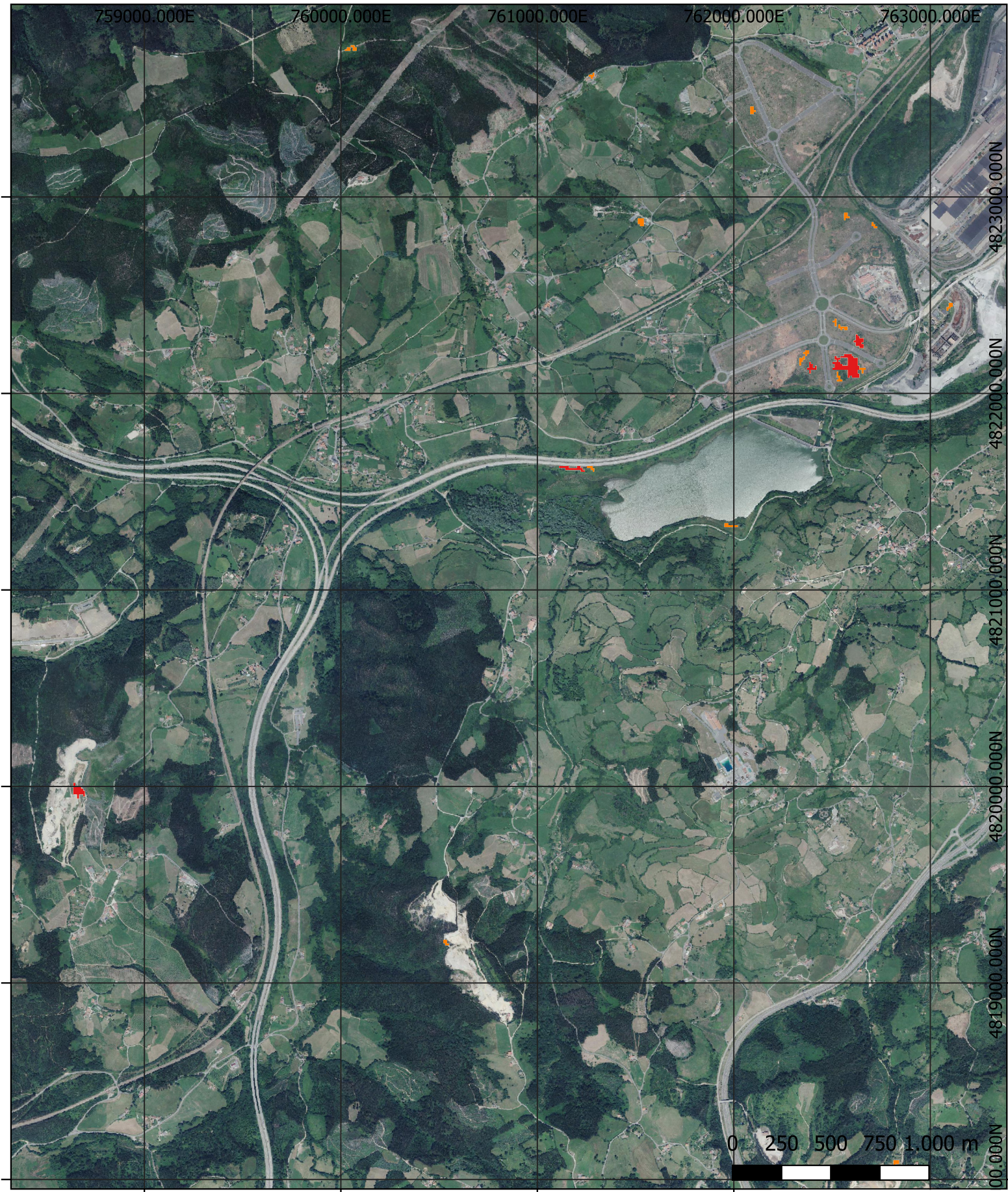
 > 90% de probabilidad

Filtro 500-1000 m²

 50% aprox. de probabilidad

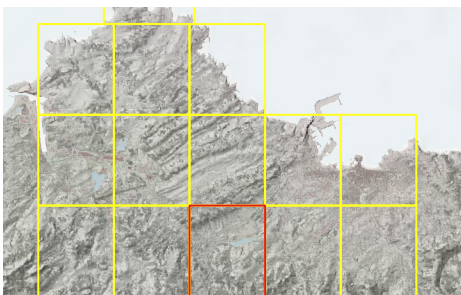


HOJA - D3



1:25.000

WGS 84 / UTM zona 29N




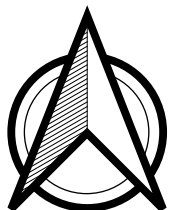
Presencia de Cortaderia

Filtro 1000 m²

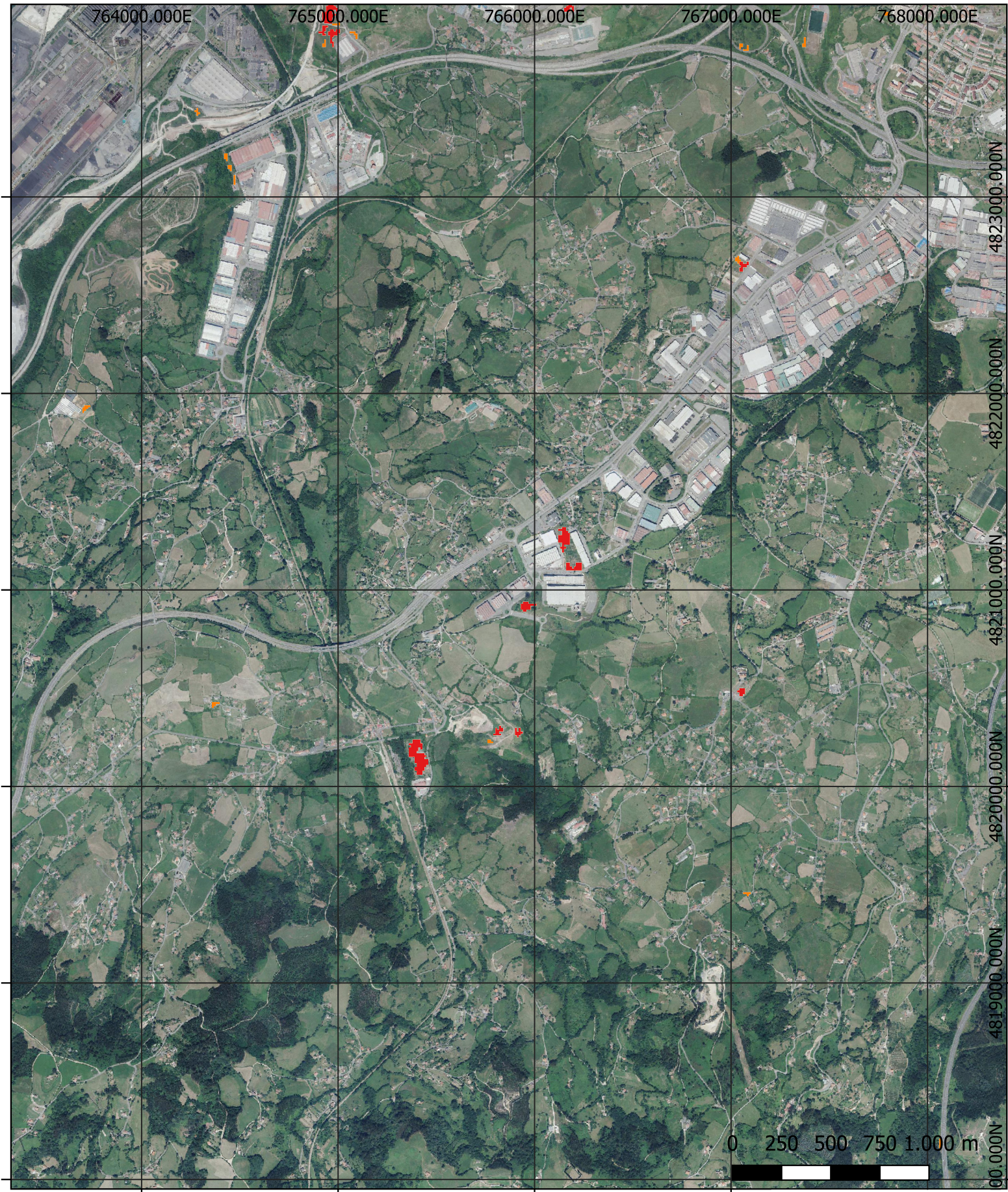
 > 90% de probabilidad

Filtro 500-1000 m²

 50% aprox. de probabilidad

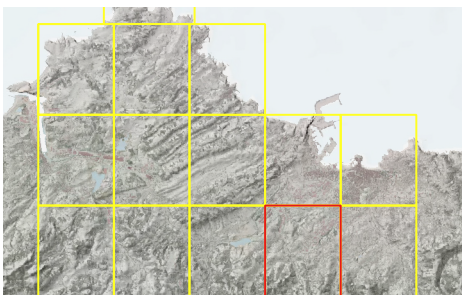


HOJA - D4



1:25.000

WGS 84 / UTM zona 29N




Presencia de Cortaderia

Filtro 1000 m²

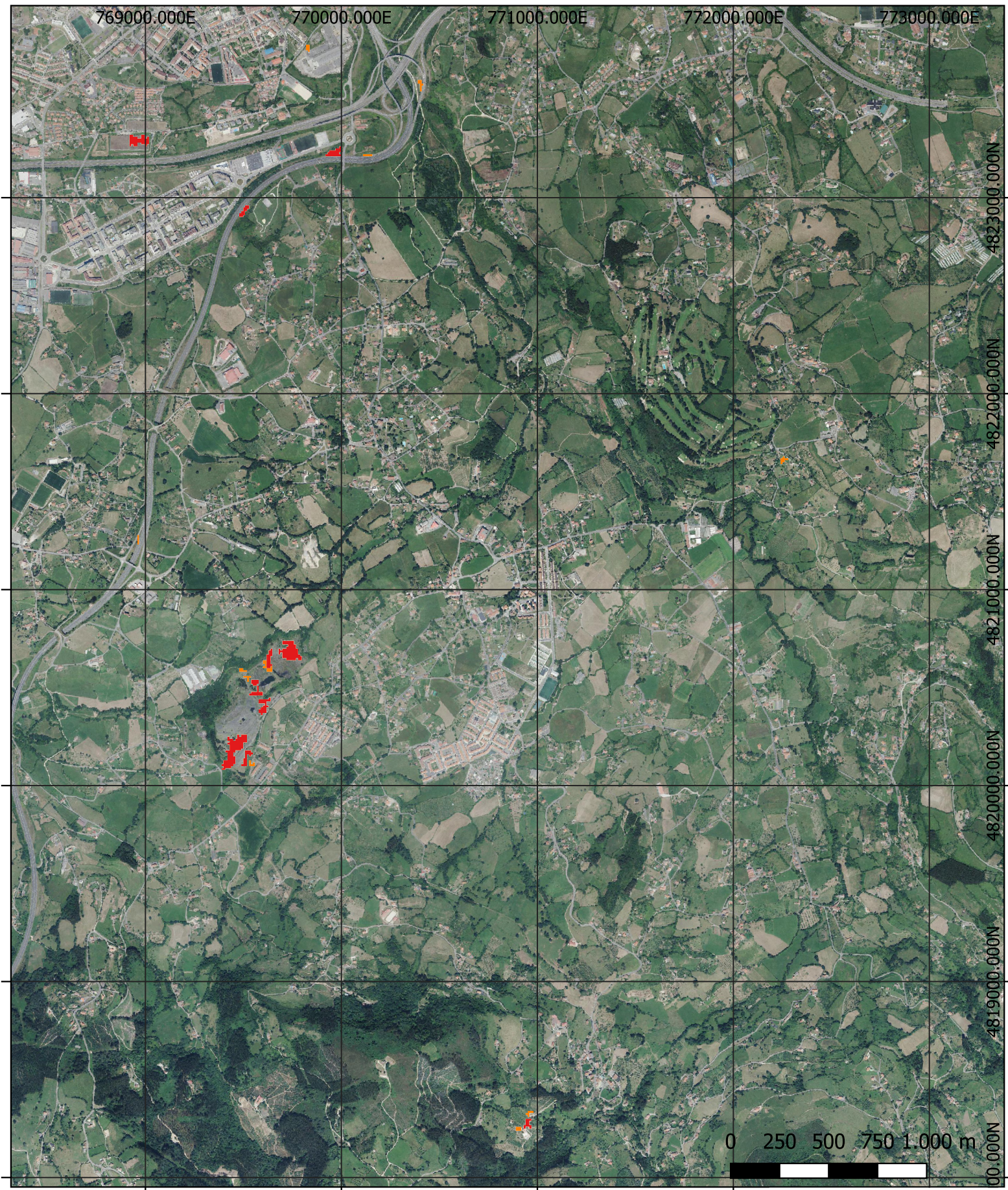
 > 90% de probabilidad

Filtro 500-1000 m²

 50% aprox. de probabilidad



HOJA - D5



1:25.000

WGS 84 / UTM zona 29N




Presencia de Cortaderia

Filtro 1000 m²

 > 90% de probabilidad

Filtro 500-1000 m²

 50% aprox. de probabilidad

