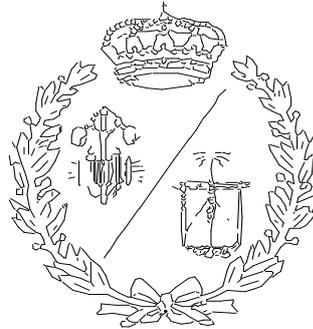


**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN**

UNIVERSIDAD DE CANTABRIA



Proyecto Fin de Grado

**IMPLEMENTACIÓN DE UNA HERRAMIENTA
DE DETECCIÓN DE CORTADERIA SELLOANA
EN ORTOFOTOS MEDIANTE TÉCNICAS DE
APRENDIZAJE PROFUNDO**

**(Implementation of a technique for detecting
cortadería selloana in ortophotos using deep
learning methods)**

Para acceder al Título de

**GRADUADO EN INGENIERÍA ELECTRÓNICA
INDUSTRIAL Y AUTOMÁTICA**

**Autor: Álvaro Antolín Diego
junio - 2025**

Resumen

En el norte de España y más concretamente en Cantabria existe un gran problema medioambiental con una planta invasora, la cortaderia selloana, un plumero originario de la Pampa sudamericana introducido como especie ornamental. Ocupa ambientes urbanos y rurales, mientras perjudica el crecimiento de especies autóctonas, tanto vegetales como animales, en espacios de gran valor ecológico. Debido a su capacidad de colonización y a la falta de concienciación ciudadana, su erradicación es complicada.

En este proyecto se propone desarrollar una herramienta de detección de esta especie mediante el uso de ortofotos (procedentes de sitios web como SigPac o PNOA), con el objetivo de facilitar su posterior erradicación. Esto simplificará la adopción de medidas prioritarias en las zonas más afectadas, de acuerdo con los resultados que se obtengan. Para ello, se emplearán técnicas de aprendizaje profundo para desarrollar metodologías de detección y segmentación de plumero, mediante redes neuronales y máquinas de vectores de soporte. Se realizará un estudio de las diferentes redes neuronales para elegir la opción que obtenga mejores resultados de detección.

Abstract

In the north of Spain and more specifically in Cantabria, there is a huge environmental problem related to an invasive plant called "cortaderia selloana", which is a Pampas grass from La Pampa, South America, introduced as an ornamental plant. It invades urban places and rural environments, while it harms the growth of native species, such as animals or plants, in places of big ecological value. Due to its colonization ability and the lack of public awareness, the eradication is difficult.

This project proposed the development of a screening tool of this vegetation by using orthophotos (from websites like SigPac or PNOA), with the aim of facilitating its further eradication. It simplifies the adoption of priority measures in the most affected areas, according to the obtained results. Consequently, different deep learning techniques will be used to develop Pampa grass' detection and segmentation methods, through neural networks and support vector machines. An analysis of the alternative neural networks will be made to select the option with the best detection results.

INDICE

1. Motivación y objetivos
2. Introducción
3. Obtención de datos
 - 3.1. Introducción
 - 3.2. stopcortaderia.org
 - 3.3. SigPac
 - 3.4. Sentinel Hub
 - 3.5. PNOA
 - 3.6. mapas.cantabria.es
 - 3.7. Generación del dataset y conclusiones
4. Redes neuronales biológicas
5. Redes neuronales artificiales
 - 5.1. Introducción
 - 5.2. Alexnet
 - 5.3. Squeezenet
 - 5.4. Redes residuales
 - 5.5. Shufflenet
 - 5.6. Darknet
 - 5.7. Conclusiones
6. SVM
 - 6.1. Introducción
 - 6.2. Conclusiones
7. Entrenamiento
 - 7.1. Introducción
 - 7.2. Entrenamiento de Squeezenet
 - 7.3. Entrenamiento de Alexnet
 - 7.4. Entrenamiento de ResNet50
 - 7.5. Entrenamiento de SVM
 - 7.6. Conclusiones
8. Fase de inferencia
 - 8.1. Introducción
 - 8.2. Inferencia mediante Alexnet

8.3. Inferencia mediante ResNet50

8.4. Conclusiones

9. Segmentación

9.1. Introducción

9.2. FCN

9.3. U-Net

9.4. PSPNet

9.5. SegNet

9.6. Otros métodos

9.7. Conclusiones

10. Desarrollo de la herramienta software

10.1. Introducción

10.2. Herramienta software global

10.3. Herramienta software para Cantabria

10.4. Conclusiones

11. Comparativa de los resultados

12. Bibliografía

13. Referencias de figuras

1. MOTIVACIÓN Y OBJETIVOS

El propósito de este trabajo es desarrollar una herramienta de fácil uso mediante la cual un usuario pueda saber si en un determinado lugar existe cortaderia selloana, sin necesidad de tener conocimientos relativos a dicha planta o a la inteligencia artificial. A su vez, gracias a la fácil empleabilidad de la herramienta, se desea favorecer a la eliminación de ejemplares de plumero de la Pampa, de manera que no supongan posibles problemas de índole sanitaria o económica a la sociedad.

Dentro de las diferentes opciones que poseen los programas, se podrá evaluar una imagen individual para determinar si posee cortaderia selloana, se permitirá obtener mapas donde se recopilen diferentes localizaciones con esta especie invasora, pudiendo usarse tanto para todo el planeta como para la autonomía de Cantabria.

2. INTRODUCCIÓN

En la actualidad, se cifran en más de 50 las plantas consideradas como especies exóticas invasoras en el territorio español. Una de estas especies más comunes y peligrosas, que a diferencia de otras como la palmera datilera o el madroño que sólo presentes en el archipiélago canario, es la cortaderia selloana, cuya presencia abarca toda la superficie nacional.

La cortaderia selloana, también conocida como plumero de la Pampa o cola de zorro, es una especie de cortaderia originaria de Sudamérica, especialmente de zonas de la Pampa argentina y el sureste de Uruguay, junto con zonas costeras de Chile. Se introdujo en el continente europeo comenzando por el Reino Unido en el siglo XVIII. En ese momento, sus finalidades eran la nutrición de los ganados en granjas y en especial, el uso como decoración en el interior de las casas y jardines, propósito que aún continúa hoy en día. Es considerada una de las plantas invasoras más peligrosas de la Europa bañada por el Océano Atlántico, además de Australia, África, América y otras zonas del viejo continente. En el caso de España, se ha acabado extendiendo a todo el territorio, especialmente en el norte, incluyendo en los archipiélagos canario y balear, lo que ha provocado que se considere como la planta invasora más peligrosa de toda la nación. Los primeros estudios nacionales relativos a esta planta surgieron en el año 1953, casualmente debido a ejemplares encontrados en la bahía de Santander, aunque se sospecha que pudo llegar al país con anterioridad, en la década de 1940, en forma de semillas procedentes de Argentina mezcladas con cargamentos de cereal.

A pesar de tener un origen geográficamente concreto, es una planta que ha logrado colonizar la gran mayoría del planeta. Según registros que se conservan y fruto del desconocimiento

que existía en ese momento, se sabe que la cortaderia selloana se consideraba una especie naturalizada en lugares como California hace casi un siglo, gracias a su comercialización. Esta actividad favoreció su normalización y expansión, hasta llegar a invadir superficies de terreno de más de 200 Km² por año en esta zona. Sabedoras del problema, las principales autoridades gubernamentales han decidido prohibir su venta y su inclusión en los catálogos nacionales de especies invasoras. No obstante, son muchos los negocios de floristerías y herboristerías donde aún se vende el plumero de la Pampa como hierba ornamental.

Los ejemplares de esta planta pueden ser tanto femeninos como hermafroditas. Al comienzo de su asentamiento en un territorio, no se considera un peligro para el ecosistema en cuestión porque los ejemplares no son capaces de competir por los recursos de otras plantas. Sin embargo, cuando alcanza la madurez a los 3 años, aumenta con creces su capacidad de colonización y ocupación del terreno. Para ello, la cortaderia selloana se ve ayudada por la formación de macollas en su parte inferior, lo cual aumenta ostensiblemente su extensión. Todo esto hace que se considere una especie muy robusta, que puede llegar a vivir hasta los 15 años.

La forma del plumero de la Pampa ayuda a su proliferación, ya que no necesita de una gran área para asentarse. Estas plantas perennes pueden alcanzar los 4 metros de altura y abarcar un diámetro de 3.5 metros. Además de la altura visible de la planta, es de igual importancia la morfología de su raíz, ya que también cuenta con un alcance similar bajo tierra, siendo capaz de sustraer el agua y las sales minerales de otras plantas cercanas. Este alcance, además de su fácil propagación, es una de las claves de su asentamiento tanto en el territorio nacional como en el resto de los países, pudiendo abarcar volúmenes de tierra superiores a los 30 m³.



The image shows a product listing for 'Plumeros de Cortaderia Selloana (Pampas) Pack 5 uds.' on an e-commerce platform. The main image shows a person holding several large, dried pampas grass bundles. Below the main image are four smaller thumbnail images showing different uses of the grass. The product title is 'Plumeros de Cortaderia Selloana (Pampas) Pack 5 uds.' with a brand name 'Flores Frescas'. It has a 5-star rating and 8 reviews. The price is 22,50 €. A 'Descuentos por cantidad' section shows three options: 1 unit for 22,50 € (recommended), 10 units for 16,50 €/unit, and 2 units for 18,50 €/unit. All options include free shipping. The page also features a quantity selector set to 1, an 'Añadir al carrito' button, and shipping information: 'Impuestos incluidos', 'Recíbelo en 1 día laborable', and 'Envío gratis para España (península)'.

Plumeros de Cortaderia Selloana (Pampas) Pack 5 uds.

Marca: Flores Frescas

★★★★★ Ver opiniones (8)

22,50 €

Descuentos por cantidad

1 unidad 22,50 € Portes gratis	Ahorra 60,00 € 10 unidades 16,50 €/unidad Portes gratis RECOMENDADO	Ahorra 8,00 € 2 unidades 18,50 €/unidad Portes gratis
--------------------------------------	---------------------------------------------------------------------------------	----------------------------------------------------------------

5/5

← 1 →

🛒 Añadir al carrito

📄 Impuestos incluidos 📅 Recíbelo en 1 día laborable 🚚 Envío gratis para España (península)

Figura 1. Captura de una página web que vende ejemplares de cortaderia selloana

Fuente: [1]

Su capacidad de adaptación a diferentes condiciones ambientales durante la edad adulta, como la sequía o ecosistemas salinos, aprovechando la humedad ambiental y el rocío o temperaturas bajas, incluso por debajo de -15°C , hace que evitar su propagación sea mucho más complicado. Incluso, sus semillas son capaces de desplazarse a lo largo de kilómetros, debido a su ligereza, al viento y a su fácil adhesión a cualquier tipo de superficie, como vehículos, ropa, calzado, etc. Esta fácil expansión del plumero de la Pampa ha provocado estimaciones superiores al millón de ejemplares de esta planta en una comunidad autónoma. No obstante, aunque esta especie crece más comúnmente en lugares frescos y húmedos, su capacidad de adaptación ha hecho que aparezca en zonas alejadas de cualquier litoral, donde sus condiciones medioambientales son más desfavorables. Otra problemática de este tipo de vegetación es su habilidad para colonizar hábitats abiertos o sin grandes fuentes de agua cercanas, y establecerse como especie dominante en dichos lugares, siendo una clara amenaza para el ecosistema en cuestión. No obstante, es difícil de encontrar en ecosistemas estables a lo largo del tiempo como, por ejemplo, bosques. El abandono de instalaciones urbanas e industriales, así como la falta de cuidados de los ecosistemas naturales, son factores cruciales a la hora del crecimiento de esta especie invasora.



Figura 2. Población de cortaderia selloana cercana a una carretera

Fuente: [2]

Su alto consumo de agua hace que el terreno donde se encuentre se seque, pero sin quedar yermo, lo cual puede perjudicar los cultivos y el resto de la vegetación circundante. Esto supone un riesgo que puede provocar la desaparición de otras especies de plantas y la destrucción de ecosistemas fluviales y la flora cercana, así como la posible extinción de los animales que se nutriesen de esas plantas extintas, y otros seres vivos de su cadena trófica. Todo esto implica una amenaza para la biodiversidad de la cual es muy difícil que un terreno invadido por la cortaderia selloana se recupere en el futuro.

Además de los problemas anteriormente expuestos, la propagación del polen de esta planta puede también afectar a la salud de las personas, en forma de alergias respiratorias y problemas derivados de las mismas, como congestión nasal, rinoconjuntivitis, o incluso en los casos más graves, asma. No obstante, también cuenta con algún beneficio, como por ejemplo evitar el riesgo de desprendimientos en taludes de carreteras con pendientes elevadas, ayudando a la estabilización del terreno, y su uso para erradicar procesos febriles en niños, al ser también sudorífera y además, ser aprovechada con fines diuréticos. También es empleada para conseguir celulosa o materia prima para la fabricación de cestos y sombreros.

Posiblemente, el problema más grave de la cortaderia selloana en el mundo actual es que, a pesar de sus problemáticas, es una planta ampliamente normalizada por la población, tanto en ambientes urbanos (caminos, arcenes de carreteras y autopistas, puertos, aeropuertos, polígonos industriales, urbanizaciones, etc) como en ambientes rurales (bordes de ríos, lagos, torrentes, explotaciones mineras, acantilados, zonas de campiña, etc). El desconocimiento generalizado existente en gran parte de la sociedad sobre estos peligros del plumero de la Pampa provoca que la respuesta de la ciudadanía ante esta emergencia, en caso de producirse, no esté correctamente coordinada y no sea eficiente a pesar de realizarse campañas de concienciación a nivel municipal, autonómico o estatal. Mediante estas campañas, las administraciones buscan erradicar los ejemplares de cortaderia selloana que se encuentren en su territorio, involucrando a la población.

Existen diferentes formas de eliminación del plumero, dependiendo de los medios materiales y económicos de los que se disponga, así como del tamaño de las plantas. La mejor época para la eliminación de estas plantas es a mediados de verano porque no se dispersan tantas semillas como en otras estaciones y se dificulta el rebrote. Este proceso de eliminación del plumero de la Pampa debe realizarse por operarios debidamente equipados con protección ocular, facial y manual, ya que sus hojas son cortantes. También es recomendable ayudarse de sierras mecánicas para eliminarlas. Con respecto al proceso, junto con el corte de la parte aérea de la planta, es necesario arrancar la raíz del suelo para evitar un rebrote. Para este procedimiento, usualmente se cava a su alrededor o se usan equipos más complejos como

un tractor. También es habitual el empleo de herbicidas para evitar que vuelva a surgir, aunque es un método más invasivo y dañino para el resto de la flora circundante. La incorrecta eliminación del plumero junto con la inadecuada deposición de sus restos una vez se elimine la planta son causas importantes de rebrote de esta planta invasora.

Además de su eliminación, una de las soluciones que se está comenzando a desarrollar es el cultivo de nuevos ejemplares en los terrenos invadidos. Posibles nuevos cultivos, especialmente de ejemplares de arbolado autóctono que sean compatibles con el resto de la fauna y de la flora para no desestabilizar más el ecosistema en cuestión, pueden contribuir a la recuperación del territorio donde se había asentado la cortaderia selloana. No obstante, esta solución no siempre puede realizarse, ya que no todos los terrenos se pueden repoblar de árboles y en caso de que conseguirse, pueden provocar a su vez nuevos daños en dicho lugar, como la alteración de cadenas tróficas. Más aún, es una solución que está todavía estudiándose y aunque se haya realizado en algún lugar de manera reciente, su implementación en un determinado terreno ha de ser analizada a lo largo del tiempo para juzgar su viabilidad.

Desafortunadamente, además de ser capaz de sobrevivir en condiciones climatológicas adversas, se sabe que el plumero de la Pampa es capaz de rebrotar en caso de incendio y sus hojas son altamente inflamables, por lo que una posible desinformación respecto a este tema puede traer consigo graves consecuencias para los ecosistemas cercanos. Es más, el desenlace en términos de ejemplares de cortaderia selloana puede ser mucho peor en caso de incendio, porque la flora cercana al plumero de la Pampa no rebrotaría y su futura expansión se vería facilitada, al ser capaz de acaparar más recursos hídricos. Referido a su relación con el mundo animal, no es una planta de la cual se nutran los animales, ya que los bordes de sus hojas son aserrados y cuentan con cristales de sílice, dificultando su digestión. Esto favorece claramente su proliferación en ecosistemas naturales, de manera generalizada, independientemente de su geolocalización, su clima o la humedad de su entorno.



Figura 3. Operarios eliminando ejemplares de plumero de la Pampa

Fuente: [3]

Todos estos problemas relativos al plumero de la Pampa acarrearán a su vez consecuencias económicas a diferentes niveles, como la pérdida del valor agrario de las tierras que invada, el colapso de redes de riego cercanas debido a la alta demanda de agua de los ejemplares, o elevados costes de los programas de restauración y conservación de los ecosistemas afectados, tanto a nivel de flora como de fauna. Por otra parte, el empeoramiento de los paisajes afectados por la cortaderia selloana y los problemas que afecten a la biodiversidad del lugar, pueden hacer que dicho lugar pierda interés turístico, por lo cual los negocios cercanos a esos paisajes pueden sufrir pérdidas monetarias.

Debido a todas las problemáticas expuestas previamente, y en especial a la falta de concienciación de la ciudadanía, es recomendable desarrollar un sistema de fácil uso capaz de detectar la cortaderia selloana a partir de imágenes aéreas, mediante el empleo de redes neuronales capaces de reconocer ambientes rurales o urbanos invadidos por esta planta, independientemente de su entorno y de otros tipos de vegetación circundantes. Otra opción que se analizará será la implementación de una máquina de vectores de soporte.

En la problemática que atañe a este proyecto, las redes neuronales deben ser capaces de, partiendo del análisis de las características de unas imágenes catalogadas como ejemplares con plumero o sin plumero, determinar si otra serie aleatoria de imágenes poseen también ejemplares de cortaderia selloana o no. Por otra parte, la máquina de vectores de soporte (SVM, según sus siglas en inglés), también servirá para poder evaluar la clasificación de imágenes del dataset.

La temática de este TFG se centra en la utilización de la IA. En la actualidad, se considera un campo del conocimiento en pleno auge y desarrollo en distintos ámbitos de la sociedad, independientemente de su naturaleza. A su vez, este nuevo campo del conocimiento es causa de nuevos horizontes laborales, académicos o de investigación, como puede ser el análisis del territorio mediante imágenes satelitales para detectar la presencia de plantas invasoras. Los avances derivados de la IA, aunque pueden ser considerados erróneamente como intangibles, van en aumento y no han de ser ignorados por la población, sino que se debe buscar la participación de la gente para perfeccionar su tecnología. La inteligencia artificial continúa su desarrollo y tiene diferentes inconvenientes de distinto carácter, tales como la toma autónoma de decisiones y la practicidad, para entre otros aspectos, ser capaz de ayudar en cualquier tipo de tarea cotidiana, con el objetivo de solucionar problemas comunes. Referido a la temática del TFG, también se puede conseguir la mejora de la calidad de vida de las personas mediante la identificación de plantaciones de cortaderia selloana y de esta forma facilitar su eliminación, lo cual podría prevenir o incluso solucionar posibles problemas de salud, ecológicos y económicos.

3. OBTENCIÓN DE DATOS

3.1. INTRODUCCIÓN

Los programas que se realizan, tanto las redes neuronales como la máquina de vectores de soporte, requieren de un proceso de obtención de datos. Por medio de este procedimiento, cualquier usuario puede disponer de imágenes, independientemente del reconocimiento a realizar.

Para el caso que se trata, y puesto que la existencia de plumero de la Pampa se considera como una emergencia a diferentes niveles, como en materia sanitaria o en materia ecológica, existen diferentes sitios web donde se puede recopilar información de parcelas urbanas y rurales que poseen cortaderia selloana. Estas fotografías se descargan con la mejor calidad posible para poder ser analizadas en los procesos de entrenamiento que desarrollen los programas. A pesar de que la finalidad del proyecto es el reconocimiento de la especie invasora mencionada con anterioridad, se requiere de otro tipo de imágenes en las cuales dicha planta no esté presente.

Es importante destacar que las imágenes son ortofotografías, es decir, representaciones de la superficie terrestre en las cuales todos los elementos que aparecen se encuentran igualmente escalados, por lo que se desprende que, a priori, los cultivos de plumero de la

Pampa no se encuentran resaltados en comparación con otros elementos de las imágenes, como otros cultivos o plantaciones.

3.2. STOPCORTADERIA.ORG

En este sitio web, que puede ser utilizado a nivel estatal, cualquier persona mediante registro online puede guardar localizaciones de plumero para alertar al resto de la población. De hecho, consta también de las localizaciones donde se ha eliminado plumero, con el fin de revisar cada cierto tiempo estos lugares y evitar que vuelva a brotar.

La página web está enmarcada en un proyecto financiado por la comisión europea, que tiene como objetivo localizar y cuantificar la expansión del plumero en parte del Arco Atlántico, especialmente en Cantabria, el suroeste de Francia o Portugal, entre otros lugares. Stopcortaderia.org también busca favorecer la concienciación de la ciudadanía, para que se sumen al proyecto y así reducir el daño en la biodiversidad más fácilmente. Otros países europeos también colaboran en menor medida con la iniciativa, ya que, a fecha de hoy, sus litorales no han sido invadidos de una manera tan grave.

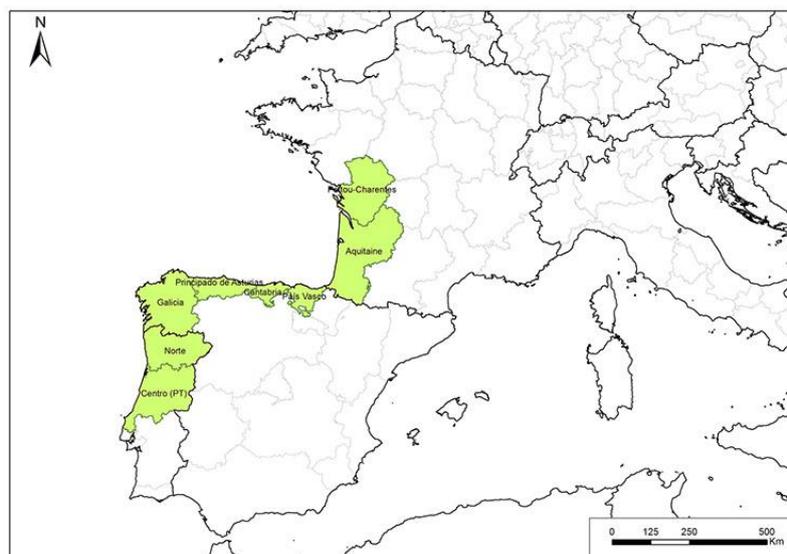


Figura 4. Representación del Arco Atlántico invadido por la cortaderia selloana

Fuente: [4]

Mediante el empleo de este sitio web, se pueden introducir nuevos ejemplares de plumero de la Pampa. Para ello, se hace click en “INTRODUCE NUEVA CITA” y se inicia sesión en EEIKO, que es una multiplataforma dedicada al control de especies exóticas invasoras. Luego, la web carga un mapa donde se debe seleccionar la zona donde se encuentre el

nuevo plumero de la Pampa, incluso ayudándose de la vista satelital para mayor comodidad. Se hace click en “INTRODUCIR CITA” y se deben rellenar los siguientes campos:

- Ejemplares dispersos
- Población abundante
- Fecha del nuevo registro
- Lugar o hábitat (duna, ribera, marisma, matorral, etc)
- Informar (sí o no)
- Notas

Obligatoriamente es necesario adjuntar una imagen del ejemplar para guardar la cita, lo que ayuda a verificar que la identificación de nuevos ejemplares es correcta. Finalmente, se pulsa en “Aceptar”. Esta cita será validada posteriormente, antes de aparecer en el mapa, para así evitar falsas localizaciones.

Usando todas las citas introducidas por los usuarios, se completa el mapa que está disponible en la página, que es actualizado cada cierto tiempo en función de la eliminación de ejemplares y de la aparición de nuevas plantas de cortadería selloana.

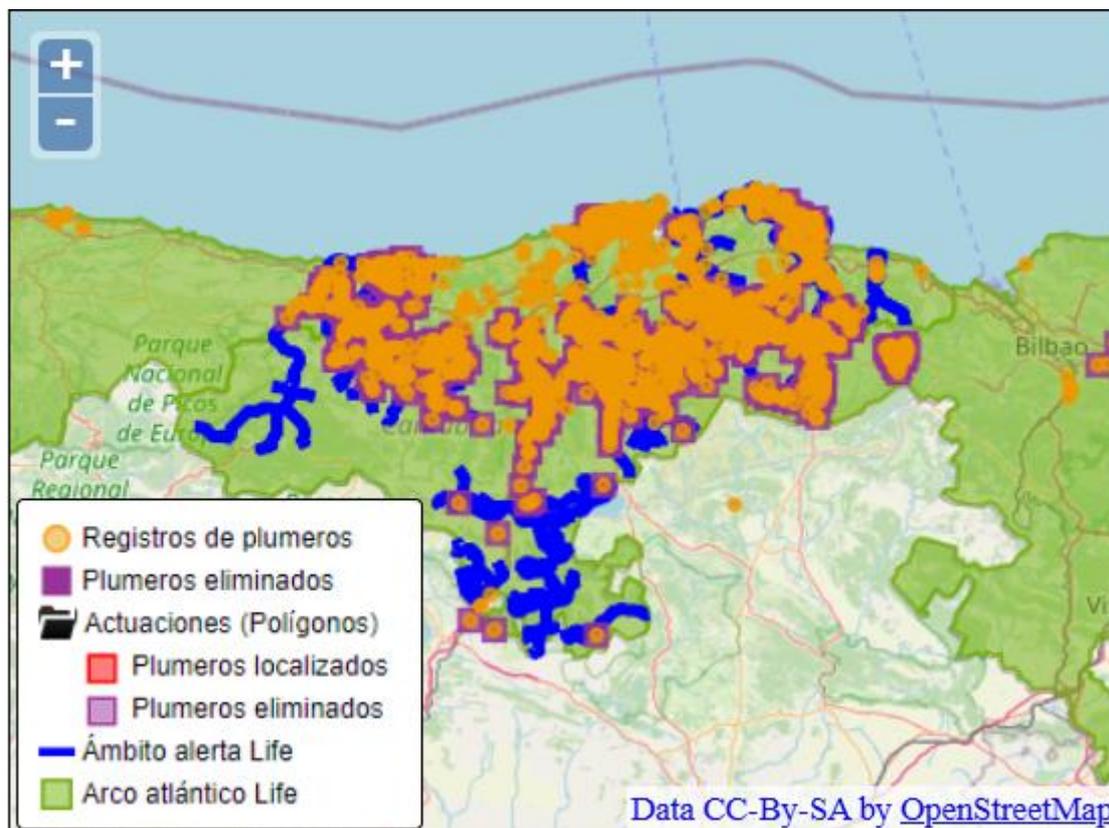


Figura 5. Imagen del mapa de stopcortaderia.org

Fuente: [5]

3.3. SIGPAC

Según sus siglas, el Sistema de Información Geográfica de Parcelas Agrícolas, desarrollado por el Ministerio de Agricultura, Pesca y Alimentación en el año 2000, permite la identificación geográfica de parcelas con su correspondiente cultivo dentro de una gran variedad, desde cultivos de cereal hasta algodón, previamente declaradas por agricultores y ganaderos. No obstante, SigPac no estuvo disponible para los usuarios en línea hasta el año 2005. A pesar de haber sido diseñado como una herramienta gráfica para ayudar a los agricultores a nivel nacional informando sobre el tipo de cultivo y la superficie cultivada, también es útil en otros campos como el urbanismo o las infraestructuras, gracias a la resolución de sus ortofotos digitales, tomadas desde avionetas o satélites, lo cual provocó que se añadiese cada vez más información sobre las diferentes parcelas.

Gracias a SigPac, los agricultores ven facilitado el acceso a las ayudas estatales gracias a la declaración de los cultivos por superficie. Esto agiliza los trámites burocráticos con las diferentes administraciones, especialmente en caso de dudas sobre el área de tierra trabajada. Gracias a esta información, los hortelanos pueden realizar cálculos esenciales para la viabilidad de sus cultivos, como la cantidad de agua y de abono que necesitan sus campos en tiempo real. Su mejor característica es la ayuda a la monitorización de los distintos cultivos, ya que las imágenes satelitales se van renovando cada cierto tiempo, permitiendo apreciar la evolución de las cosechas de una manera mucho más cómoda. Igualmente, SigPac permite realizar mediciones de distancias en las parcelas, con la finalidad de que los agricultores puedan conocer sus terrenos de manera óptima, en caso por ejemplo de que deseen realizar cambios tanto en su superficie cultivada como en su superficie cultivable.

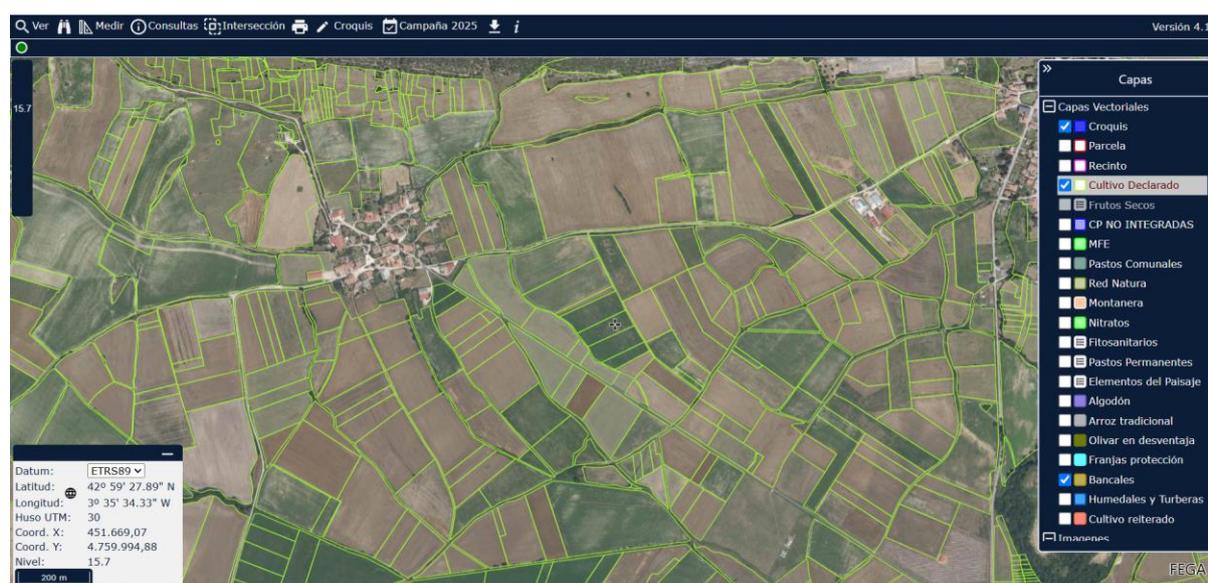


Figura 6. Interfaz gráfica de la web de SigPac

Fuente: [6]

Una vez se declara un nuevo cultivo, su información pasa a ser de dominio público y cualquier usuario puede verla. Para ello, clicando en “Consultas” y “Cultivo Declarado” o simplemente pulsando la letra “D”, se puede obtener la información detallada como el tipo de cultivo de una parcela determinada, la superficie cultivada o las ayudas económicas susceptibles de ser recibidas. Esta información puede obtenerse tanto en una vista preliminar como descargando la documentación en formato PDF.

GOBIERNO DE ESPAÑA
MINISTERIO DE AGRICULTURA, PESCA Y ALIMENTACIÓN

sigpac
FONDO ESPAÑOL DE GARANTÍA AGRARIA (FEGA)

sistema de identificación de parcelas agrícolas

Campaña declarativa 2024

Recinto SIGPAC en el que se ubica la línea de declaración						
Provincia	Municipio	Agregado	Zona	Polígono	Parcela	Recinto
49 - ZAMORA	198 - ROALES	0	0	2	375	1

C.A. Gestora	Nº Expediente	Sistema Explotación	Superficie (ha)	Ayuda Directa C.Principal	Ayuda Directa C.Secundario	Ayuda PDR	Cultivo Principal	Cultivo Secundario	Aprovechamiento	Imprimir
8 - CASTILLA Y LEÓN	49031779	Secano	6,0400	18,19			5 - CEBADA			

Tipo de Ayuda Directa	
18 - Ayuda Básica a la Renta para la Sostenibilidad (ABR)	
19 - Ayuda Complementaria Redistributiva	

Figura 7. Información ofrecida en SigPac sobre un cultivo declarado

Fuente: [7]

En caso de realizar el trabajo actual utilizando SigPac, esta información no es necesaria. Únicamente se requeriría de la descarga de las imágenes para completar los conjuntos de datos.



Figura 8. Fotografía aérea de una parcela con plumero obtenida mediante SigPac

Fuente: [8]

Más allá de las imágenes de cortaderia selloana, se requiere de otro tipo de imágenes en las que no exista la especie invasora. Esto se debe a que tanto las redes neuronales como la máquina de vectores de soporte, para distinguir imágenes de plumero de la Pampa, requieren de imágenes donde no aparezca este tipo de plantas. Para estas nuevas imágenes, también se emplea la web de SigPac, y se elige todo tipo de instantáneas, como capturas de bosques, ya que los árboles pueden tener tonalidades similares al plumero si son vistos desde un plano cenital. Otras imágenes aéreas empleadas para entrenar la red neuronal y la máquina de vectores de soporte son simplemente campos, plantaciones urbanas, campos de cosecha de diferentes tipo o pastos herbáceos.

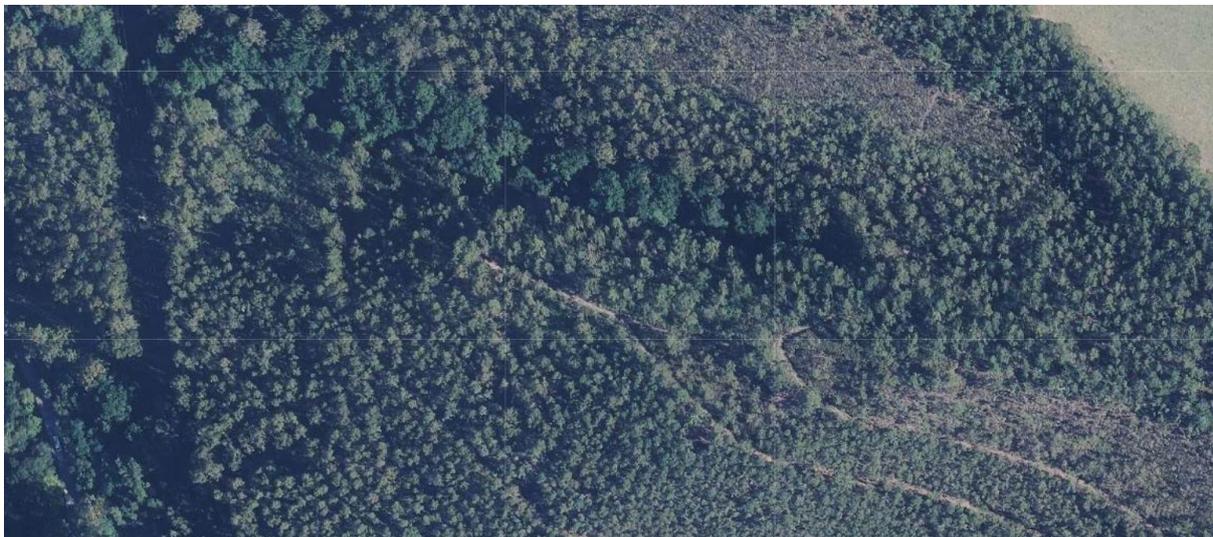


Figura 9. Fotografía aérea de una parcela sin plumero mediante SigPac

Fuente: [9]

Aparte de lo expuesto, SigPac posee la posibilidad de aumentar las imágenes antes de descargarlas, sacándolas desde una menor altura, mediante una barra situada en la parte superior izquierda de la pantalla. Dicha barra regula el enfoque desde un valor mínimo de 5 hasta un valor máximo de 20. Cambiando este valor, se pueden descargar imágenes de una misma zona del territorio con distintos enfoques. Gracias a esto, se obtienen imágenes con diferentes matices, siendo el cultivo de cortaderia selloana más visible y nítido en unos casos que en otros.

3.4. SENTINEL HUB

Otra opción para procesar y trabajar con imágenes satelitales es Sentinel Hub. Conforman un sistema de información geográfica (también llamado SIG, según sus siglas en español), lo que permite la relación de diferente información con una localización geográfica concreta. Dichas instantáneas proceden de la misión de observación terrestre Sentinel-2 que, a su vez, se ve cimentada en el uso de dos satélites diferentes principalmente, el Sentinel-2A y el Sentinel-2B. Esta misión fue diseñada por la Agencia Espacial Europea (ESA). No obstante, emplea una amplia gama de satélites en el presente, con el fin de cubrir toda la superficie terrestre de manera actualizada en tiempo real.

Cada uno de los satélites tiene una función específica, lo que hace que se recopilen imágenes con diferentes características, como fotografías destinadas a servicios de vigilancia atmosférica y oceánica durante las 24 horas, instantáneas para la monitorización de la calidad del aire y la medición de contaminantes de la atmósfera o la simple obtención de imágenes de radar.



Figura 10. Imagen del satélite Sentinel-2

Fuente: [\[10\]](#)

Para poder desarrollar esta función, los satélites están constantemente en puntos opuestos del globo terráqueo, es decir, poseen un desfase de 180° entre sí. Cada uno de ellos envía imágenes cada 10 días, pero gracias a una optimización de su funcionamiento, un satélite envía imágenes 5 días después del otro de manera sistemática. Las imágenes que se toman barren un máximo de hasta 290 Km de ancho. Gracias al avance de la web, Sentinel ofrece tanto sus imágenes originales procedentes directamente del satélite como unas imágenes mejoradas, capaces de eliminar problemas derivados de las condiciones atmosféricas, como la baja nitidez producida por nubosidades. Todas estas imágenes obtenidas son de dominio público para los usuarios.

También permite la medición de distancias y la búsqueda de ciertas características específicas de las imágenes. Para facilitararlo, cuenta con la presencia de capas que destaquen unos elementos u otros, como carreteras o fronteras terrestres y marítimas. Posee otras opciones tales como la detección de presencia de vegetación o el cálculo del NDVI en un área concreta, gracias al cual se puede medir la salud de la vegetación dependiendo de la reflexión de la luz sobre las plantas en diferentes frecuencias, lo que ofrece información sobre la densidad de las plantas existentes en una imagen satelital.

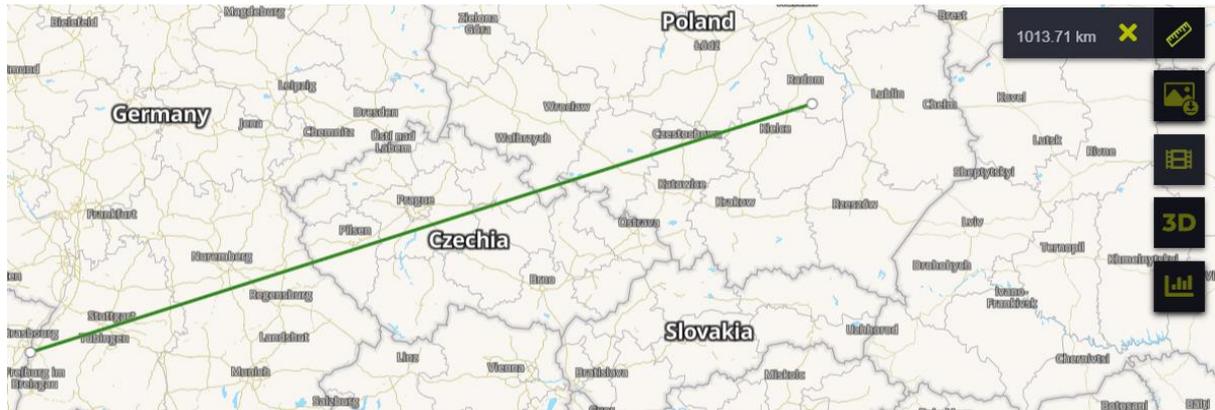


Figura 11. Ejemplo de medición de distancias en Sentinel Hub

Fuente: [11]

Algunas de las novedades que posee Sentinel Hub en comparación con otras opciones es la posibilidad de cambiar la visibilidad de las áreas usando la opción de 3D y la capacidad de utilizar histogramas para destacar diferentes colores en el terreno. Esto puede ayudar a la hora del reconocimiento de diferentes cultivos y cosechas que cuenten con tonalidades específicas.

Para una mejor utilización, tiene un menú que permite el acceso a diferentes opciones para visualizar. Algunas de las más destacadas están enfocadas a la geolocalización de incendios, para saber su posible extensión, así como inundaciones y sequías para comprobar la zona afectada, la contaminación atmosférica de una región o la detección de nieve en diferentes cotas. Estas opciones de uso hacen que Sentinel Hub sea de gran utilidad en diferentes ámbitos, como la física, la lucha contra la contaminación o la seguridad ciudadana. Además de estas consultas, se pueden guardar las localizaciones que se deseen, empleando cualquier satélite específico, del día que se quiera, con la finalidad de crear un “timelapse” y ver su evolución temporal del terreno, así como también realizar comparaciones mediante superposición de fotografías.

Dependiendo de las imágenes que se deseen adquirir, puede ser más útil un satélite u otro. Para saber cuál de ellos se ajusta más a nuestras necesidades, es posible clicar en los interrogantes al lado del nombre de cada uno. Por ejemplo, si se usa el satélite Sentinel-2 se puede saber cuándo han sido tomadas sus imágenes, su resolución y los usos más comunes que poseen dichas instantáneas, como en este caso la monitorización de la vegetación y el control de áreas quemadas. Para más ayuda, se puede ajustar el intervalo temporal de las imágenes, de forma que el usuario pueda encontrar imágenes con las características que requiera.

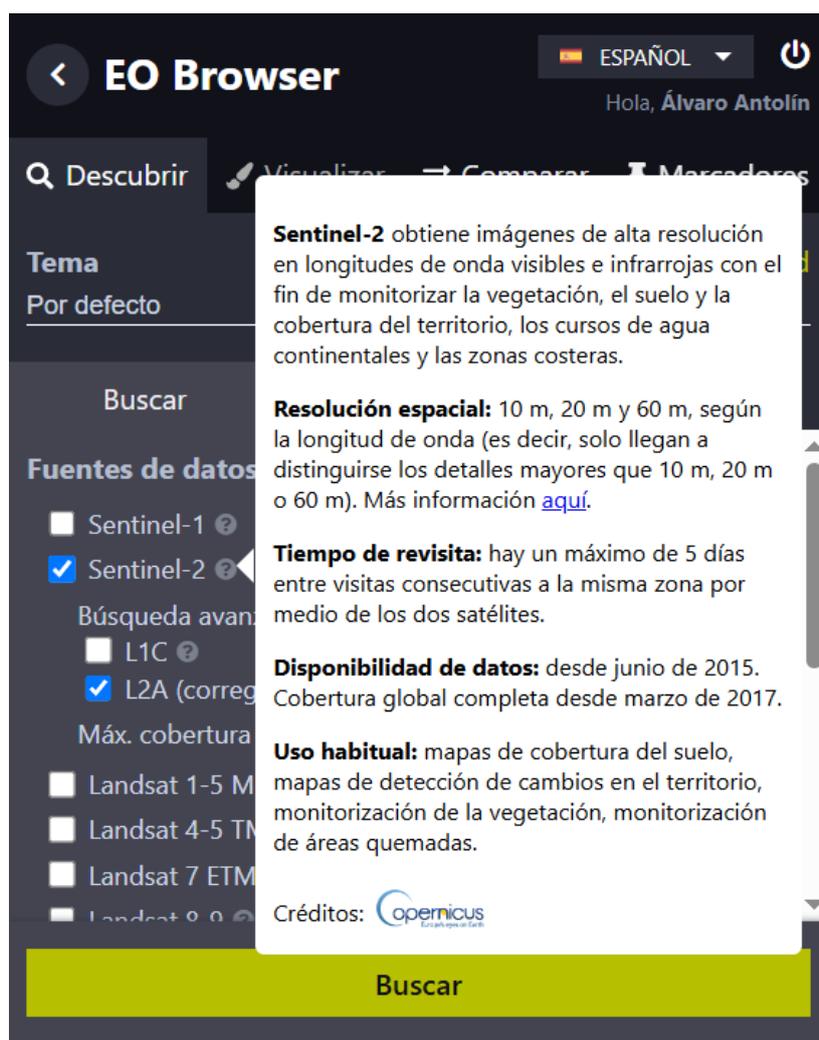


Figura 12. Información disponible en la web sobre el satélite Sentinel-2

Fuente: [12]

Con el fin de facilitar el acceso a los datos, Sentinel Hub permite especificar la zona deseada de diferentes formas. Las formas más comunes de búsqueda son por la introducción de la ubicación por nombre o por coordenadas GPS. Después de especificar un lugar determinado, en la barra izquierda de "Descubrir" se muestran las imágenes disponibles, junto con el

satélite que tomó la imagen, la fecha, la hora en la que se realizó y el porcentaje de nubosidad de dicha instantánea. A la hora de visualizarlas, se pueden almacenar mediante el uso del icono de la chincheta o marcador. Repitiendo este proceso con varias imágenes, se genera un dataset. Después, en “Comparar”, se pulsa en “Añadir todos los marcadores”, de forma que se cargan las imágenes guardadas.



Figura 13. Reconstrucción del curso del río Nilo usando imágenes de diferentes años

Fuente: [13]

Gracias a esta herramienta, se pueden comparar imágenes de diferentes días de manera simultánea, con el fin de monitorizar la evolución de cultivos, ríos, lagos e icebergs. Para la temática de este TFG, se puede usar para comprobar la evolución temporal la evolución de diferentes terrenos invadidos por la cortadería selloana de una manera fiel.

3.5. PNOA

Según sus siglas, el Plan Nacional de Ortografía Aérea es un proyecto conjuntamente desarrollado entre la administración general del Estado y las CCAA. Iniciado en 2004, proporciona imágenes aéreas y modelos digitales de elevaciones en todo el territorio nacional.

En 2009, se introdujo la tecnología LiDAR, usada para la obtención de modelos digitales de alta precisión, especialmente enfocada en la adquisición de imágenes en lugares donde sería más complicado obtenerlas empleando otras tecnologías, como en áreas inundadas, en bosques y en el control de carreteras, lo que hace de PNOA una opción adecuada a la hora de extraer imágenes complejas. Esta tecnología es el acrónimo de Light Detection and Ranging. Siguiendo esta técnica se emiten pulsos de láser para medir distancias y

movimientos, gracias al fenómeno físico de reflexión. Sabiendo la velocidad a la que se envía el rayo y pudiendo medir el tiempo que tarda en volver hasta el punto desde el que partió, se sabe con exactitud la distancia entre el emisor y los diferentes receptores, porque puede captar elementos a diferentes niveles, como edificios o árboles, siendo capaz de atravesarlos dependiendo de la intensidad y de la frecuencia del pulso, o el propio suelo.

Cada uno de los puntos resultantes del uso de LiDAR se clasifica a través de códigos numéricos de enteros en el archivo con extensión “.LAS”, formato que permite el intercambio de archivos con información sobre nubes de puntos.

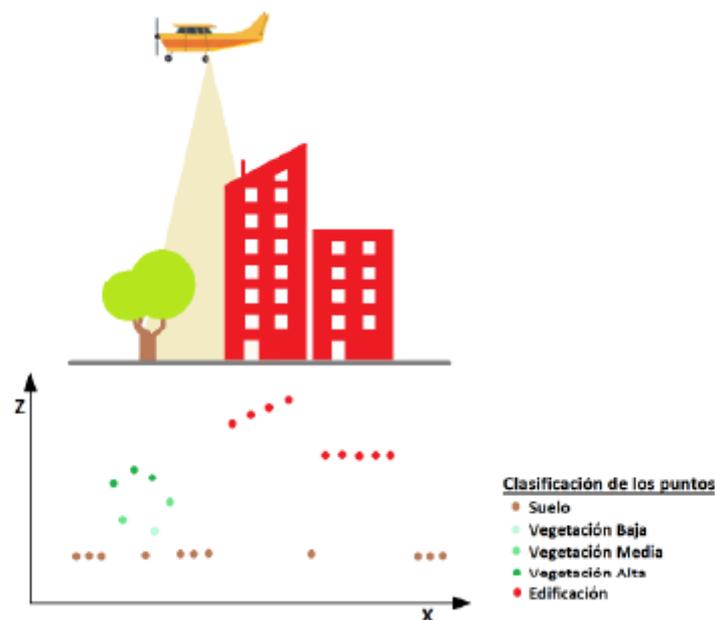


Figura 14. Ejemplo de clasificación de puntos mediante la tecnología LiDAR

Fuente: [14]

Para el tratamiento de la información, se reciben en primer lugar los datos en bruto. Después, se realiza un control de calidad y se cambia el tipo de coordenadas elipsoides a ortométricas. Posteriormente, se asocian colores a las imágenes para diferenciar los elementos dentro de las mismas, usando el espacio RGB o el espectro infrarrojo. Finalmente, se realiza una clasificación de la información y se distribuyen los resultados.

La tecnología LiDAR posee diferentes finalidades. Principalmente, está presente en el control de la población forestal, gracias a su facultad de penetración dentro de los bosques. Esto permite revisar la evolución de dichos ecosistemas tanto en condiciones normales como en caso de incendios, para saber cómo se recupera el territorio posteriormente. Además, también es capaz de monitorizar otros entornos como dunas o glaciares, que son ambientes

donde los cambios pueden ser inapreciables si se usan otras tecnologías. Otra de las funcionalidades que posee se encuentra en el urbanismo, desde el estudio del crecimiento de las ciudades hasta el análisis de las carreteras, desde su construcción hasta su implementación. Estos avances a su vez provocan innovaciones tecnológicas en el campo de la telecomunicación, al poder conocer con exactitud el urbanismo de una zona determinada, lo que favorece la planificación y la instauración de redes de transmisión de datos, tanto en zonas urbanas como rurales. Su capacidad de atravesar la vegetación también tiene una gran utilidad en la arqueología. Gracias a esta habilidad, el empleo de PNOA colabora en hallazgos arqueológicos de diferente tipo, como asentamientos antiguos, fortificaciones o murallas de diferentes períodos históricos.

3.6. MAPAS.CANTABRIA.ES

Es una herramienta destinada al estudio y al análisis únicamente de la geografía cántabra. Se caracteriza por albergar en una sola página las múltiples funcionalidades explicadas en los servicios anteriormente explicados, como la posibilidad de consultar parcelas de cultivo por medio del catastro, posibilidad de mediciones de distancias y áreas de territorio, así como la obtención de la latitud, en otras palabras, la distancia angular entre un punto cualquiera y la línea del Ecuador, y la longitud, que es la separación existente entre un punto y el meridiano de Greenwich, de un determinado punto del mapa.

Su utilidad va más allá de los elementos que ofrece, ya que está disponible para su uso en diferentes equipos además de ordenadores y móviles, como iPad mediante un sistema operativo iOS o tabletas empleando un SO de Android. Además, cuenta con información mucho más detallada con respecto a los años, ya que sus registros históricos llegan hasta 1956, aunque de una manera mucho menos detallada que los registros más recientes.

La gran ventaja que posee con respecto al resto, excepto stopcortaderia, es que también cuenta con parcelas de terreno que tienen cortaderia selloana. Para poder buscar las localizaciones de este tipo de planta en el mapa se debe activar su capa, que se encuentra en el apartado “Especies invasoras”, dentro de “Conservación de la Naturaleza”.

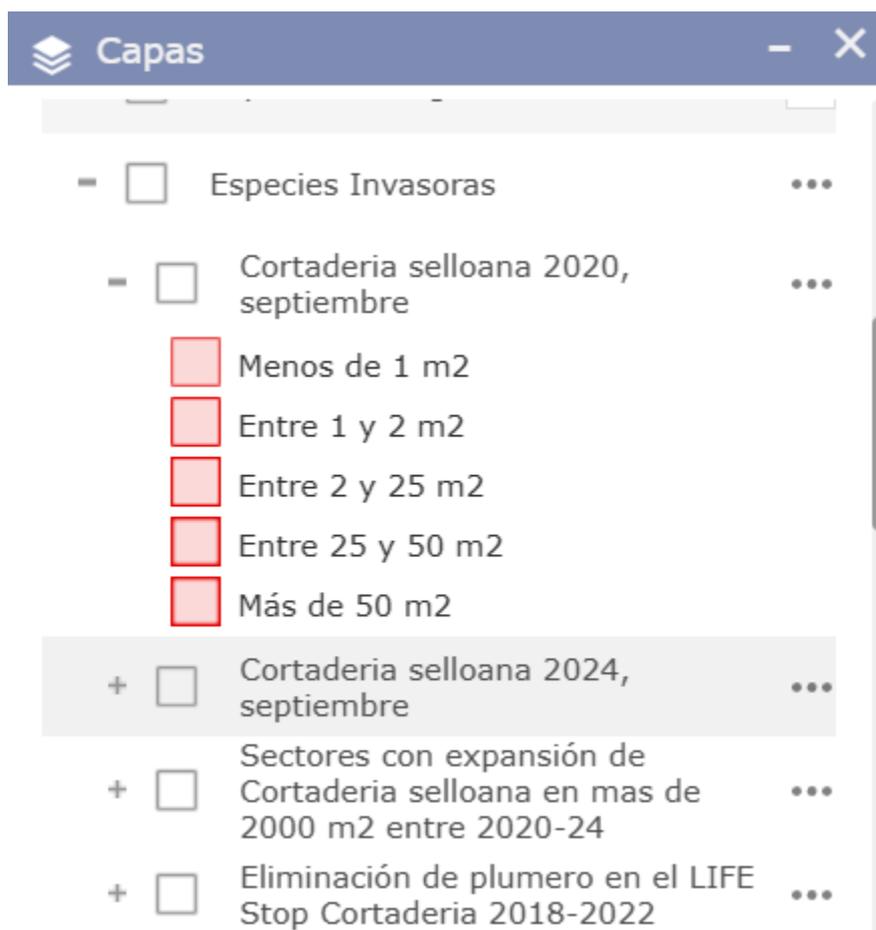


Figura 15. Imagen de las subcategorías dentro de “Especies Invasoras”

Fuente: [15]

En esta subclase, existen diferentes categorías, dependiendo de la expansión de las parcelas y del momento en el que se adquirieron los datos. La división en función de la superficie invadida es de gran utilidad para saber cuáles son los focos más importantes. Aparte, también es de gran ayuda la opción que permite resaltar los plumeros eliminados, al igual que existía en cierta medida en Stopcortaderia porque, como se ha explicado con anterioridad, pueden existir rebrotes.

Dependiendo de la elección que se realice, en el mapa se resaltan unas zonas u otras. Además de poder elegir el área del plumero, se puede escoger un espacio temporal más concreto para comprobar tanto la evolución de la cortaderia selloana detectada como las nuevas detecciones.

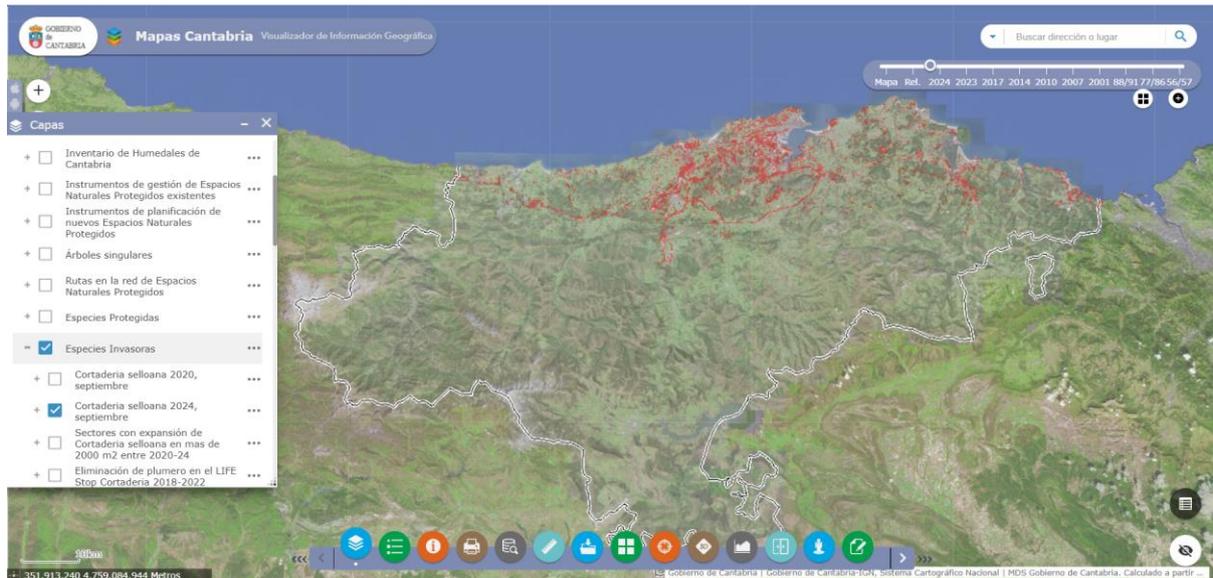


Figura 16. Ejemplares de cortadería selloana detectados en septiembre de 2024

Fuente: [16]

Haciendo zoom en zonas concretas, se pueden ver con un detalle mucho mayor las zonas concretas detectadas. Esta herramienta puede ser de gran ayuda, ya que aparte de poder tener la etiqueta teórica en las imágenes del dataset y poder comprobar, constituye otra opción de verificación a tiempo real de las detecciones realizadas por los programas, señalando la cortadería selloana en imágenes satelitales a diferentes alturas.

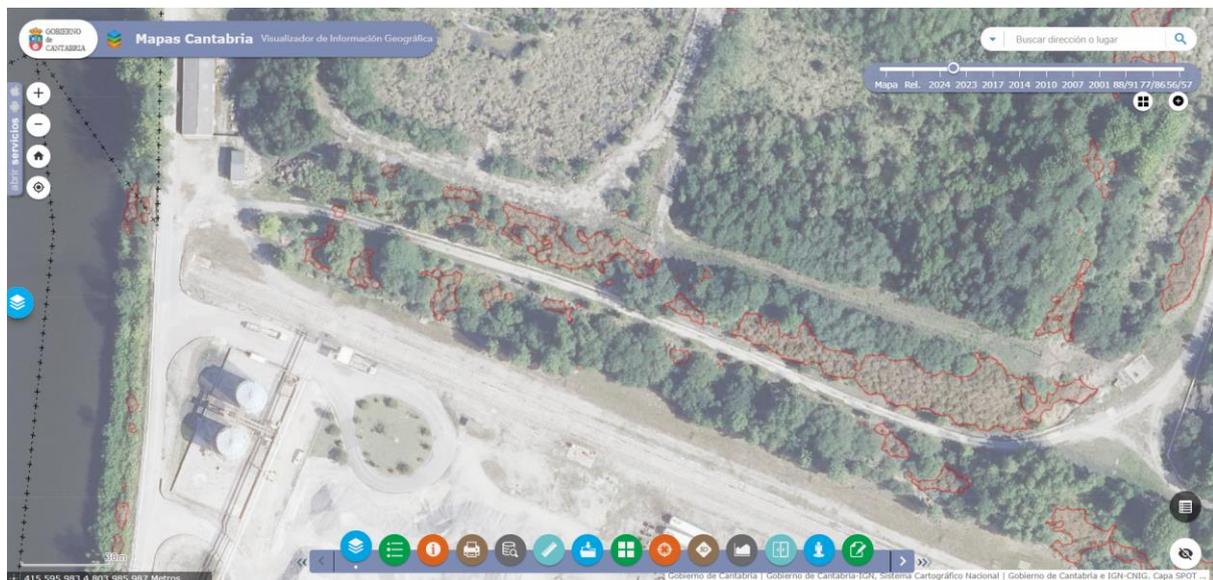


Figura 17. Captura de los ejemplares de plumero de la Pampa detectados con mayor detalle

Fuente: [17]

Una vez realizada la configuración, aparecen en la parte inferior los resultados de las detecciones. Se especifica la longitud del tramo que posee cortadería selloana, la exactitud de la medida, la fecha de la referencia, así como un enlace al centro de descargas donde se puede obtener mediante un archivo PDF la información completa de la detección. En definitiva, es una herramienta útil porque recoge diferentes funcionalidades presentes en otras páginas webs, además de ofrecer también información sobre la cortadería selloana, como ocurría en stopcortaderia.org, pero con una mejor resolución.

3.7. GENERACIÓN DEL DATASET Y CONCLUSIONES

Habiendo analizado todas las posibles opciones relativas a la extracción de datos, se determina que la mejor forma de extraer las imágenes que componen el dataset es consultar en primer lugar las localizaciones del mapa disponible en stopcortaderia.org y después extraer las imágenes de esos mismos lugares en SigPac. Esto se debe a la precisión y a la fiabilidad que posee en primer término la web de stopcortaderia.org en comparación con mapas.cantabria.es, así como a la alta resolución de las imágenes procedentes de SigPac si se compara con otras webs como PNOA.

Para ello, con ambas pestañas abiertas en el navegador, se consulta en primer lugar las localizaciones de cortadería selloana en el mapa de stopcortaderia.org. Teniendo cautela con el control del zoom, se puede determinar con relativa exactitud la posición del cultivo. Posteriormente, sabiendo esa geolocalización, se entra en la pestaña del mapa de SigPac y se busca. Una vez se pueden comparar ambas fuentes, si coincide que la localización de plumero de la Pampa de stopcortaderia.org muestra un cultivo visible de manera clara en la otra web, se descarga la imagen correspondiente usando diferentes valores de zoom para que los programas sean más completos en cuanto a la detección.



Figura 18. Localización de un ejemplar de plumero de la Pampa en stopcortaderia.org

Como cualquier localización de stopcortaderia.org, corresponde a una ubicación en SigPac.



Figura 19. Captura del cultivo en SigPac anteriormente encontrado en stopcortaderia.org

Fuente: [19]

Este procedimiento se repite para todas las imágenes del conjunto de datos, excepto para las imágenes en las que no aparece cortadería selloana. En tal caso, el proceso de adquisición es más rápido ya que no es necesario comprobar información adicional en otra web más allá de SigPac. No obstante, el único problema es garantizar que las fotografías que compongan el dataset de imágenes sin plumero de la Pampa tenga la calidad necesaria. Una vez se

tengan estas instantáneas, se clasificarán en determinados directorios en función de la presencia o de la ausencia de la especie invasora.

Por otra parte, en el desarrollo de las herramientas software que se explicarán en este proyecto, será necesaria la carga de imágenes mediante PNOA, dado que sus imágenes también son de buena calidad y pueden descargarse a tiempo real para trabajar con ellas.

4. REDES NEURONALES BIOLÓGICAS

En biología, una neurona es cada una de las células nerviosas que forman tanto el cerebro como el sistema nervioso central. Es una de las células más comunes en el cuerpo humano, siendo capaz de ramificarse en diferentes patrones y estructuras. Se encarga de transmitir información desde cualquier parte del cuerpo hasta el cerebro, que elaborará la respuesta. Para la recepción y la transmisión de la información en la propia neurona, se utilizan neurotransmisores, que transforman la información en una señal electroquímica que se propagará.

La estructura de la neurona posibilita la transmisión de datos. En primer término está el soma, que contiene el núcleo y la información genética, que son elementos necesarios para la supervivencia celular. Luego están las dendritas, que son proyecciones de las neuronas capaces de recibir los estímulos procedentes de otras células nerviosas. Por último, está el axón, que nace del soma y se encarga de la transmisión de impulsos eléctricos a otras neuronas u otros músculos para permitir la comunicación del sistema nervioso. La unión de unas neuronas con otras da lugar a una red neuronal biológica, formando a su vez conexiones más complejas hasta abarcar todo el cuerpo del ser vivo y llegando hasta el cerebro, que recibe y procesa la información recogida, para actuar en consecuencia.

El estudio de las neuronas recibe el nombre de “neurociencia”. Esta ciencia, enfocada tanto a los seres humanos como a los animales, analiza y estudia los sistemas nerviosos para saber cómo se crean las conductas. La neurociencia y el estudio del cerebro, al igual que otras disciplinas dentro de la medicina, no deja de ser un campo multidisciplinar, ya que los interrogantes que deja el conocimiento pueden estudiarse desde un enfoque genético, filosófico, biológico o psicológico. Además, dentro de estos campos existen a su vez divisiones como el estudio de la red neuronal a nivel celular, es decir, simplemente las neuronas o a nivel global, así como sus conexiones.

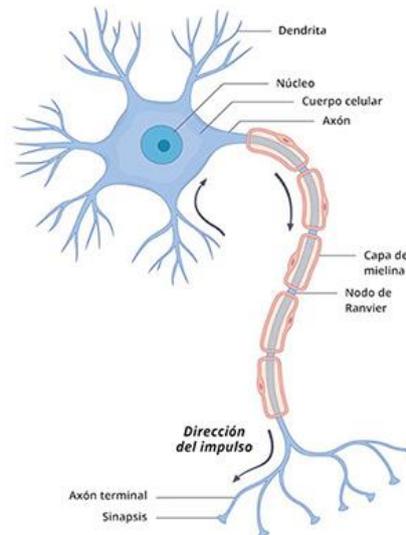


Figura 20. Representación de una neurona con sus diferentes partes señaladas

Fuente: [20]

Hoy en día, la neurociencia continúa con una serie de aspectos sin resolver de diversa índole, como la base neuronal de las enfermedades mentales, la plasticidad neuronal a lo largo del tiempo, la integración sensorial tanto en seres humanos como en animales o los mecanismos mentales que intervienen en los sueños, lo cual ayuda a que sea uno de los campos de estudio más importantes dentro de la medicina actual.

5. REDES NEURONALES ARTIFICIALES

5.1. INTRODUCCIÓN

Se puede considerar que el estado del arte referido a las redes neuronales es avanzado y tiene visos de que continúe desarrollándose aún más en el corto y medio plazo. Tanto es así que, en el año 2024, el premio Nobel de física fue ganado por John J. Hopfield y Geoffrey E. Hinton, quienes se considera que han estipulado las bases de la inteligencia artificial. Sus innovaciones han permitido el uso de Deep Learning en la actualidad para aplicaciones como el reconocimiento de imágenes o el reconocimiento de voz.

En Matlab, una red neuronal es un modelo que busca asemejarse a la estructura que componen las neuronas del cerebro humano, con capas de nodos interconectadas. En este caso, se dice que la red neuronal es artificial. Esta red es capaz de aprender información, clasificarla y pronosticar eventos futuros en base a la información procesada. Dentro de las redes neuronales, existen las redes neuronales convolucionales, que componen arquitecturas especializadas para Deep Learning. Este tipo de redes neuronales está especializado en el

reconocimiento de objetos a partir de fotografías, así como la clasificación y la segmentación de imágenes.

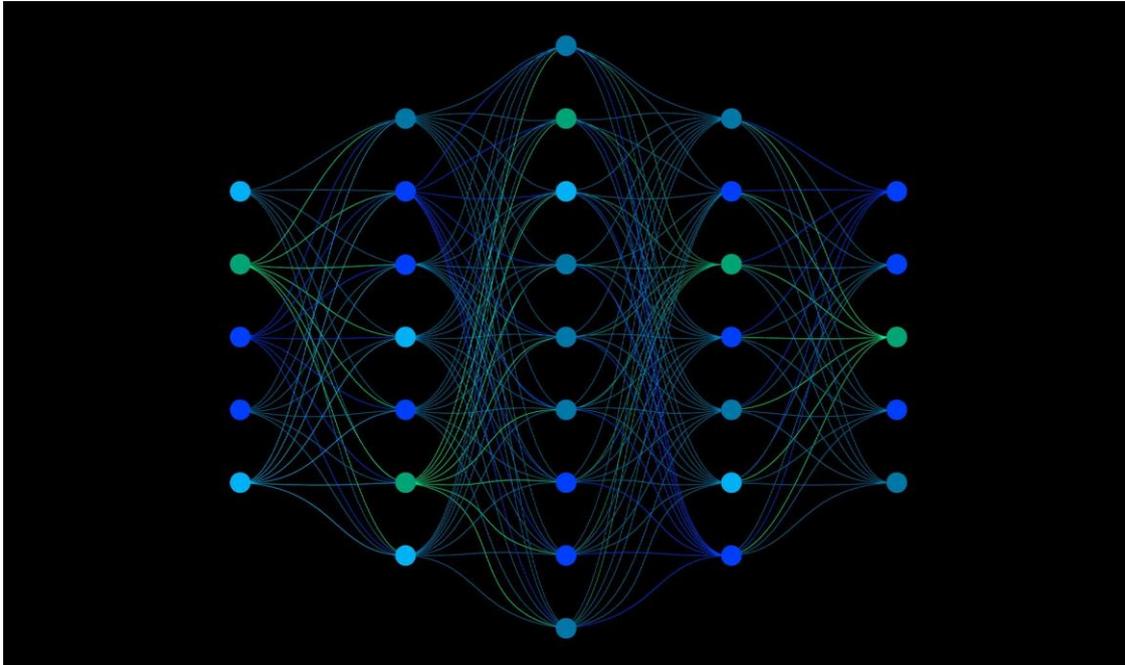


Figura 21. Representación de las distintas capas de una red neuronal

Fuente: [21]

En comparación con otros sistemas de clasificación, una red neuronal es más tolerante a fallos, al disponer de información redundante en varios puntos. Además, son modelos flexibles, dado que la señal de salida se mantiene, aunque existan cambios en la entrada. No obstante, requiere mucho más tiempo de entrenamiento para el desempeño de una tarea concreta si no está preentrenada o si se tiene una gran cantidad de datos al inicio.

Por consiguiente, es reseñable la recomendación del uso de redes neuronales preentrenadas en lugar de confeccionar redes neuronales desde 0 o “from scratch”, según su terminología en inglés. Es muy difícil que una red neuronal sin un entrenamiento previo sea capaz de generalizar ni se pueda adaptar a nuevas tareas. Por otra parte, las redes neuronales preentrenadas presentan arquitecturas que han demostrado su correcto funcionamiento con anterioridad, por lo que sólo es necesario adaptarlo a una nueva tarea, como puede ser la clasificación de imágenes en función de la presencia o de la ausencia de cortadería selloana. Además, para una mayor facilidad para el usuario, dichas arquitecturas son diseñadas específicamente con objetivos determinados como la extracción de características para la clasificación de imágenes, análisis de textos para traducción automática, búsqueda de información inteligente o reconocimiento de voz. Todas estas ventajas hacen que el uso de redes preentrenadas sea mucho más recomendable que construir por completo una red

neuronal desde su capa de entrada hasta su capa de clasificación, pasando por las diferentes convoluciones y transformaciones pertinentes.

Es a su vez importante mencionar que la IA y las redes neuronales no son lo mismo. La inteligencia artificial es un campo amplio dedicado a crear sistemas que en condiciones normales requieren de una inteligencia humana, mientras que las redes neuronales se encuentran incluidas en la IA, dentro del espectro del aprendizaje automático. En este tipo de aprendizaje, los equipos son capaces de realizar sus tareas sin necesidad de ser programados. En concreto, las redes neuronales están habitualmente dedicadas a la detección de patrones en imágenes y a la realización de tareas dependiendo de la información recogida.

Matemáticamente, el cálculo de la salida de una red neuronal es sencillo, aunque consta de varias partes.

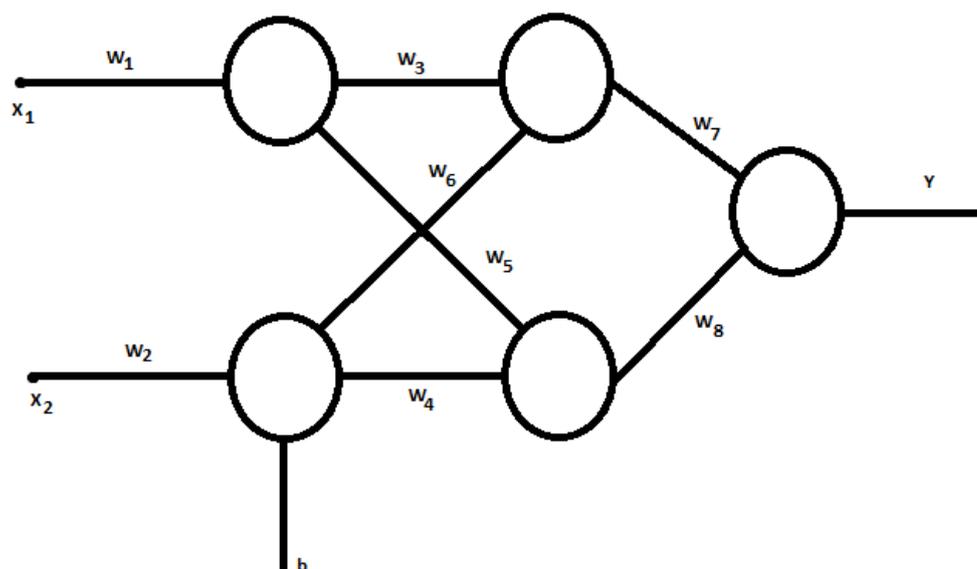


Figura 22. Representación teórica de una red neuronal artificial

Fuente: Elaboración propia

La salida Y de la red neuronal tiene la siguiente función:

$$Y = f(W \times x + b)$$

Siendo:

W : La matriz de los pesos.

x : El vector de entrada.

b: El vector de sesgos, utilizado para compensar el resultado. Su existencia dentro de la red neuronal no es obligatoria.

f: Función.

Y: La salida de la red neuronal.

Para su obtención, se deben calcular los valores de las salidas de cada una de las neuronas por separado, preferiblemente desde el final de la red hasta el comienzo, para que sea más fácil.

$$Y = X_5 \cdot W_7 + X_6 \cdot W_8$$

A partir de esta expresión, obtengo los valores de sus vectores de entrada X_5 y X_6 , usando vectores de entrada previos.

$$X_5 = X_3 \cdot W_3 + X_4 \cdot W_6$$

$$X_6 = X_4 \cdot W_4 + X_3 \cdot W_5$$

A su vez, X_3 y X_4 están formadas por sus vectores de entrada X_1 y X_2 , que son los vectores de entrada de la red neuronal, multiplicados por los pesos de cada una de sus neuronas. Además, en este caso, X_4 posee un sesgo sumado.

$$X_3 = X_1 \cdot W_1$$

$$X_4 = X_2 \cdot W_2 + b$$

Partiendo nuevamente de la ecuación de salida que se tenía originalmente, se puede sustituir los diferentes valores de X_i por X_1 y X_2 . De esta forma, se obtiene el valor de la salida en función de las entradas, de los pesos y de los sesgos.

$$Y = (X_1 \cdot W_1 \cdot W_3 + (X_2 \cdot W_2 + b) \cdot W_6) \cdot W_7 + ((X_2 \cdot W_2 + b) \cdot W_4 + X_1 \cdot W_1 \cdot W_5) \cdot W_8$$

Este valor de Y se pasará por la función de activación f y se obtendrá el valor correspondiente para cada par de entradas X_1 y X_2 .

El funcionamiento de la amplia mayoría de las redes neuronales artificiales está fundamentado tanto en la propagación hacia adelante y como en la propagación hacia atrás. Esto define el flujo de la información a lo largo de la estructura, y en última instancia sus posibles salidas y su comportamiento.

La propagación hacia adelante (en inglés "forward propagation") es el mecanismo matemático responsable de hacer que los datos de la red neuronal fluyan desde su introducción en la capa de entrada, siendo transformados y procesados mediante funciones de activación,

pesos, etcétera, hasta la capa de salida. De esta forma la red genera un “output”, es decir, una predicción. Este camino, que constituye el funcionamiento normal de cualquier red neuronal, es determinístico para la información, unidireccional y no depende de la fase en la que se encuentre la red en cuestión. Los datos comienzan en una capa de entrada, que otorga cada dato a una neurona de la red. Cuando estos datos llegan desde la capa de entrada hasta las capas ocultas, cada neurona determina la suma acumulada de entradas. Posteriormente, se usan funciones de activación, de forma que se añada no linealidad, lo que complica la tarea de la red neuronal y la hace más robusta frente a posibles perturbaciones. No obstante, antes de usar este tipo de funciones, se debe determinar la activación correspondiente, mediante la siguiente fórmula:

$$h_j = \sum_i \omega_{ji} \times X_i + b_j$$

Siendo:

1. h_j : Activación que usará la función de activación posterior.
2. ω_{ji} : Peso entre las neuronas i y j .
3. X_i : Entrada de la capa anterior.
4. b_j : Bias o sesgo. Es la tendencia del modelo a realizar predicciones simplificadas, pudiendo generar errores sistemáticamente.

Este proceso se repite constantemente hasta que se alcanza la capa final, donde se realiza la clasificación. Aquí, la red neuronal convierte los datos recibidos en una categoría, como un valor numérico o una etiqueta, que es el resultado final.

La propagación hacia atrás (también llamada “retropropagación” o en inglés “back propagation”) es un método de cálculo destinado al entrenamiento de redes neuronales artificiales basado en la regla de la cadena. Según esta regla, por cada paso hacia adelante a lo largo de la red neuronal, se da un paso hacia atrás con el fin de ajustar los parámetros del modelo. Mediante este método, se busca minimizar el error cuadrático medio y de esta forma, mejorar la precisión de las predicciones.

$$e.c.m. = \frac{1}{2} \sum_{k=1}^j (y_k - o_k)^2$$

Teniendo el valor del error cuadrático medio, siendo y_k la salida esperada por la red y o_k la salida obtenida, se cambian los pesos desde el final hasta su comienzo, de ahí el nombre de “propagación hacia atrás”. Este proceso se repite hasta acabar con el conjunto de datos de

entrenamiento. El límite del cálculo se marca por un error mínimo alcanzable o una cantidad máxima de iteraciones preestablecida.

“Back propagation” utiliza la regla “Delta” para cambiar el valor de los pesos de la red. En esta regla, se utiliza el gradiente de la función del error, por lo que la función ha de ser derivable y lógicamente, continua. Se actualiza el peso con la dirección opuesta al gradiente del error, es decir, cambia su peso original por otro peso que reduzca el error cuadrático medio usando la pendiente. Se emplea el factor de aprendizaje “ α ” para asegurar que el error se hace más pequeño. La fórmula de la regla “Delta” es:

$$\Delta w_{ji} = \alpha \cdot (t_j - y_j) \cdot g'(h_j) \cdot x_i$$

Siendo los diferentes parámetros:

1. Δw_{ji} : Cambio del peso entre las neuronas i y j .
2. α : Factor de aprendizaje.
3. t_j : Valor deseado.
4. y_j : Salida de la red neuronal.
5. $g'(h_j)$: Derivada de la función de activación de la neurona evaluada en el punto h_j .
6. x_i : Entrada en la posición i .

5.2. ALEXNET

Existen diferentes redes neuronales disponibles para usarse en Matlab. La primera de ellas en orden de creación fue AlexNet. En su momento, esta red supuso una revolución para la inteligencia artificial porque fue una de las primeras aplicaciones prácticas del aprendizaje profundo. Hasta ese momento, la clasificación de imágenes se realizaba mediante algoritmos que, además de ser complejos a la hora de programar, eran sumamente concretos en función de la clasificación que se deseara realizar, por lo que su utilización no se podía generalizar. Además, los datasets que se empleaban en este tipo de algoritmos eran muy reducidos y estaban enfocados en clasificaciones pequeñas, por ejemplo, usando exclusivamente 2 categorías que se pudieran diferenciar sin complejidad en las imágenes.

No obstante, en AlexNet los conjuntos de datos de entrenamiento son claramente más grandes, por lo que se considera una red neuronal profunda, capaz de replicar el comportamiento cerebral en la clasificación de fotografías. Por este motivo, esta estructura demostró que una red neuronal convolucional mejoraba el rendimiento de los softwares de clasificación de imágenes tradicionales hasta la fecha, siendo capaz de entrenarse con grandes cantidades de datos diferentes para posteriormente realizar la clasificación con éxito. Entre sus logros más notables, destaca la consecución de la competición de reconocimiento

de imágenes a gran escala de ImageNet, en el año 2012. Este concurso, celebrado de manera anual hasta 2017, ha sido uno de los factores más importantes que han provocado el avance de la IA, especialmente en el reconocimiento de imágenes. No sólo ha permitido a los investigadores testear sus modelos de redes neuronales con datasets a gran escala, sino que también ofrecía una visión global, en esos momentos concretos, de la situación de la inteligencia artificial en el reconocimiento de imágenes.

La victoria de AlexNet acrecentó su fama y su utilización, ayudándose esta última a su vez en su arquitectura simple de 8 capas, 5 de ellas dedicadas a la extracción de características y las 3 capas restantes usadas para la toma de decisiones. Todas ellas emplean normalización ReLU excepto la capa final. A través de la función de activación ReLU, la salida es igual a la entrada en caso de que dicha entrada sea positiva. En caso contrario, la salida es nula. Gracias a la no activación de su salida en caso de que la entrada sea nula, el cálculo de la red es más eficiente al no activar todas las neuronas, lo cual en comparación con otras funciones de activación permite que se puedan emplear muchas otras capas sin suponer una elevada carga computacional.

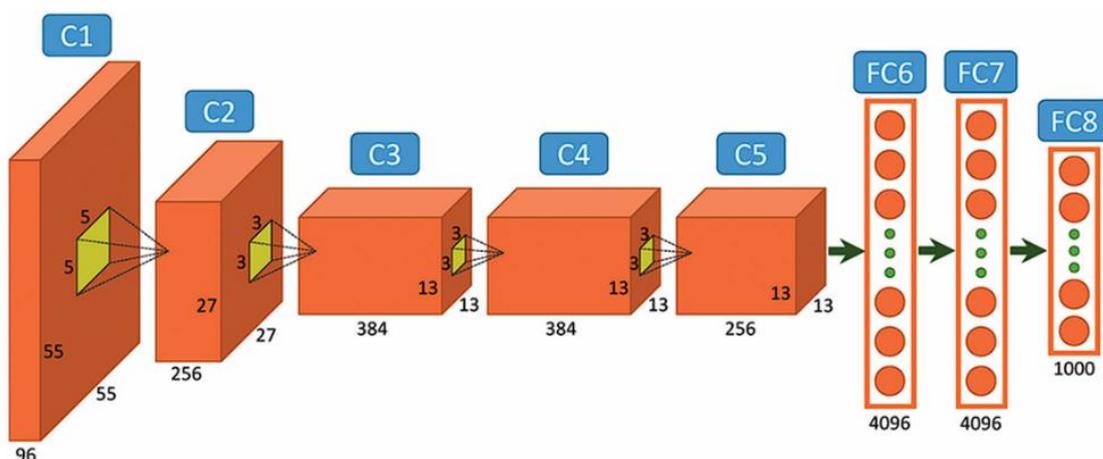


Figura 23. Representación gráfica de la red neuronal AlexNet

Fuente: [23]

A pesar de ser una arquitectura de red muy primitiva en comparación con otras redes neuronales posteriores, cuenta con algunas características especiales. Su principal particularidad es el empleo de “local response normalization”, o según sus siglas en inglés, LRN. “Local response normalization” imita parcialmente el proceso que desempeña la retina en el ojo de un ser vivo, es decir, emula la transformación de las imágenes en impulsos nerviosos que son enviados al cerebro. Mediante esta técnica el valor de la neurona excitada se normaliza en función de las neuronas cercanas, lo que modula la activación de diferentes

neuronas cercanas simultáneamente. Gracias a LRN, el modelo gana en estabilidad y reduce el riesgo de sobreajuste.

El empleo de capas de normalización en AlexNet es debido a la necesidad de regulación de las activaciones, especialmente en capas ReLU porque sus salidas no están acotadas. El proceso de LRN hace que ciertas neuronas que tienen valores de activación más elevados tengan una mayor importancia en el conjunto de la red que otras neuronas. Esto hace que la información se aproveche de una manera más eficiente, al poder ser filtrada dependiendo de su relevancia.

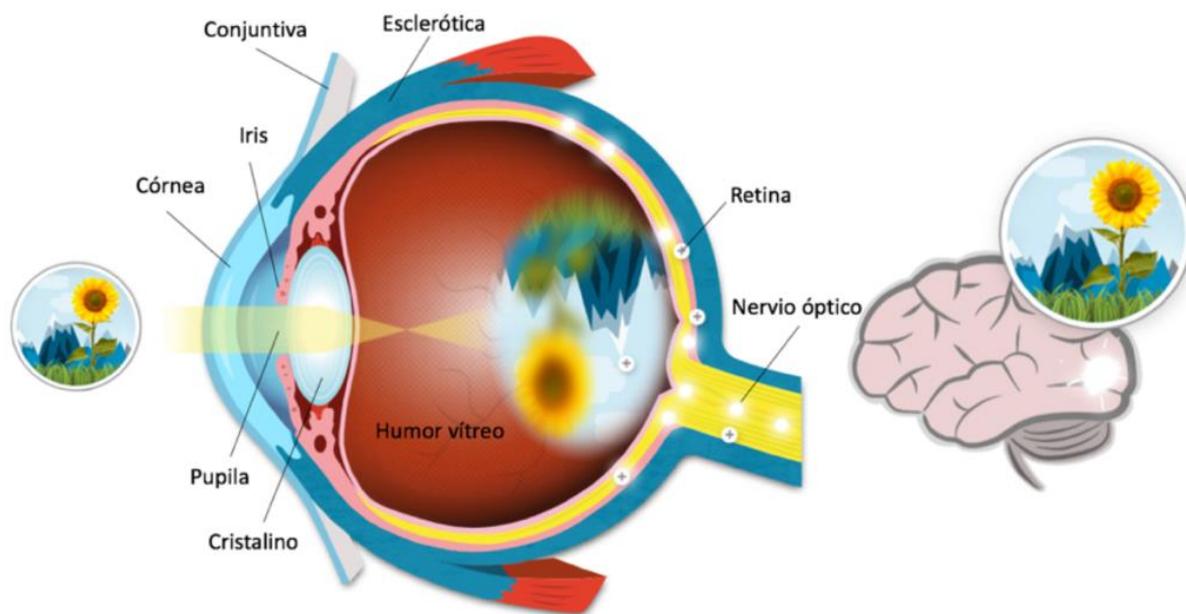


Figura 24. Representación gráfica del funcionamiento de la vista humana

Fuente: [24]

Además de estas novedades, AlexNet fue la primera red neuronal que empleó diferentes técnicas con el objetivo de mejorar el proceso de aprendizaje de la red, previniendo el overfitting. Esta problemática ocurre cuando la red neuronal se ciñe excesivamente a los datos de entrenamiento, lo que hace que no sea capaz de generalizar cuando emplea datos cuyas características sean ligeramente diferentes. Para evitarlo, en primer lugar, se desarrollan capas de ReLU Activation, mediante las cuales se añade no linealidad a la red neuronal. Cuenta con otras técnicas como Data Augmentation, que aumenta el conjunto de datos de entrenamiento, de forma que la red neuronal pueda aprender más información. Si esto no se realiza, la estructura puede ser incapaz de generalizar cuando emplea datos nuevos. Otro procedimiento más avanzado, aunque también extensamente utilizado es “Dropout”, que previene el sobreajuste haciendo que ciertas neuronas de la red dejen de funcionar de manera aleatoria. Este apagado se realiza en cada iteración del entrenamiento

y gracias a él, la red neuronal en su conjunto reduce la cantidad de características específicas, lo cual ayuda a una mejor generalización en las clasificaciones.

Entre otras nuevas utilidades, también resalta “overlapping pooling”, desarrollada con el objeto de reducir la dimensión de las características mediante el solapamiento de las regiones de los mapas, así como modificaciones en los valores de los canales R, G y B.

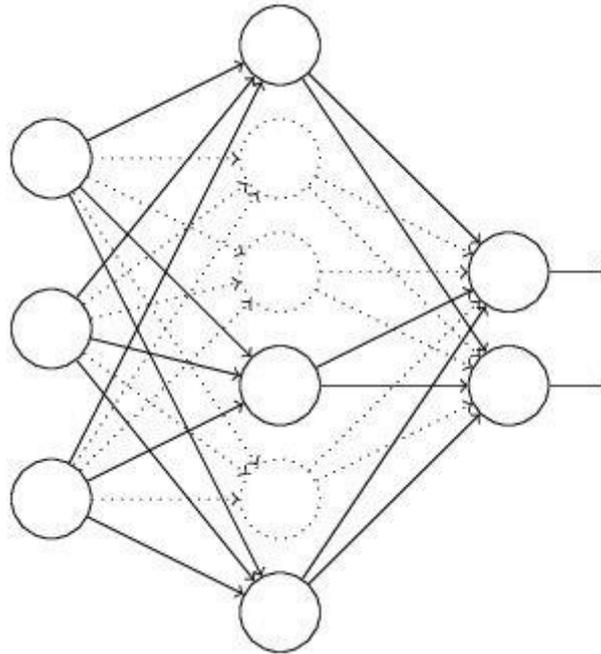


Figura 25. Representación del proceso de Dropout en una red neuronal

Fuente: [25]

El surgimiento de AlexNet ha sido de gran utilidad en el desarrollo de hardware aplicado a redes neuronales, gracias al empleo de GPUs. Estas unidades, que también reciben el nombre de “tarjetas gráficas”, aceleran el proceso de reconocimiento de imágenes y procesamiento de gráficos en cualquier dispositivo, desde placas base hasta ordenadores, pasando por los teléfonos móviles. Desde su utilización en esta red neuronal, las GPUs han estado presentes en el desarrollo de otras redes neuronales convolucionales posteriores, pasando a ser un elemento clave en el desarrollo y en la popularidad de la IA y el Deep Learning. Tal ha sido el desarrollo de esta tecnología que sus recursos están optimizados en comparación con las CPUs, ya que requieren de menos recursos para desempeñar sus funciones sin bajar su potencia y su precisión.



Figura 26. Ejemplos de GPUs de Intel

Fuente: [26]

Igualmente, esta red neuronal permitía hacer uso de una gran cantidad de imágenes para su clasificación, que podían ser procesadas de mejor manera gracias a su arquitectura. Dentro de ese procesado, AlexNet puede detectar bordes, texturas y formas. No obstante, tuvo limitaciones desde el comienzo de su desarrollo, ya que se necesitaba mucha memoria para su entrenamiento. A fecha de hoy, la implementación de AlexNet permite la clasificación de imágenes en 1000 categorías diferentes, y la simpleza de su arquitectura ha contribuido a que se desarrollen nuevas redes neuronales más sofisticadas y con más capas, tomando esta estructura como referencia.

En la práctica, AlexNet está enfocada al reconocimiento de objetos, usando la base de datos de ImageNet, que cuenta con imágenes de categorías muy diferentes entre sí, con un amplio abanico que se extiende desde teclados de ordenador hasta animales, como instantáneas de perros labradores. No obstante, es capaz de clasificar otros tipos de imágenes como plantaciones con plumero de la Pampa en caso de realizarse su entrenamiento correspondiente, siendo una alternativa que analizaré posteriormente.

5.3. SQUEEZENET

Más allá de las redes neuronales explicadas previamente, he decidido usar SqueezeNet como una de las redes neuronales para la detección del plumero. Fue diseñada en el año 2016 por parte de miembros de una empresa de tecnologías de sistemas de percepción para automóviles llamada DeepScale, junto con la colaboración de la universidad de Stanford y de la universidad de Berkeley, en California. Posee 68 capas, pero contando únicamente capas de profundidad, la arquitectura se reduce a 18, siendo su objetivo alcanzar un equilibrio entre una alta precisión y una baja complejidad, lo que hace que sea una red neuronal empleada también en sistemas embebidos, aunque en menor medida que ShuffleNet.

Tanto por su nombre (“squeeze” significa “apretar” en inglés) como por el bajo número de capas, es fácil imaginar que es una red neuronal compacta. De hecho, la compactibilidad de esta red neuronal fue provocada por la búsqueda de redes neuronales de menor tamaño, y con menor ancho de banda, pero conservando prestaciones de redes neuronales anteriores, como AlexNet. En tareas de clasificación de imágenes, SqueezeNet es capaz de reducir 50 veces la cantidad de parámetros, sin utilizar procesos de cuantización. En caso de emplearlos, la compresión de la información podría ser 400 veces mayor que usando AlexNet, lo que conlleva una clara mejora en el tratamiento de la información. A pesar de estas reducciones drásticas de tamaño de los datos, SqueezeNet es capaz de reconocer más de 1000 objetos, lo cual aumenta su practicidad y fomenta su utilización.

Su alta potencia, otro de sus puntos a favor, se debe en parte a su ligereza en términos de memoria, y además, es capaz de comprimir los datos reduciendo el número de pesos y sesgos de la red neuronal y después, expandir dichos datos. Este proceso, que constituye la clave de la estructura de la red, se conoce como módulo Fire. En la fase de compresión de este módulo, se sustituyen los filtros 3x3 por filtros 1x1, lo cual reduce el número de parámetros junto con el consumo de memoria, optimizando el procesamiento. La reducción de características ocasionada por la fase de compresión es tan significativa que recibe coloquialmente el nombre de “cuello de botella”. Después, en la fase de expansión, parte de la información que es clave para la clasificación se reconstruye usando tanto filtros 1x1 como 3x3. Al término de sucesivos módulos Fire se alcanza el final de la red neuronal, donde se realiza la clasificación de las imágenes utilizando sólo la información importante.

Gracias a este sistema, SqueezeNet prescinde de información espuria y redundante y, en definitiva, puede aumentar la rapidez de la red en base a la reducción de parámetros, pero sin comprometer la precisión. Además, este sistema es clave para la posibilidad de desarrollo de SqueezeNet en dispositivos con menores recursos computacionales. Todo esto hace que la red se diferencie del resto de estructuras neuronales artificiales en la optimización de la información.

El perfeccionamiento a lo largo de los años tanto de la tecnología del módulo Fire como de la arquitectura de la red ha dado lugar a diferentes versiones, entre ellas SqueezeNet v1.1, que mejora su precisión un 1'4% usando el conjunto de datos de entrenamiento de ImageNet, o SqueezeNet v2, que mejora su rendimiento alrededor de un 2% reduciendo 2'4 veces la cantidad de parámetros gracias a los módulos Fire.

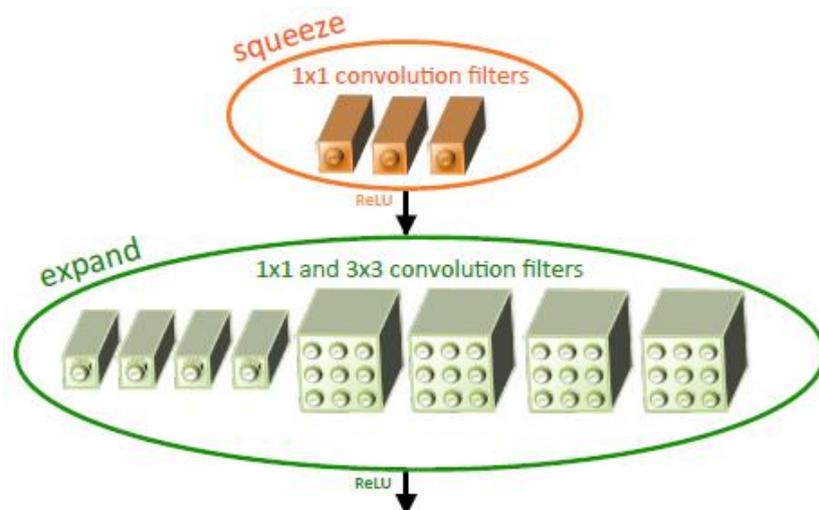


Figura 31. Representación de un módulo Fire

Fuente: [31]

Además de los métodos explicados para agilizar el trabajo de la red, existen otros métodos de optimización y reducción, especialmente enfocados en la implementación de redes neuronales con arquitecturas grandes en dispositivos móviles o microcontroladores.

El primer método de ellos es la poda o "pruning". A través de esta técnica, se eliminan tanto conexiones como neuronas que la red no necesita para el desempeño de su tarea, sin empeorar el rendimiento. Adicionalmente, también ajusta el valor de los pesos de las neuronas restantes, para que la estructura en su conjunto no sufra variaciones en su funcionamiento. De esta forma, la red neuronal no sólo aprende la información importante, sino que también prescinde de conexiones innecesarias. El "pruning" está también presente en la red AlexNet con el mismo objetivo, la agilización de su funcionamiento prescindiendo de conexiones innecesarias o redundantes entre las capas, así como de neuronas que no intervengan en la tarea a realizar.

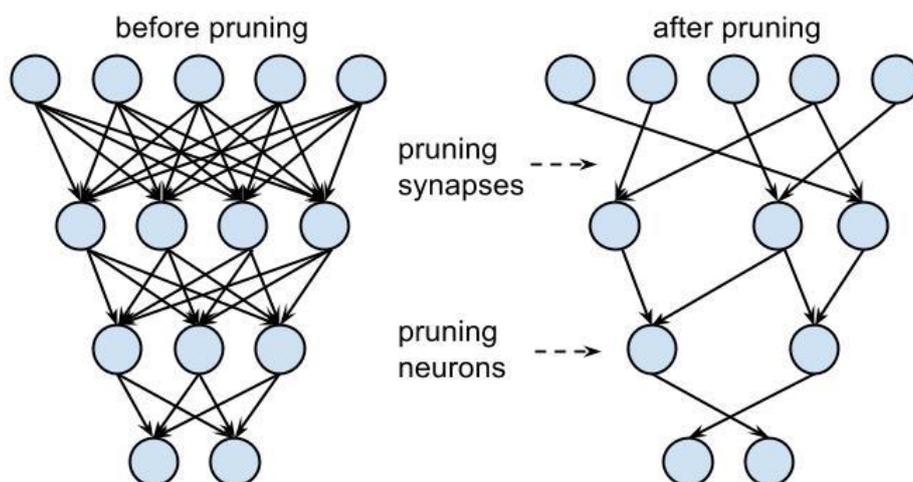


Figura 32. Representación de la red neuronal resultante después del proceso de poda

Fuente: [32]

Por otra parte, el proceso de cuantización reduce la precisión de los pesos de las neuronas de la red, generalmente, desde los 32 bits hasta los 8 bits. Esto provoca que la red utilice menos cantidad de memoria al bajar el número de bits por peso, aunque es probable que varias neuronas de la red tengan pesos comunes. Esta modificación es útil porque a pesar de la posibilidad de que la precisión de la red neuronal en la tarea de clasificación de imágenes baje, la velocidad en el entrenamiento aumenta considerablemente. En definitiva, el proceso de cuantización agiliza la red y reduce el tiempo de inferencia, aún a costa de poder empeorar el rendimiento. Es importante valorar que los procesos de poda y cuantización son independientes, es decir, una red neuronal SqueezeNet se puede mejorar a través de "pruning" sin haber hecho el proceso de cuantización, y viceversa.

El tiempo de inferencia es el tiempo que tarda la red neuronal, después de haber sido entrenada, en razonar y extraer conclusiones de datos con los que no había trabajado hasta ese momento. Esto implica que la inferencia es un proceso que supone un paso más allá en el desarrollo del programa, inmediatamente posterior al entrenamiento. En el ámbito de las redes neuronales, ambos conceptos son fácilmente confundibles. Sin embargo, la mayor diferencia entre ambos procedimientos es que durante la inferencia, la red neuronal ya entrenada trabaja con datos hasta ese momento completamente desconocidos para ella y no requiere de una gran cantidad de información, al contrario que en el entrenamiento.

Existen dos tipos de inferencia:

1. Por lotes: recibe este nombre porque es la forma en la que se procesa la gran cantidad de datos recopilados. Es utilizada en redes neuronales cuya tarea no se desarrolla en

tiempo real, es decir, no se necesita una salida de inmediato. Se optimizan los recursos utilizando múltiples datos a la vez.

2. En línea: Es empleada en redes neuronales que desempeñan tareas en tiempo real, como la conducción autónoma de vehículos. Necesita baja latencia y respuestas inmediatas. Esto dificulta su optimización, lo que hace que consuma más recursos que la inferencia por lotes.

El aumento de velocidad resultante de los procesos anteriormente mencionados convierte a SqueezeNet en una buena alternativa para tareas de inteligencia artificial que emplean este último tipo de inferencia. No obstante, su simpleza también puede acarrear problemas tales como la necesidad de un entrenamiento más duradero y complejo que otras redes neuronales preentrenadas más grandes enfocadas en este fin para obtener unas prestaciones similares ante un mismo problema.

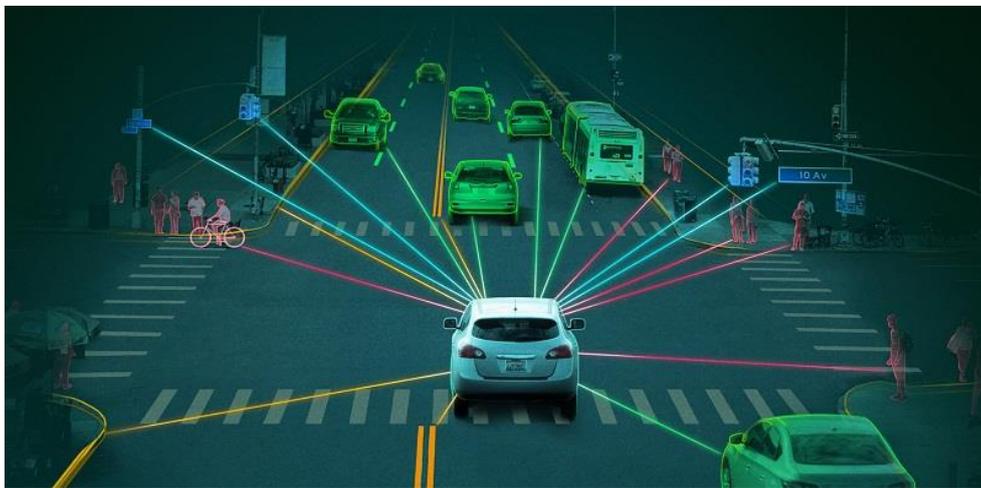


Figura 33. Ilustración de la detección de objetos y personas en la conducción

Fuente: [33]

Además de lo que permite la red neuronal en sí, SqueezeNet se puede combinar con otras tecnologías como YOLO, por ejemplo, en softwares de reconocimiento facial. Uno de los casos más claros de esta combinación se ha desarrollado en centros de seguridad, centros de salud y hospitales, especialmente durante la pandemia del Covid-19, como medida de verificación para la correcta colocación de las mascarillas quirúrgicas.



Figura 34. Ejemplo del uso de SqueezeNet y YOLO en el uso correcto de mascarillas

Fuente: [34]

En el caso de este trabajo, el uso de SqueezeNet se verá enfocado en el reconocimiento de ejemplares de cortadería selloana en fotografías satelitales.

5.4. REDES RESIDUALES

Las redes neuronales residuales se basan, como otras muchas redes neuronales, en AlexNet. El origen de este tipo de estructuras se debió a la problemática de que, al añadir nuevas capas a esta red neuronal, su rendimiento empeoraba al modificarse los datos excesivamente. Para evitar esto, se debía construir a partir de AlexNet usando capas adicionales, que son simples funciones de identidad, que no alteran la información. Estas funciones o mapeos de identidad ayudan a propagar la información sin modificaciones tanto hacia adelante como hacia atrás, así como evitan que el rendimiento baje a pesar de emplear más capas que otras redes.

Existen distintos tipos de redes neuronales residuales, dependiendo de su profundidad. Dentro de ellas, destacan ResNet-18, ResNet-34, ResNet-50, ResNet-101 y ResNet-102, mientras que también hay redes neuronales residuales mejoradas, como ResNext.

Uno de los grandes avances que posee este tipo de red neuronal es que, en lugar de tratar de aprender directamente la imagen de entrada, aprende de los residuos. Estos residuos, que en la práctica son pequeñas modificaciones que se hacen directamente en la imagen, no suponen más que la diferencia existente entre la entrada real y la salida esperada, mientras que otras redes neuronales, tanto anteriores como posteriores, captan la transformación completa que se desarrolla desde el “input” hasta el “output”. Cada una de las capas de la red neuronal residual, que posteriormente compone los bloques residuales, va modificando la instantánea progresivamente, de forma que la red neuronal en su conjunto aprenda mucho más que usando la fotografía original sin ningún tipo de modificación gracias a las conexiones de salto que pueden superar varias capas sin modificar la información, hasta el punto de que

no se obtiene la salida en función de la entrada original, sino de los residuos. Los bloques residuales son tan importantes en esta arquitectura que bautizan a la red neuronal, ya que ResNet es la abreviatura de “Residual Network” en inglés. Esta relevancia de las conexiones de salto y de los bloques residuales en ResNet es idéntica a la importancia de los módulos “Fire” en SqueezeNet, ya que ambos procedimientos edifican las estructuras.

Teniendo en cuenta que en las imágenes de las que dispongo existe una gran variedad de particularidades, tales como elementos urbanos o rurales, cosechas varias cercanas a la cortadería sellos, otros tipos de vegetación o árboles, tanto la red neuronal residual como la máquina de vectores de soporte en su conjunto mejoran sus aprendizajes gracias a las continuas modificaciones que realiza ResNet.

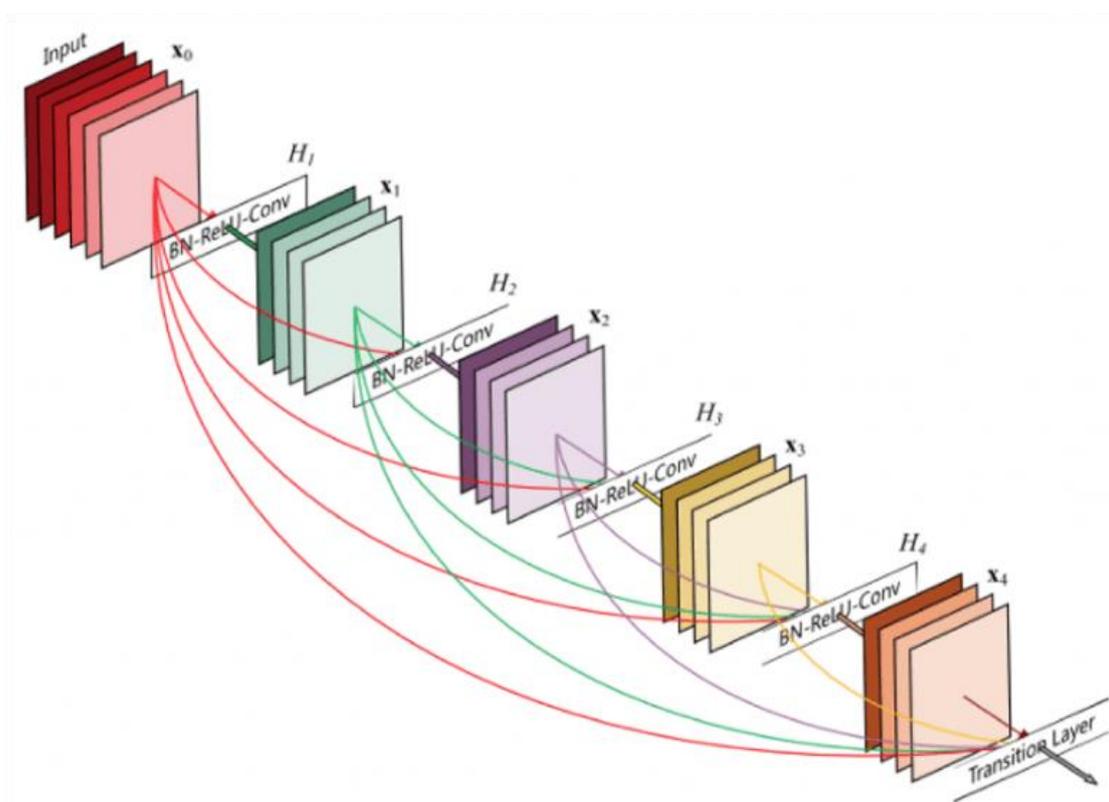


Figura 36. Ejemplo de capas de una red residual

Fuente: [36]

Un problema existente en otras redes neuronales anteriores a ResNet-50 es el desvanecimiento del gradiente, razón añadida por la cual se crearon las redes neuronales residuales. En una red neuronal, el gradiente es la variación de la función de pérdida con respecto a la variación de los pesos, es decir, nos indica si la solución para bajar el error del modelo es aumentar o disminuir un peso determinado. El gradiente puede desaparecer por culpa de la existencia de neuronas en la red que cuentan con tan poco peso que no varían,

lo cual provoca que no aprendan de manera correcta y se pierda información. Por consiguiente, pueden obtenerse las características de las imágenes en las primeras capas de forma errónea, lo que acaba repercutiendo en el resto de la red neuronal. El desvanecimiento del gradiente puede ocurrir en cualquier entrenamiento de cualquier tipo de red neuronal profunda. Es más, cuanto mayor profundidad posee, peor se actualizan los pesos de las neuronas. Es un problema sensible, ya que la variación del gradiente está directamente relacionada con la variación de los pesos de la totalidad de la red.

Para evitar el desvanecimiento del gradiente, existen varias soluciones. La opción principal es que la información atraviese menos capas porque cada una de ellas transforma la entrada y los pesos se reducen. Para ello, se realizan las conexiones de salto, también llamadas “skip connections” en inglés.

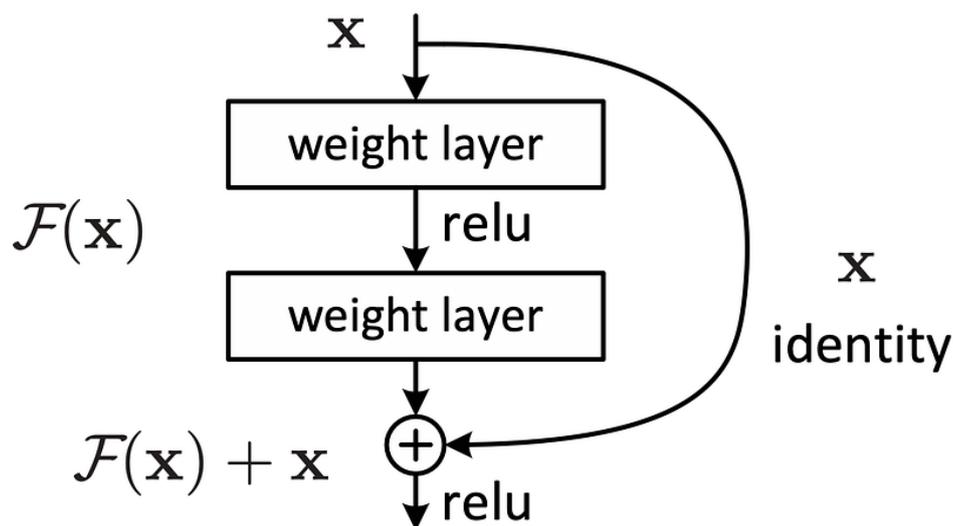


Figura 37. Representación de una conexión de salto

Fuente: [37]

Según se muestra en la representación anterior, el valor de $F(x)+x$ es la salida del bloque residual siendo $F(x)$ el residuo o, en otras palabras, lo aprendido por la red neuronal, que es la diferencia entre la salida y la entrada.

El desvanecimiento del gradiente también puede solucionarse cambiando ligeramente las funciones de activación ReLU.

$$f(x) \begin{cases} 0 & \text{si } x \leq 0 \\ x & \text{si } x > 0 \end{cases}$$

En su lugar, se puede usar la función Leaky ReLU.

$$f(x) \begin{cases} 0.01x & \text{si } x \leq 0 \\ x & \text{si } x > 0 \end{cases}$$

Empleando este tipo de función nunca se obtendrá un valor nulo para posibles valores negativos en la entrada, lo cual asegura que la red neuronal siempre obtenga resultados coherentes. Otra potencial solución es utilizar el método de poda de forma que se reduzca el tamaño de la red neuronal, quitando pesos de ciertas partes que no funcionan bien o que no son necesarias para el desempeño de la tarea, lo cual además de mejorar el comportamiento de la red neuronal, optimiza sus prestaciones agilizando la obtención de los resultados.

En definitiva, una red neuronal ResNet-50, en lugar de aprender directamente toda la información, aprende residuos gracias a las conexiones residuales haciendo que los datos que viajan no varíen y se produzca una fragmentación de la información más eficiente que analizar toda la imagen por completo. Esto implica una clara mejora en el rendimiento de la red. Además de ser utilizada en el código de la máquina de vectores de soporte, evaluaré una posible solución de la utilización de ResNet-50 para reconocer cortadería selloana en las imágenes.

5.5. SHUFFLENET

Existen otro tipo de redes neuronales utilizadas únicamente en dispositivos con limitaciones de memoria, potencia o procesamiento, como ShuffleNet. Este tipo de red neuronal fue ideada en 2017 con el objetivo de que fuese utilizada en dispositivos móviles, como teléfonos o drones, convirtiéndose en la red neuronal de referencia para sistemas embebidos. Al igual que las anteriores redes neuronales, es utilizada para reconocimiento de objetos en fotografías.

Al comienzo del flujo de la red neuronal, se usa una convolución grupal para filtrar los canales. Este procedimiento se realiza en paralelo, para aumentar la velocidad. No obstante, esta convolución limita la comunicación entre los canales. Con el objetivo de solucionarlo, se realiza una técnica llamada “mezclado de canales”, o como sería su traducción al inglés, “channel shuffle”. De esta forma, se optimiza el coste operacional y se reduce el tiempo de ejecución de la tarea, sin comprometer la precisión de la predicción.

A su vez, el auge y el desarrollo de los dispositivos móviles en todo el mundo, ha hecho que ShuffleNet se extienda mucho más. Este proceso se realiza tras una serie de convoluciones, donde se extraen las características de las imágenes. Es importante mencionar que las convoluciones que realiza ShuffleNet son principalmente 3x3. Este tipo de convoluciones es el más eficiente porque requiere de menos parámetros y menos cálculos para obtener resultados similares a las convoluciones 5x5 o 7x7, pero usando menos memoria y agilizando el conjunto de la red. También utiliza convoluciones 1x1 en grupo, pero en menor medida.

En comparación con AlexNet, esta red neuronal alcanza velocidades mucho mayores. Si bien puede pensarse que, al ser utilizada en dispositivos que cuentan con capacidades de cálculo limitadas, su precisión se ve mermada. No obstante, la optimización de su desarrollo y su específico enfoque en dispositivos con memorias pequeñas ha evitado esta problemática.

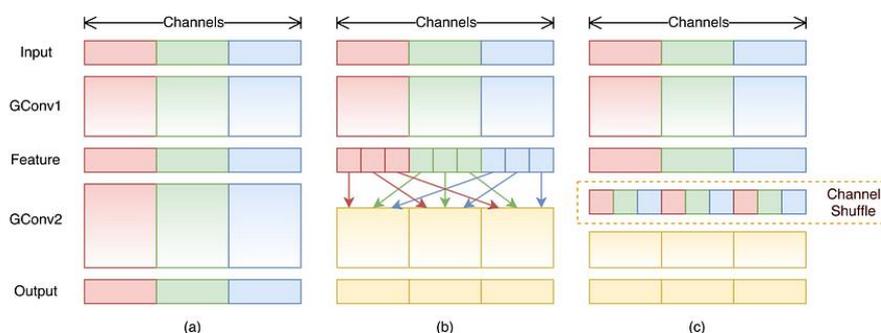


Figura 29. Ilustración teórica del mezclado de canales

Fuente: [29]

El surgimiento de ShuffleNet se debe a la necesidad de buscar redes neuronales capaces de ser empleadas en dispositivos embebidos. Esta clase de dispositivos forman parte de equipos mayores, desempeñando tareas específicas dentro de ellos. Sin la existencia de los sistemas embebidos, la tarea en su conjunto no se podría realizar. Gracias al continuo desarrollo de la electrónica y a su vez, de los microprocesadores, la optimización de redes neuronales específicas para este tipo de dispositivos está en auge.



Figura 30. Ejemplos de aplicaciones de sistemas embebidos

Fuente: [\[30\]](#)

5.6. DARKNET

La red neuronal DarkNet, al igual que la red AlexNet, es usada para identificación y localización de objetos, pero a diferencia de otras redes neuronales, no hace predicciones de clases catalogando la imagen en su conjunto, sino que reconoce objetos individualmente dentro de la propia imagen. Al trabajar con la imagen completa, se generan menos falsos positivos que utilizando otras estructuras a la hora de la clasificación, ya que DarkNet analiza las fotografías con una mayor profundidad y no se predice una categoría para la imagen global.

Para la detección de los elementos en la imagen, este tipo de red neuronal emplea un algoritmo denominado YOLO (You Only Look Once). Siguiendo este método convolucional de una etapa de alta rapidez y bajo consumo, se fracciona la imagen principal en distintas imágenes y se otorgan diferentes probabilidades a cada parte de manera individual. Al hacer esta división, aumenta la velocidad y la precisión de la red neuronal, siendo capaz de generalizar las representaciones de los objetos pudiendo detectar más de 9000 clases, lo cual diferencia a DarkNet del resto de estructuras de manera positiva. Estas generalizaciones aportan robustez al conjunto de datos de entrenamiento, independientemente de la introducción de nuevas imágenes.

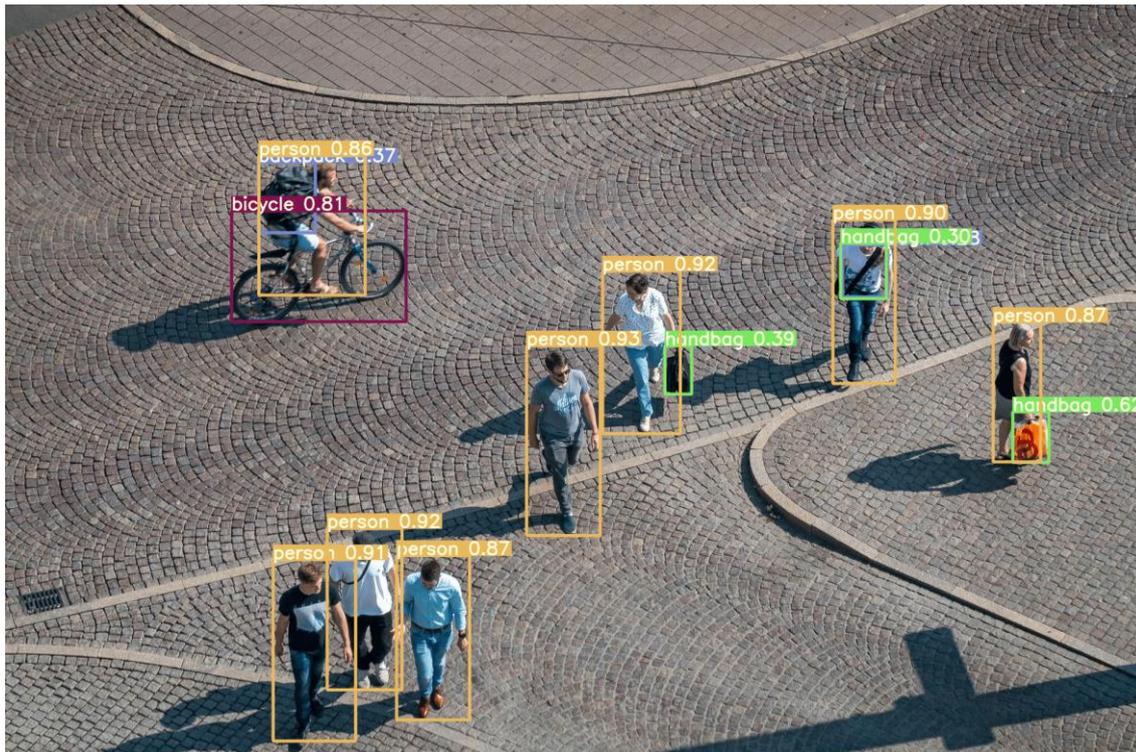


Figura 27. Representación del empleo de la tecnología YOLO en una imagen

Fuente: [27]

Siguiendo esta técnica, se divide la imagen introducida en una cuadrícula compuesta de celdas. Después, se generan las “bounding boxes” (o en español, cajas limitadoras) y se calcula la probabilidad de cada una de ellas, también llamado “score”. Cada uno de los recuadros ofrece una serie de información completa para poder determinar la puntuación de confianza de la detección a través de una fórmula. Dicha información es medible a través de los siguientes parámetros:

1. x: Posición del objeto en el eje horizontal respecto a un origen de coordenadas conocido.
2. y: Posición del objeto en el eje vertical respecto a un origen de coordenadas conocido.
3. w: Anchura del objeto detectado.
4. h: Altura del objeto detectado.

Gracias a esta información, la red neuronal emplea la siguiente ecuación en cada parcela de la cuadrícula:

$$\text{Score} = P(\text{Class} | \text{Object}) \cdot P(\text{Object}) \cdot \text{IoU}_{\text{pred, truth}} = P(\text{Class}) \cdot \text{IoU}_{\text{pred, truth}}$$

Esta medida de la confianza se realiza en todas las celdas de la imagen. Para ello, se calculan anteriormente las probabilidades de clase condicional, la probabilidad de que haya un objeto

en una celda y las cajas delimitadoras, con sus correspondientes coordenadas y los valores de confianza. Cabe nombrar que la fórmula anterior emplea el teorema de Bayes. Según este teorema, se calcula la probabilidad de un suceso, teniendo en cuenta una probabilidad previa o condicional. En el caso de la tecnología YOLO, se calcula la probabilidad de que un objeto sea de una determinada clase, sabiendo previamente que existe un objeto en esa zona correspondiente de la cuadrícula, lo cual supone una probabilidad condicionada. Si el resultado de la ecuación, es decir la confianza, es alta, se puede deducir que existe un objeto en esa área con la categoría predicha correctamente. No obstante, en caso de que el resultado sea bajo, la predicción se descarta, siendo DarkNet capaz de prescindir de las probabilidades que no alcanzan un valor mínimo previamente definido, también llamado umbral.

A diferencia de otras redes, para su entrenamiento se requiere de archivos de texto con las coordenadas de los diferentes elementos existentes en dichas imágenes, junto con las propias fotografías. Al ser una forma de entrenamiento tan radicalmente diferente a otras redes y tan específica, existen páginas web capaces de generar conjuntos de datos de entrenamiento específicos para DarkNet.

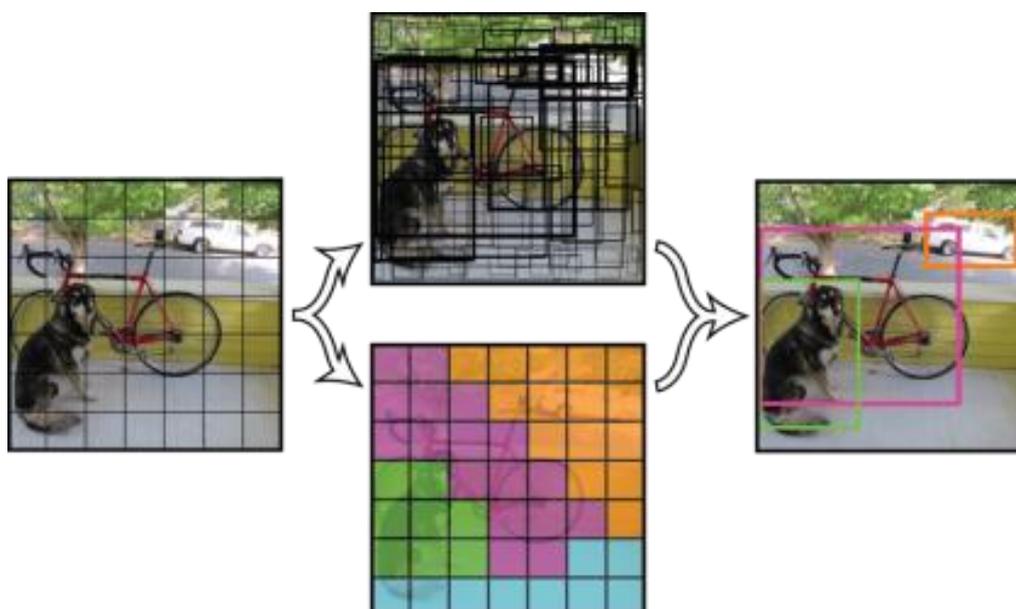


Figura 28. Representación del funcionamiento de un algoritmo YOLO

Fuente: [28]

En comparación con otras redes neuronales, la arquitectura de DarkNet es bastante simple porque únicamente posee una serie de capas convolucionales y dos capas completamente conectadas. La simpleza de su estructura, además del algoritmo que emplea, es también una razón de su rapidez de reconocimiento.

Desde la creación de YOLOv1 en el año 2016, esta metodología destacó por su velocidad. No obstante, el mayor reto al que se ha enfrentado esta técnica es la precisión, obteniendo rendimientos ligeramente inferiores a otras redes neuronales de su momento, como le ocurrió a YOLOv7 en 2023. Aun así, se puede determinar que cumple con creces el compromiso entre velocidad y precisión. El desarrollo de la tecnología YOLO ha llegado a permitir la detección de elementos partiendo de archivos de vídeo. Mediante la realización de capturas o la extracción directa de "frames", DarkNet permite el reconocimiento de objetos o de personas. Esto hace que dicha red neuronal sea utilizada también en sistemas de seguridad de diferentes negocios y establecimientos.

Principalmente, el campo de utilización más importante de DarkNet es la sanidad, gracias a su capacidad de detección de diferentes anomalías en fotografías. Esto permite realizar diagnósticos eficaces sin la necesidad de grandes medios, ya que en una misma imagen se pueden detectar distintas marcas que puedan ser indicadoras de una enfermedad o una lesión. Su tecnología sigue desarrollándose de la mano de otros campos como la conducción autónoma de automóviles, dada la necesidad del reconocimiento de otros elementos de la carretera, así como de transeúntes.

5.7. CONCLUSIONES

Como se ha demostrado, el estado del conocimiento en lo referido al reconocimiento de imágenes mediante IA es avanzado. Desde el surgimiento de AlexNet, el estudio y la evolución de las redes neuronales ha ido desarrollándose hasta nuestros días.

La generación de redes neuronales que más allá de la identificación de imágenes, como SqueezeNet o las redes residuales, es capaz de reconocer diferentes elementos dentro de una misma fotografía, como DarkNet, demuestra el compromiso por el desarrollo y las mejoras en estas tareas. Por otra parte, considerando que la sociedad actual requiere de mejoras constantes en el rendimiento de los medios tecnológicos de los que se dispone, es razonable pensar que tanto la inteligencia artificial como el análisis de instantáneas continuará desarrollándose en vías generales, pero sin renunciar a reconocimientos más específicos.

6. SVM

6.1. INTRODUCCIÓN

SVM son las siglas de Support Vector Machine, o en español, máquina de vectores de soporte. Es un método de clasificación en el cual se busca determinar el hiperplano óptimo, que es el nombre que recibe la separación máxima entre las diferentes clases. Se busca

maximizar esta distancia porque ayuda al programa a generalizar el conjunto de datos. Esto es debido al hecho de que poseer un margen amplio hace que el modelo en su conjunto sea mucho más robusto a variaciones, ruido y se reduce ostensiblemente la probabilidad de obtener una mala clasificación.

Este algoritmo es empleado en problemas de reconocimiento de imágenes y de clasificación binaria, como es el caso del TFG, ya que se puede clasificar entre imágenes que poseen plumero de la Pampa e imágenes sin cortadería selloana. Explicado de una forma matemática, se usa la siguiente ecuación:

$$\beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \dots + \beta_p \cdot x_p = 0$$

En esta fórmula, los distintos valores de x determinan las coordenadas de un punto en el espacio en cuestión, mientras que los coeficientes β son los parámetros geométricos del plano. Si un punto determinado cumple esta ecuación, dicho punto está sobre el hiperplano, por lo que no pertenecería a ninguna clase. En caso de que no se cumpla, dependiendo de si el resultado es positivo o negativo, dicho punto estará a un lado u otro del hiperplano, es decir, pertenecerá a una clase o a otra. Por este motivo, también es muy útil al ser capaz de destacar valores atípicos en grupos, lo cual dependiendo del estudio puede ayudar a reconocer anomalías o errores en conjuntos de datos.

El hiperplano es un “espacio de n dimensiones”. En un espacio de dos dimensiones, como puede ser una simple imagen, dicho espacio es una línea recta o curva. No obstante, en un espacio de tres dimensiones, el hiperplano es un plano. La máquina de vectores de soporte se encarga de orientar dicho hiperplano tan lejos como sea posible de los puntos más cercanos de cada una de las clases, también llamados “datos críticos” o “vectores de soporte”. Su relevancia es debida a que el hiperplano se debe situar en el punto medio entre ambos, y su hipotética eliminación haría que la localización de dicho hiperplano se modificara. La situación de los datos críticos ayuda a corroborar el correcto diseño del hiperplano, ya que, si se puede verificar que están correctamente clasificados en su grupo, el resto de los datos de cada clase también lo estará.

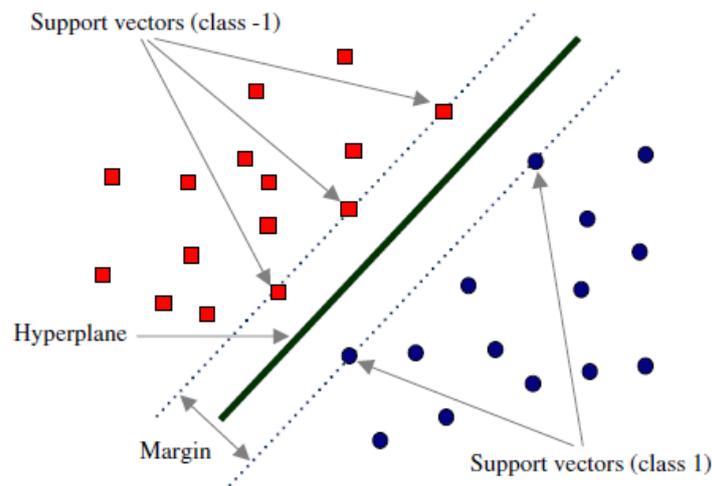


Figura 35. Representación de un hiperplano y sus partes más características

Fuente: [35]

Según la ilustración anterior, la línea negra sería el hiperplano óptimo que separa las clases, marcadas como cuadrados rojos y círculos azules, mientras que los datos de dichas categorías más cercanos al hiperplano son los vectores de soporte de cada uno de los dos tipos. Debido a su proximidad al hiperplano se consideran datos críticos, porque su hipotética eliminación afectaría directamente a la ubicación de la frontera entre las clases. La complejidad de la obtención del hiperplano óptimo guarda una proporcionalidad directa con la cantidad de datos que se tengan por clase, independientemente de que sean posteriormente considerados datos críticos o no.

A pesar de ser utilizado en este trabajo como otra alternativa a las redes neuronales, el código de SVM emplea a su vez una red neuronal, que en este caso es ResNet-50, que es una estructura neuronal residual empleada especialmente para el reconocimiento de imágenes.

6.2. CONCLUSIONES

Aunque en este trabajo y en términos generales la máquina de vectores de soporte sea una herramienta que sea más desconocida en la IA en beneficio de las redes neuronales, se puede asegurar que es una herramienta útil en tareas de clasificación de datos. A pesar de ser habitualmente usada como sistema de clasificación de datos, SVM es una metodología de gran utilidad a la hora de evaluar conjuntos de imágenes y detectar posibles errores.

En este proyecto, se utilizó como alternativa a otras redes neuronales para el reconocimiento de cortadería selloana, pero su fácil código se puede adaptar a otras posibles funciones de una manera muy sencilla, dependiendo del dataset del que se disponga y del objetivo a cumplimentar.

7. ENTRENAMIENTO

7.1. INTRODUCCIÓN

Para el correcto cumplimiento de una tarea por parte de una red neuronal, es necesario que exista un entrenamiento. Se conoce por este nombre al proceso que permite que la red neuronal responda de manera autónoma al recibir distintos valores de entrada. De una manera simplificada, implica enseñar a la red neuronal una tarea en concreto, como puede ser el reconocimiento de plantas en imágenes. Para ello, se otorgan diferentes pesos a las neuronas de la red, de forma que las respuestas a la salida se encuentren dentro de un espectro definido con anterioridad, mediante una serie de categorías. Para conseguirlo, también se debe disponer de un conjunto de datos etiquetados en carpetas, lo cual hace que se catalogue el aprendizaje como supervisado. Como se mostrará con posterioridad, los nombres de las variables de salida son los nombres de dichas carpetas. Cuanto mayor sea la especificación de la clasificación a realizar, más avanzada será la red neuronal en cuestión y más compleja será su arquitectura y su funcionamiento.

7.2. ENTRENAMIENTO DE SQUEEZENET

Para comenzar con el código, se carga la red neuronal y las imágenes. Se genera un datastore o almacén de datos, habiendo descomprimido el archivo donde están originalmente las imágenes a emplear. Se declara que las clases son los nombres de las carpetas que existen dentro de dicho archivo. En este caso, “plumero”, donde están almacenadas las imágenes satelitales con cortaderia selloana y “no plumero”, donde hay imágenes satelitales que no cuentan con ese tipo de especie vegetal.

Posteriormente, se define un porcentaje de imágenes para entrenamiento, es decir, la cantidad de imágenes empleadas por la red neuronal para captar la apariencia de un cultivo con cortaderia selloana. El porcentaje restante determina las imágenes que serán usadas para la validación, es decir, para determinar la precisión de la red neuronal, verificando que las predicciones coinciden con las categorías reales de las imágenes.

Para adaptar la red neuronal preentrenada a la tarea de la detección de cortaderia selloana, se deben reemplazar una serie de capas. En primer lugar, la capa “conv10” es cambiada por una nueva capa convolucional ya que debe tener tantos filtros como clases, es decir, 2 (llamado “numClasses” en el código). De esta forma, se puede garantizar que la red neuronal sólo ofrece esas posibles soluciones en la salida. Además, se fija el tamaño del filtro en 1 y se fija el factor de aprendizaje de los pesos en 10. Según esta última modificación, la red

neuronal aprende 10 veces más rápido en esta capa que en el resto de las capas de la red neuronal.

```
newConv = convolution2dLayer(1, numClasses, "Name", "new_conv", "WeightLearnRateFactor", 10);
```

Figura 38. Modificaciones realizadas en la capa convolucional

Fuente: Elaboración propia

Por otra parte, la capa “prob”, que convierte las salidas de la última capa convolucional en probabilidades debe cambiarse porque se pasa a tener 2 clases de salida. Por último, la capa final de clasificación se cambia, ya que estaba adaptada al número de clases original, que era igual a 100 y ahora es igual a 2. En caso de no realizar este cambio, las predicciones realizadas por la red serían incoherentes.

Para mejorar el entrenamiento y que la red neuronal aprenda de más información diferente a partir de los datos de entrada, se usa `imageDataAugmenter`. A pesar de contar con algo más de 200 imágenes en total, este comando es útil porque usando diferentes transformaciones que cambian el dataset, hace que la red aprenda nuevos detalles relevantes de las imágenes de entrenamiento. Esto es especialmente útil si no se cuenta con suficiente volumen de imágenes para entrenar la red neuronal.

En este caso, se genera una serie de transformaciones aleatorias a las imágenes para que la red neuronal capte nueva información que no está presente en el dataset original. Concretamente, se realiza un reflejo con respecto al eje X junto con traslaciones a lo largo de los ejes X e Y. De esta forma, se dificulta la tarea a la red neuronal porque el plumero de la Pampa no se encuentra siempre en el mismo lugar en todas las imágenes del dataset.

```
imageAugmenter = imageDataAugmenter(...
    'RandRotation', [-30 30], ... %Utilizo una rotación aleatoria
    'RandXTranslation', [-20 20], ... %Utilizo una traslación aleatoria en horizontal
    'RandYTranslation', [-20 20], ... %Utilizo una traslación aleatoria en vertical
    'RandXReflection', true ... %Utilizo un proceso de reflexión aleatorio
);
```

Figura 39. Captura de las transformaciones realizadas a las imágenes

Fuente: Elaboración propia

Es importante recordar que el entrenamiento es supervisado. Esto se debe a que la red toma los nombres de las carpetas como referencia para las clases, es decir, son imágenes etiquetadas. No obstante, este tipo de entrenamiento presenta varias limitaciones. En primer lugar, no es capaz de clasificar imágenes por sí mismo. Además, requiere de más tiempo de entrenamiento que otros tipos de redes y depende directamente de la intervención de un ser humano en el etiquetado de los datos de entrada, por lo cual es más susceptible de fallar en

su resultado. Para nuestro caso, dado que se deben clasificar únicamente 2 tipos de imágenes, es una buena solución.

Los resultados varían dependiendo del porcentaje de imágenes de entrenamiento y de imágenes de prueba. Además, la selección de las imágenes tiene un claro componente aleatorio, al ser así especificado en el código. Como se ha explicado anteriormente, la cortadería selloana crece indistintamente tanto en terrenos urbanos como rurales de todo tipo. Esto queda representado en las imágenes escogidas. No obstante, dicha característica aumenta la complejidad a la hora de realizar la clasificación final, ya que el plumero puede ser confundido con más elementos de las fotografías, como otros tipos de vegetación en el caso de imágenes de tipo rural. Es importante recalcar que, tanto en la red neuronal como en la máquina de vectores de soporte, la precisión ideal a obtener como resultado es de 100%, independientemente de la complejidad del dataset, de la variedad de imágenes o del tipo de reconocimiento que se desea implementar.

Finalmente, se modifican las opciones del entrenamiento de la red neuronal, a través de “Adam”. Este algoritmo es utilizado habitualmente para el entrenamiento de redes neuronales, mediante el cual se va modificando adaptativamente la tasa de aprendizaje, que indica si la red neuronal está aprendiendo o no. Para su funcionamiento, se calculan los momentos del gradiente, que ofrecen información para conocer cómo se comportan, y a su vez indican si se deben subir o bajar los pesos de la red para que el error del modelo baje.

El primer momento, denominado m_t , estima la tendencia del gradiente, mientras que el segundo momento, de nombre v_t , evita grandes cambios.

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot (g_t)^2$$

Los parámetros β_1 y β_2 controlan cuánto cae el primer y el segundo momento respectivamente. Ambos tienen valores cercanos a la unidad, gracias a lo cual se consideran más gradientes anteriores.

Mediante el cálculo de estos momentos, se ajustan los pesos de la red de forma adaptativa y bajan las oscilaciones durante el entrenamiento, luego el modelo gana en estabilidad y se actualizan sus parámetros. Por otra parte, la red neuronal también gana en velocidad porque converge más rápido, al corregir los sesgos y se vuelve más estable.

Al ejecutar el programa de la red neuronal, se obtiene la representación de la evolución del entrenamiento de la red, con diferentes apartados. En primer lugar, se extraen las gráficas de la precisión y de la pérdida. Además, se obtienen los valores analíticos del entrenamiento,

junto con el tiempo transcurrido durante el entrenamiento o el punto del proceso en el que se encuentre.

Por otra parte, el valor de “Validation Frequency” determina cada cuanto tiempo se evalúa la validación del modelo. El ciclo de entrenamiento se divide en 30 períodos o iteraciones. En cada uno de esos períodos, se emplean todo el conjunto de datos una vez, porque el parámetro “Iterations per epoch” es igual a 1. El valor de “Validation accuracy” es igual al porcentaje de precisión, es decir, el resultado que se desea saber.

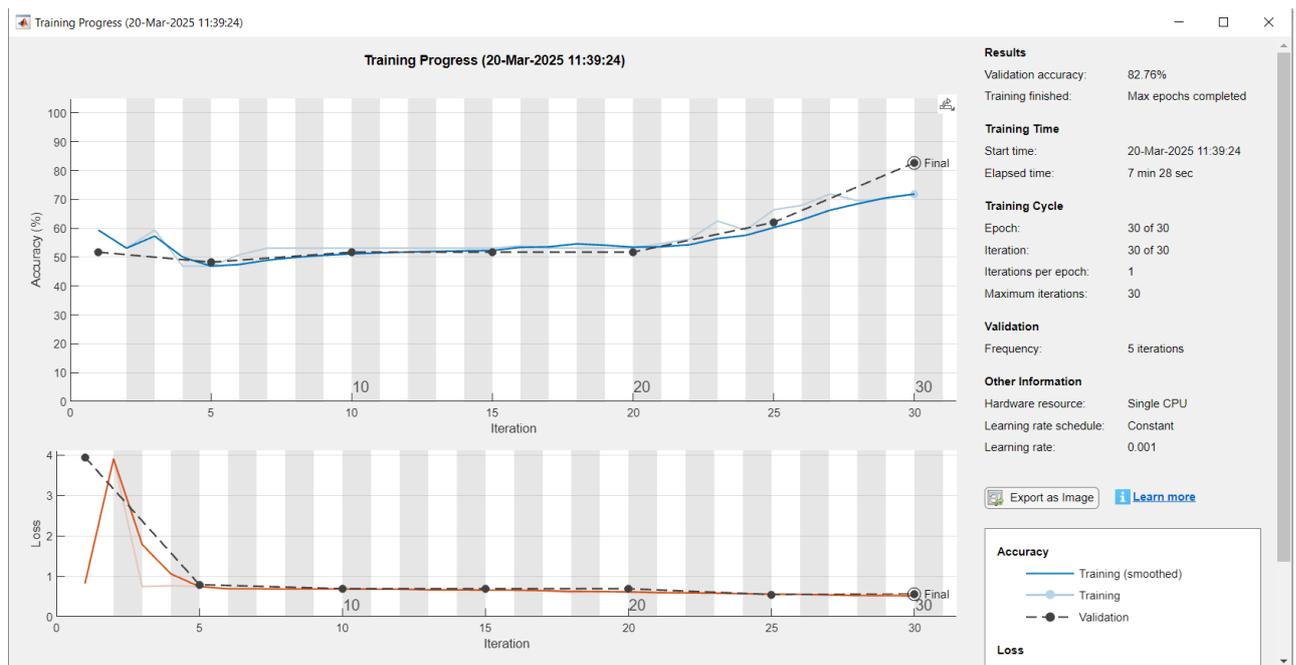


Figura 40. Captura de las gráficas del proceso de entrenamiento de la red neuronal

Fuente: Elaboración propia

Por lo que respecta a la función de pérdida, es una forma de representación de la diferencia entre el valor previsto y el valor real. El máximo que se aprecia en el inicio de esta gráfica corresponde a las primeras iteraciones, donde se produce un ajuste de los pesos de las neuronas de la red. Después, va bajando de una manera constante. Se puede determinar que el modelo es correcto porque dicha representación disminuye gradualmente y se acaba estabilizando. Además, esta gráfica también confirma que no existe sobreajuste, ya que el hecho de que no suba el valor de la validación en la gráfica de la pérdida revela una generalización correcta si se usa el resto de las imágenes.

Por otra parte, se puede extraer como conclusión que el porcentaje de adecuación de la validación es aceptable valorando la complejidad de los datos, ya que a pesar de que el plumero de la Pampa puede ser reconocible de una manera sencilla en imágenes aéreas,

cada imagen tiene elementos del entorno que entorpecen el reconocimiento, como otros tipos de vegetación. Tanto es así que, como ya se ha detallado, entre las imágenes de entrenamiento que no son de plumero, existen también bosques, praderas, cultivos, etc, específicamente seleccionados para complicar la tarea de la red neuronal debido a su posible parecido, teniendo en cuenta que se emplean únicamente imágenes satelitales, y así mejorar su capacidad de no devolver predicciones incorrectas.

Para una representación más clara y visual, se muestra una serie de imágenes aleatorias del dataset como resultado final, con la categoría teórica (“Categoría real”), la cual corresponde con el nombre de la carpeta, y con la categoría predicha por la red neuronal (“Categoría predicha”), con el fin de evaluar directamente el rendimiento de la red en las imágenes.

De esta forma, se obtienen por una parte las predicciones en las que la red neuronal utiliza imágenes sin cortadería selloana.



Figura 41. Predicción realizada por la red de una parcela sin plumero

Fuente: Elaboración propia

A su vez, la red neuronal devuelve también imágenes que poseen plumero de la Pampa de manera teórica, junto con su categoría predicha.



Figura 42. Predicción realizada por la red de una parcela con plumero

Fuente: Elaboración propia

Con el objetivo de comprobar la robustez y la eficacia de la red neuronal, se pueden modificar los parámetros de entrenamiento y ver los diferentes resultados. Variando los porcentajes de imágenes usadas en entrenamiento y verificación, se puede juzgar más fácilmente el rendimiento de la red neuronal SqueezeNet.

Para empezar, se introduce un valor bajo en las imágenes de entrenamiento, usando sólo el 60%. Por otra parte, se emplea el 25% de las imágenes para validar y el 15% para realizar la prueba de la red.

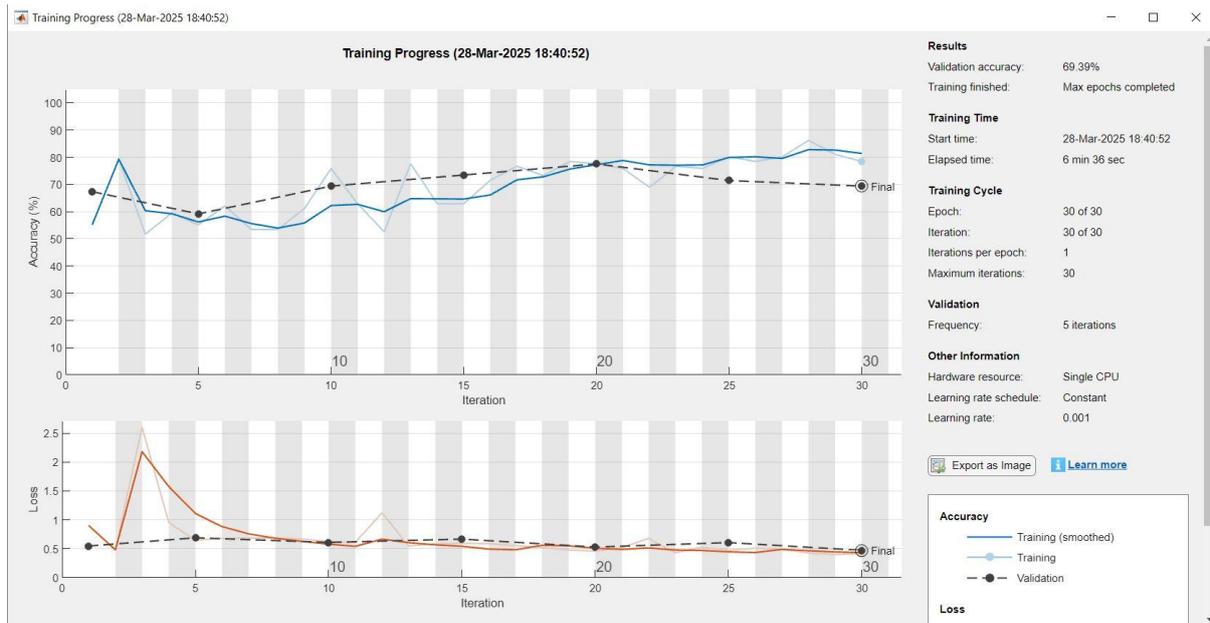


Figura 43. Captura de las gráficas de rendimiento y de pérdida con la anterior configuración

Fuente: Elaboración propia

De la misma manera, se obtienen diferentes imágenes del dataset, junto con la etiqueta teórica y la etiqueta predicha.

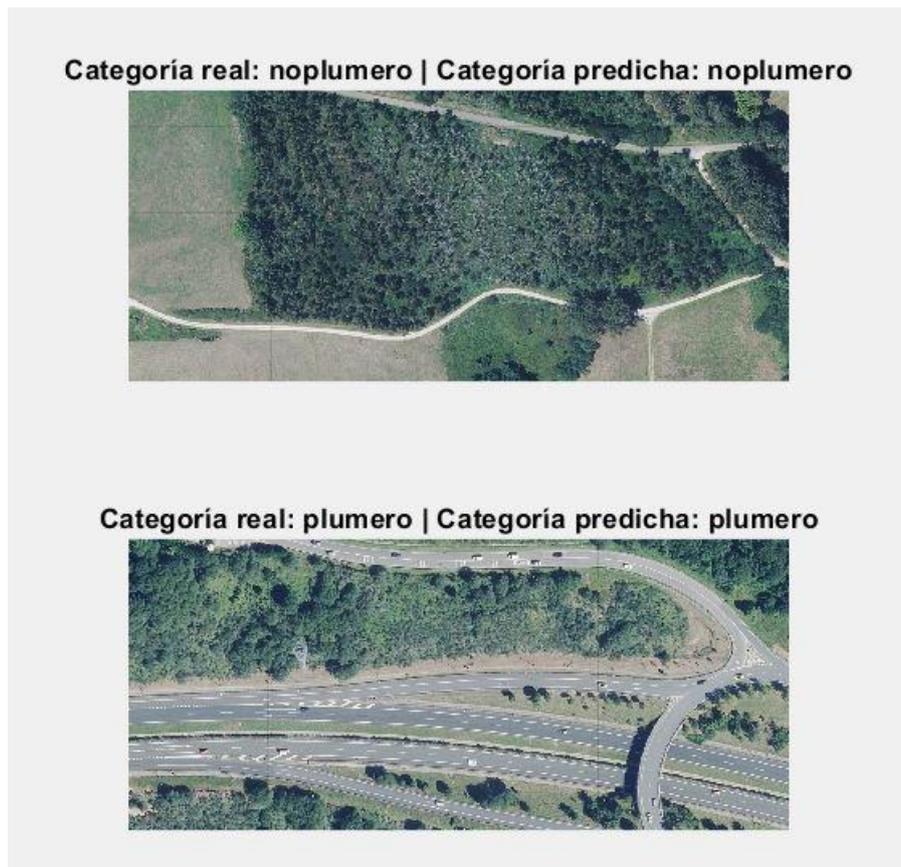


Figura 44. Captura de la predicción de 2 imágenes con la configuración anterior

Fuente: Elaboración propia

Después, se aumenta en cierta medida el porcentaje de imágenes destinadas al entrenamiento, usando el 70%, mientras que se emplea el 20% de las imágenes para validar y el 10% para el proceso de prueba.

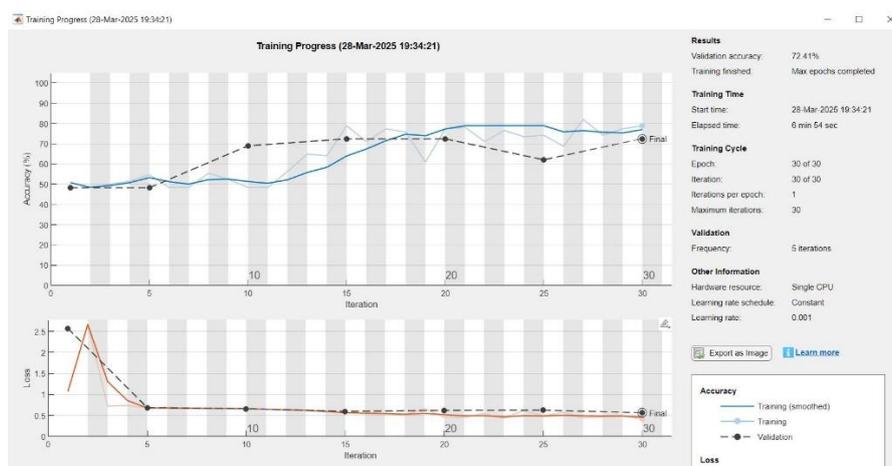


Figura 45. Imagen de las gráficas de rendimiento y pérdida usando esta configuración

Fuente: Elaboración propia

Tras obtener las gráficas correspondientes a la precisión y a la pérdida, el programa me devuelve una serie de fotografías con sus etiquetas correspondientes.



Figura 46. Representación de imágenes predichas por la red neuronal con esta configuración

Fuente: Elaboración propia

Para concluir, se realiza el último entrenamiento de SqueezeNet, según el cual, se usan el $\frac{3}{4}$ partes del dataset para el entrenamiento, mientras que se utiliza el 15% de las imágenes para validar y el 10% para probar la red neuronal.

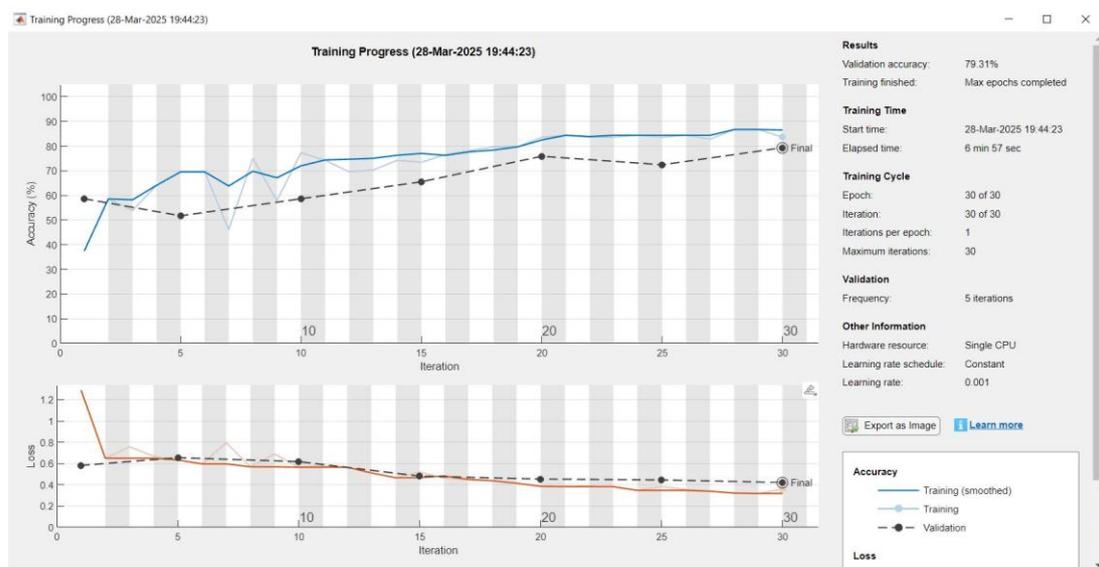


Figura 47. Gráficas de la precisión y de la pérdida de SqueezeNet con esta configuración

Fuente: Elaboración propia

Como en los anteriores entrenamientos, se devuelve al término del programa un conjunto aleatorio de las imágenes con su etiqueta teórica, que corresponde con el nombre de su carpeta y junto con la categoría predicha por la red neuronal entrenada.

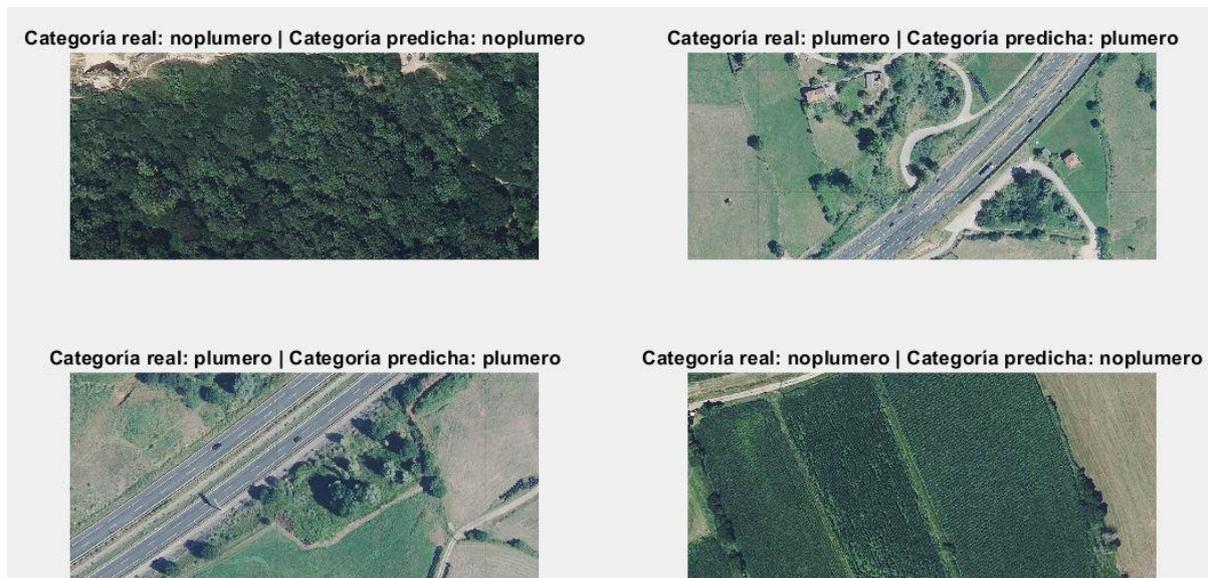


Figura 48. Diferentes predicciones realizadas por la red en esta simulación concreta

Fuente: Elaboración propia

7.3. ENTRENAMIENTO DE ALEXNET

En comparación con el código de la red neuronal de SqueezeNet, AlexNet cuenta con pocas diferencias. Esto se debe principalmente a que ambas redes neuronales deben afrontar el mismo problema relativo al procesamiento y reconocimiento de imágenes.

Sus principales diferencias se encuentran en la arquitectura. SqueezeNet basa gran parte de su funcionamiento, como se ha mostrado anteriormente, en módulos Fire para el tratamiento y la compresión de la información. No obstante, AlexNet tiene una estructura mucho más simple, utilizando capas convolucionales y capas completamente conectadas o en inglés “fully connected (fc) layers”. Esto modifica el código porque, al contar con 2 posibles soluciones (plumero o no plumero), se deben de ajustar las probabilidades de la salida.

```
%Modifico la capa completamente conectada para que coincida con el número
%de clases, que son 2: plumero y no plumero
newFc = fullyConnectedLayer(numClasses, "Name", "new_fc", "WeightLearnRateFactor", 2);
lgraph = replaceLayer(lgraph, "fc8", newFc);
```

Figura 49. Captura de los cambios realizados en el código

Fuente: Elaboración propia

En caso contrario, la red neuronal tendría un resultado incoherente y no sería capaz de detectar las imágenes con cortadería selloana, sino que ofrecería otras predicciones erróneas. Una vez se ejecuta el programa, se obtiene la gráfica que miden la precisión de la red neuronal y la representación de la función de pérdida.

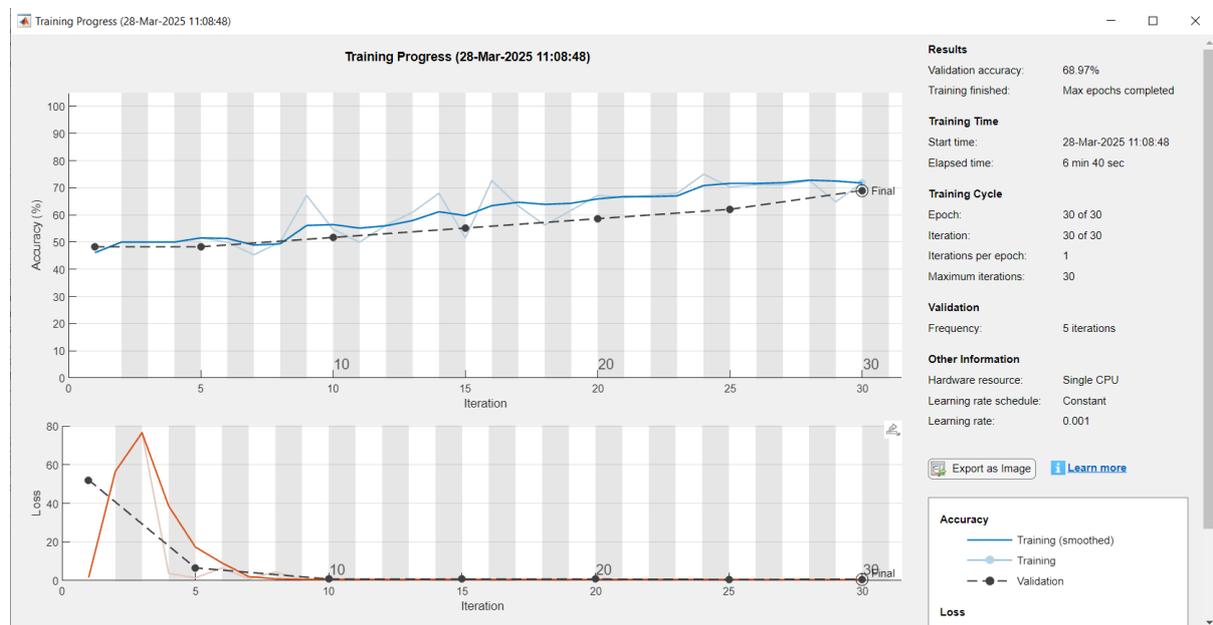


Figura 50. Captura de las gráficas de precisión y de pérdida de la red neuronal de AlexNet

Fuente: Elaboración propia

Modificando los valores de los porcentajes de imágenes usadas para el entrenamiento, para la validación y para la prueba, se obtienen diferentes resultados en cuanto a la precisión. Al igual que en SqueezeNet, se muestran al final una serie de imágenes existentes en el dataset, con su etiqueta teórica y su etiqueta predicha, para tener una salida más ilustrativa para el usuario.

Primeramente, se genera una simulación utilizando el 60% de imágenes para entrenar, el 25% de ellas para validación y el 15% como prueba.

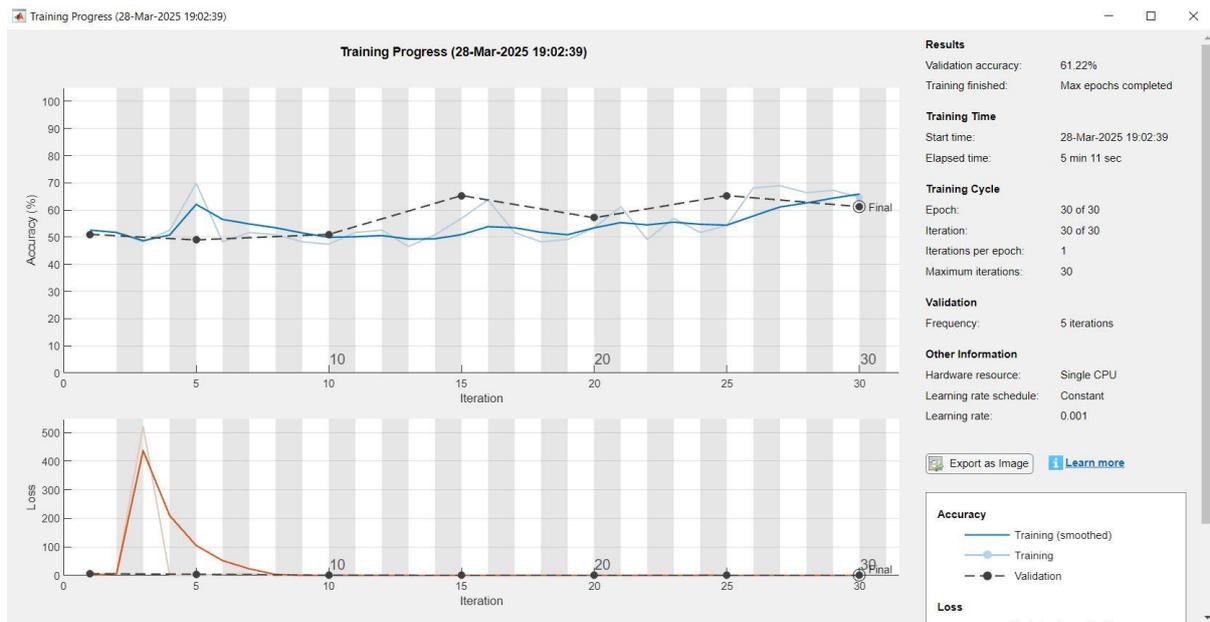


Figura 51. Representación de las gráficas usando la anterior configuración de imágenes

Fuente: Elaboración propia

Al igual que como se ha mostrado en SqueezeNet, se devuelve una serie de imágenes después de obtener las gráficas de entrenamiento y de pérdida, donde se muestran las categorías teórica y predicha por la red.

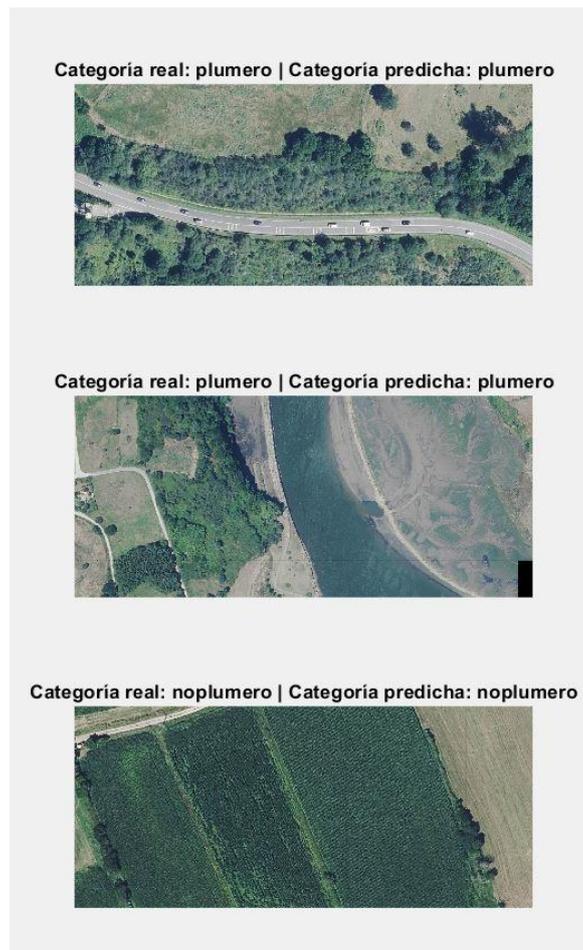


Figura 52. Captura de algunas imágenes detectadas con la anterior configuración

Fuente: Elaboración propia

Posteriormente, se cambian los porcentajes y uso el 65% de las imágenes para el entrenamiento de la red, el 20% para realizar la validación y las fotografías restantes se usan para la prueba del modelo.

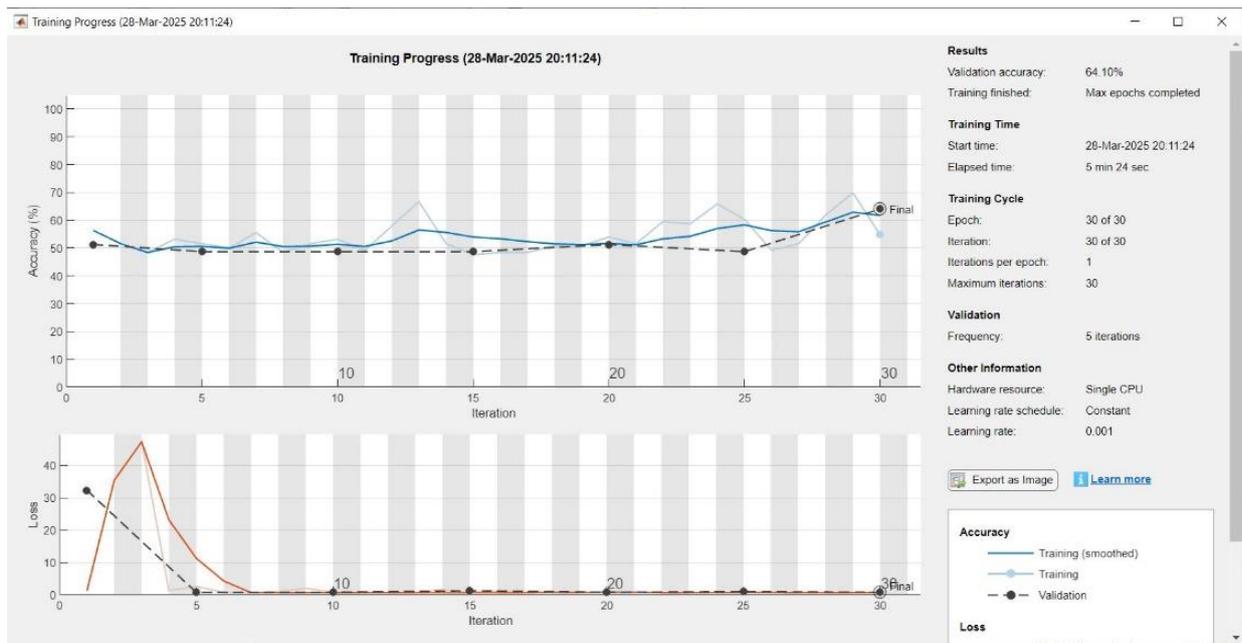


Figura 53. Imagen de las gráficas de precisión y pérdida usando la anterior configuración

Fuente: Elaboración propia

Con esta nueva configuración, se muestran por pantalla imágenes predichas por la red neuronal al finalizar su entrenamiento.



Figura 54. Representación de algunas de las predicciones realizadas

Fuente: Elaboración propia

Para continuar, se aumenta el valor de las imágenes de entrenamiento al 70%, manteniendo la misma proporción para las imágenes de validación. Por consiguiente, la red usará el 10% de las imágenes para la prueba.

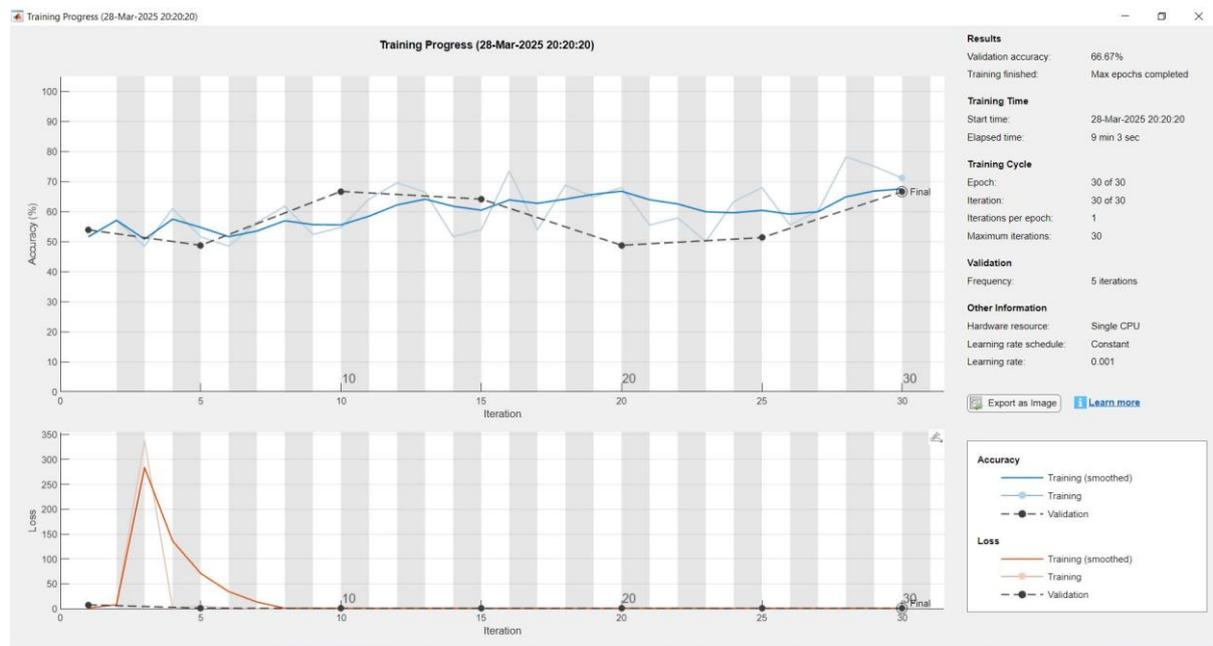


Figura 55. Gráficas de precisión y pérdida usando la anterior configuración

Fuente: Elaboración propia

En este caso, se muestra un ejemplo correcto de cada una de las categorías teórica y predicha.



Figura 56. Predicciones de la red con esta configuración, tanto de plumero como sin plumero

Fuente: Elaboración propia

Por último, se realiza una ejecución de AlexNet usando el 75% de las imágenes para el entrenamiento, el 15% para la validación y el resto de ellas para la prueba de la red neuronal.

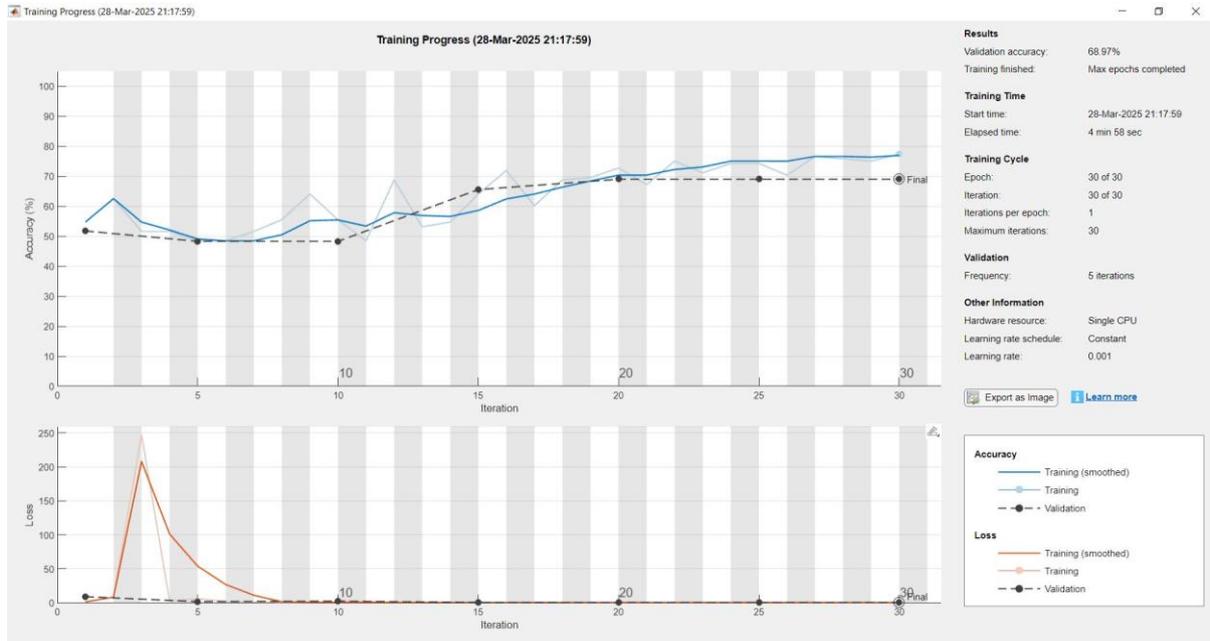


Figura 57. Representación de las gráficas de precisión y pérdida de la red neuronal con esta configuración

Fuente: Elaboración propia

En este último entrenamiento, se obtiene una serie de imágenes con sus respectivas predicciones.

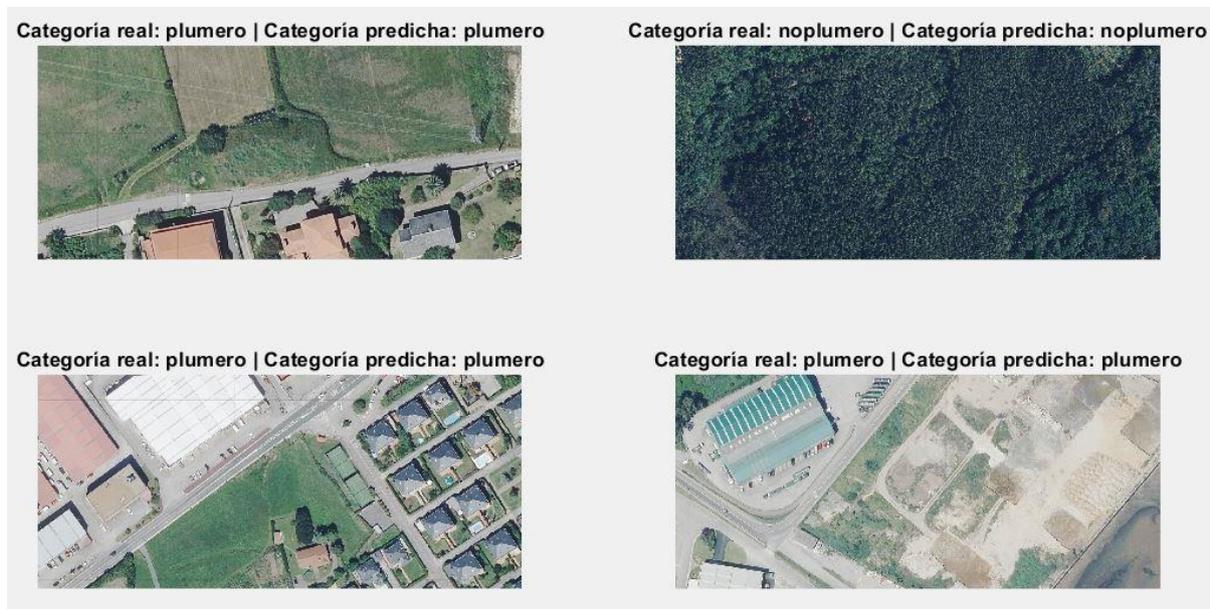


Figura 58. Representación de 4 imágenes de la base de datos con sus etiquetas teórica y predicha

Fuente: Elaboración propia

Los resultados obtenidos en las gráficas de entrenamiento tienen coherencia, siendo 68'97% el mejor rendimiento que se obtiene. Esto es congruente ya que, al ser AlexNet una red neuronal de menor calidad para el reconocimiento de imágenes que SqueezeNet, su rendimiento debe de ser peor en la mayoría de las ejecuciones que se hagan, aunque se puede afirmar que los resultados son aceptables. La antigüedad de AlexNet también dificulta a la red a la hora de generalizar un conjunto de datos complejo. Sin embargo, la gráfica de la pérdida es prácticamente perfecta. Esto es un claro indicador de que la red aprende las características de las imágenes del plumero de manera óptima, dentro de las posibilidades que ofrece.

7.4. ENTRENAMIENTO DE RESNET50

De una manera similar a otras redes neuronales, se implementa un programa de detección de cortadería selloana en imágenes satelitales mediante el uso de la red neuronal ResNet-50.

El comienzo del código es idéntico al código de otras redes neuronales porque el tratamiento de los datos de entrada no se modifica. Esto implica que, nuevamente, las imágenes se obtienen al descomprimir el archivo .zip y las etiquetas que conforman las posibles salidas son los nombres de las carpetas, es decir, "plumero" y "noplumero".

Por consiguiente, es necesario modificar ciertas capas de la red neuronal porque originalmente, ResNet-50 no está adaptada para distinguir únicamente 2 clases. De igual manera, también es obligatorio modificar el dimensionado de los datos de entrada a la red neuronal, que en este caso son [224 224 3]. Esto difiere ligeramente de otras estructuras como AlexNet, cuyas dimensiones para los datos de entrada son [227 227 3]. Estas diferencias se deben al diseño de la red, siendo en la actualidad el formato [224 224 3] más habitual, en parte por haberse seguido ese diseño en diferentes datasets de entrenamiento en la actualidad como ImageNet.

A pesar de ser otra red neuronal alternativa a las explicadas, continúa existiendo la posibilidad de que el dataset original no sea suficientemente bueno para el entrenamiento. Por ello, empleo también almacenes aumentados.

Finalmente, se modifican los parámetros del entrenamiento de manera idéntica a los otros programas y se devuelve por pantalla una serie de imágenes para comprobar su etiqueta teórica y su etiqueta propuesta, además de representar gráficamente el proceso del entrenamiento.

Al igual que en los otros programas, se hacen sendas ejecuciones modificando las propiedades del entrenamiento, realizando en todas ellas una selección aleatoria de las imágenes dentro del dataset pero modificando las cantidades destinadas al entrenamiento y a la verificación.

En primer lugar, se entrena la red neuronal residual usando la mitad del dataset para el proceso del entrenamiento, mientras que empleo el 30% de las imágenes para validar la red neuronal.

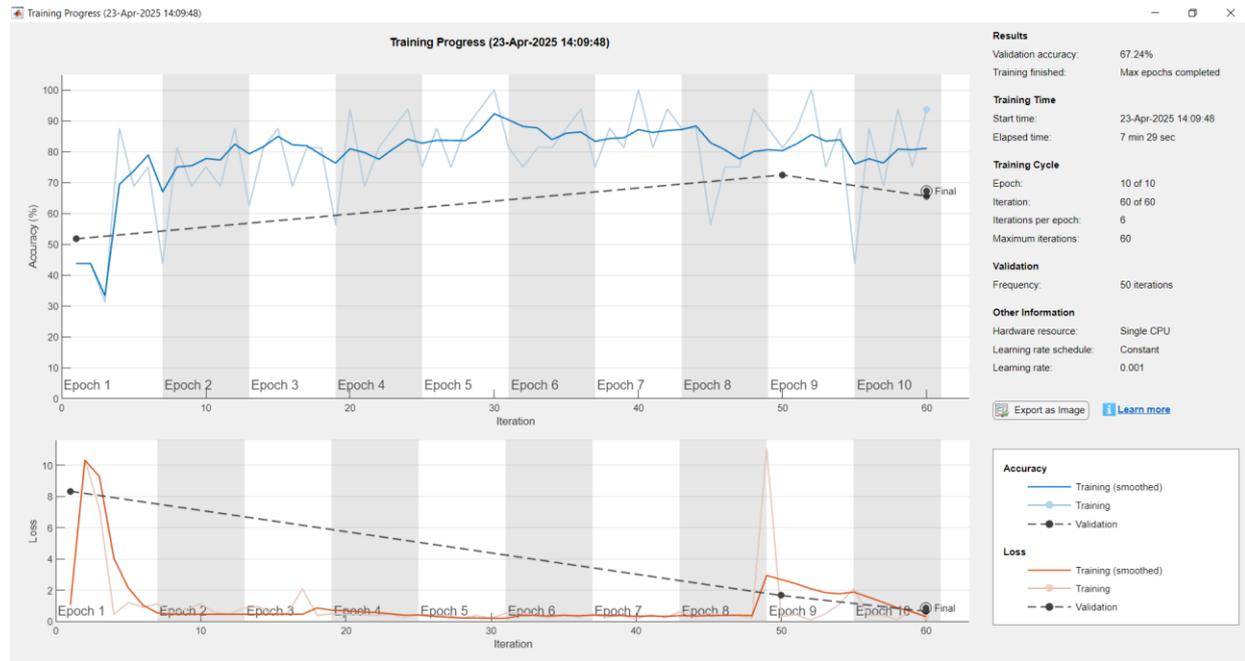


Figura 59. Gráficas resultantes del anterior entrenamiento

Fuente: Elaboración propia

Nuevamente, el programa devuelve imágenes que utiliza la red neuronal en su proceso de prueba con sus etiquetas.



Figura 60. Predicciones realizadas por la red al término del entrenamiento previo

Fuente: Elaboración propia

Para ofrecer más información en la simulación y a diferencia de en otras, aparece la palabra “Epoch” en las gráficas del progreso de entrenamiento porque se realiza más de una iteración por época.

Posteriormente, se realiza un entrenamiento utilizando el 70% de las imágenes del dataset para entrenar la red, el 20% para realizar la validación del modelo, y el 10% restante para el proceso de prueba.

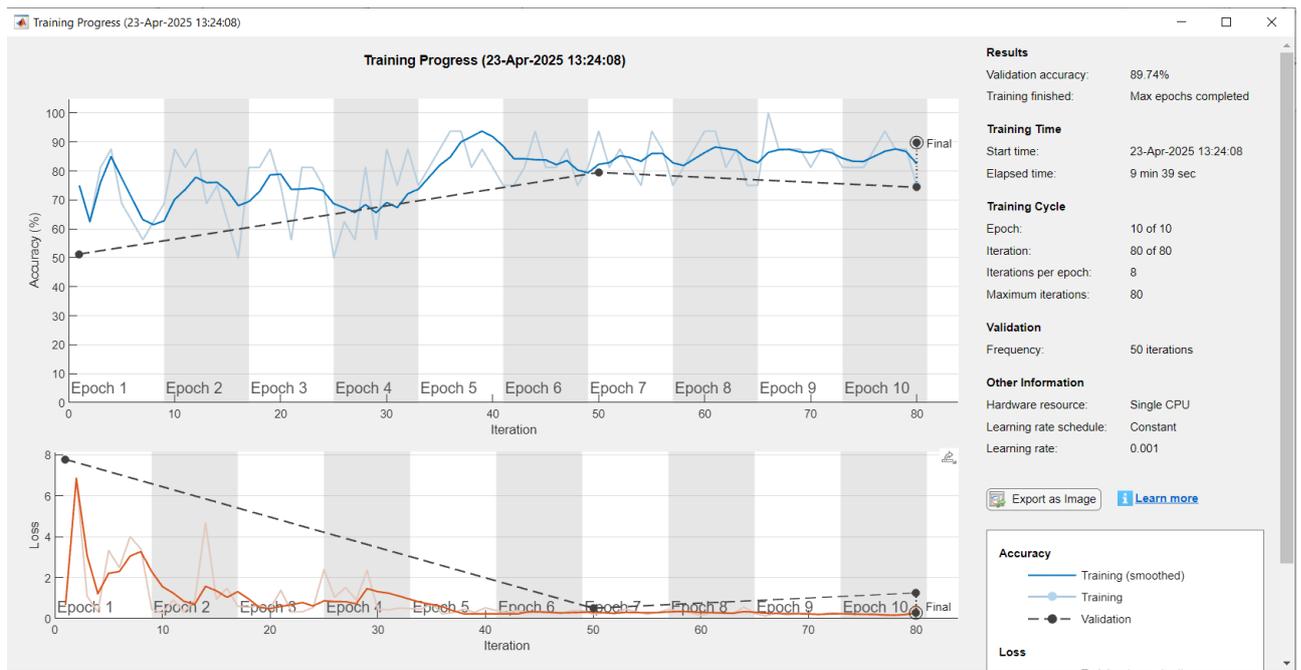


Figura 61. Captura de las gráficas de precisión y de pérdida usando la anterior configuración

Fuente: Elaboración propia

En este nuevo entrenamiento, se extrae un par de imágenes con cortadería selloana con las correspondientes categorías predichas. Una de ellas, en un ambiente rural y otra, en un ambiente urbano.



Figura 62. Pareja de imágenes predichas por la red neuronal

Fuente: Elaboración propia

Las gráficas obtenidas en el empleo de ResNet-50 muestran que el rendimiento de la red es elevado, así como su pérdida decae rápidamente. No obstante, también se puede apreciar una cierta inestabilidad a lo largo del proceso, que es provocada por la profundidad de dicha arquitectura. Al ser una red neuronal con una tecnología más avanzada que SqueezeNet y AlexNet, es la mejor opción explicada para poder detectar imágenes con plumero de la Pampa.

7.5. ENTRENAMIENTO DE SVM

Al igual que en los programas de las redes neuronales, en primer lugar, se descomprime el archivo .zip donde se encuentran las imágenes a analizar. A partir del archivo descomprimido, se etiquetan las imágenes gracias a los nombres de las carpetas que contenían el archivo comprimido. En este caso, “plumero” y “no plumero”. El porcentaje de imágenes empleadas en el entrenamiento y en el testeo será modificado posteriormente en el código.

```
%Descomprimo el archivo de imagenes
unzip('nuevasimagenes.zip','nuevasimagenes');
%Creo el almacén de datos
imds=imageDatastore('nuevasimagenes','IncludeSubfolders',true,'LabelSource','foldernames');
%Defino el porcentaje de entrenamiento. El porcentaje de verificación es el
%restante
[imdsTrain,imdsTest]=splitEachLabel(imds,0.8,'randomized');
```

Figura 63. Descompresión del archivo de imágenes y creación del almacén de datos

Fuente: Elaboración propia

Tras representar un número de imágenes aleatorias, se especifica que la capa de la cual se extraerán las características será la última capa fc1000. Como bien indica su nombre, cada una de las neuronas de esta capa está conectada a todas las neuronas de capa anterior. Al tener este tipo de conexión, la red neuronal puede aprender patrones mucho más complejos, incluso no lineales o generar nuevas características, partiendo de toda la información recopilada por las capas anteriores.

```
%Defino la capa de la cual se extraen las características. Esta capa se
%puede modificar para evaluar el comportamiento del SVM
layer="max_pooling2d_1";
%Cargo la red neuronal residual
net=resnet50;
```

Figura 64. Dos de las líneas más importantes de la red neuronal

Fuente: Elaboración propia

Se carga la red neuronal y se ajustan las dimensiones de las imágenes para el correcto funcionamiento de la red neuronal, mediante la creación de almacenes aumentados. Como se explicó anteriormente, los almacenes aumentados sirven para que la red neuronal aprenda nuevas características y así evitar un posible problema de “overfitting”. Para solucionarlo, albergarán nuevas imágenes modificadas, partiendo de las capturas originales. Estas modificaciones se pueden clasificar en transformaciones geométricas, capaces de cambiar el espacio y el diseño de las imágenes, y transformaciones fotométricas, utilizadas para alterar

los valores de los canales R, G y B. Estas modificaciones permiten que la red neuronal capte otros matices que no existían en las imágenes originales, por lo cual aprende nueva información. Gracias a esto, el modelo gana en solidez en el reconocimiento del plumero de la Pampa y se obtiene un mejor resultado. El uso de almacenes aumentados es especialmente útil cuando no se dispone de un dataset grande, o cuando las características de las imágenes de las que se dispone no difieren mucho unas de otras, como en este caso.

Posteriormente, se extraen las características que usará para clasificar las imágenes. Estas características servirán para medir la precisión del modelo, relacionando el parecido que existe entre ellas.

```
%Realizo la extracción de características de las imágenes
%¿Qué ha aprendido la red en esta capa usando las imágenes de entrenamiento?
featuresTrain = activations(net, augimdsTrain, layer, 'OutputAs', 'rows');
%¿Qué ha aprendido la red en esta capa usando las imágenes de verificación?
featuresTest = activations(net, augimdsTest, layer, 'OutputAs', 'rows');
```

Figura 65. Extracción de características usando ambos conjuntos y obtención de las etiquetas

Fuente: Elaboración propia

Se realiza el entrenamiento del SVM, mediante el empleo de `fitcecoc`. Esta función es usada para crear el clasificador y para entrenar un modelo multiclase en función de las características extraídas y las etiquetas. Gracias a `fitcecoc`, se reduce cualquier problema de clasificación de imágenes, independientemente del número de clases, en un problema binario. Gracias a este clasificador, se generan predicciones y para evaluar la calidad de éstas, se calcula la precisión.

La ejecución del programa de SVM devuelve como resultado esta precisión, que al igual que en la red neuronal, es el porcentaje de imágenes correctamente etiquetadas, tanto de “plumero” como de “no plumero”.

Para una mejor representación visual de los resultados obtenidos, se realiza una gráfica lineal, donde se relacionan el porcentaje de datos usado para entrenamiento en el eje horizontal con el resultado de la precisión de la máquina de vectores de soporte en porcentaje en el eje vertical.

Datos de entrenamiento (%)	Resultado
10	76%
20	73,55%
30	77,21%
40	78,45%
50	78,35%
60	83,33%
70	81,03%
80	87,18%

Figura 66. Tabla con los resultados obtenidos del SVM

Fuente: Elaboración propia

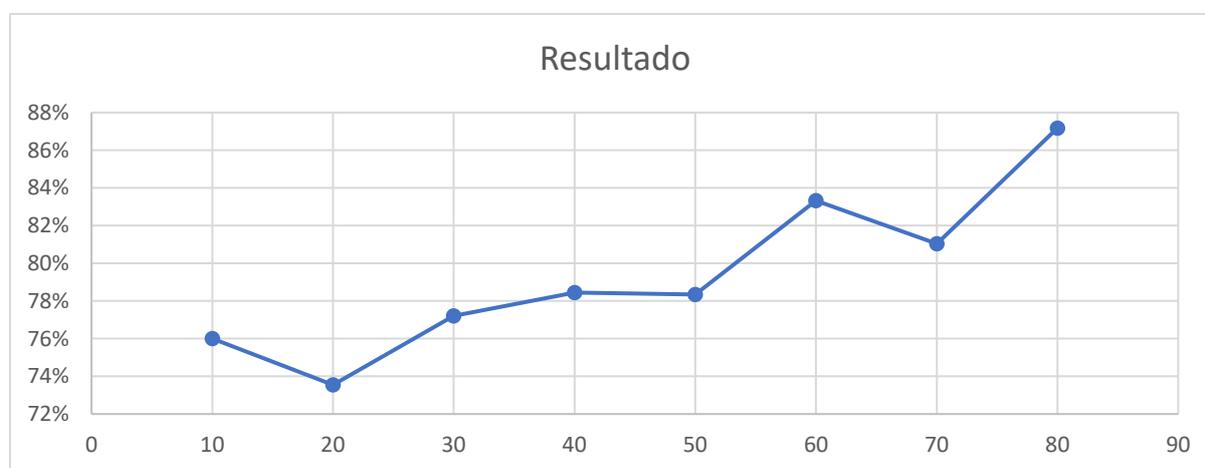


Figura 67. Representación lineal del rendimiento frente al porcentaje de datos de entrenamiento

Fuente: Elaboración propia

Con el fin de obtener un resultado más gráfico, que clarifique los resultados obtenidos en el programa de SVM, se obtiene una matriz de confusión. Esta representación se considera por sí mismo una métrica para poder evaluar el rendimiento de un modelo de clasificación. De una manera simple e intuitiva se engloban todos los resultados obtenidos, tanto las predicciones correctas como las predicciones incorrectas, recopilando todos los datos que ha usado el programa durante la fase de prueba.

Por ejemplo, en el supuesto de haber usado únicamente el 20% de los datos para el entrenamiento del programa, extrayendo las características de la capa fc1000, se obtenía la siguiente representación.

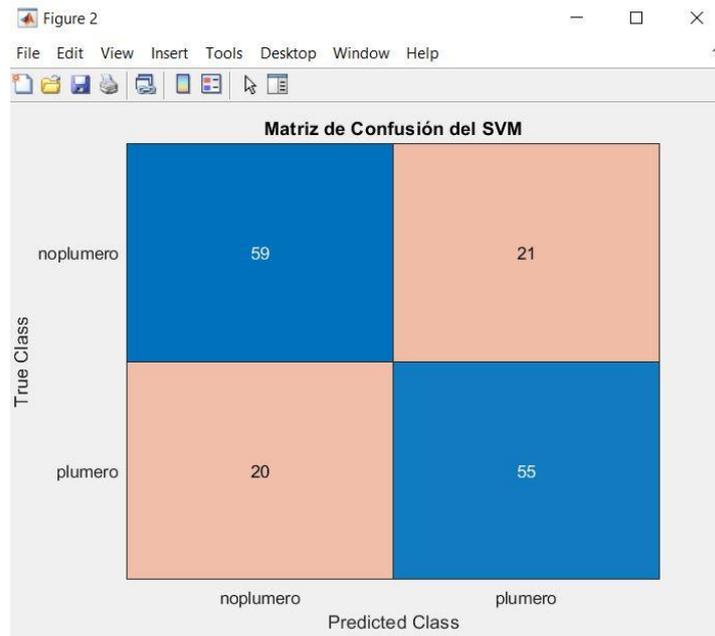


Figura 68. Matriz de confusión obtenida con la configuración anterior

Fuente: Elaboración propia

En el caso anterior, se obtuvo el peor resultado de la máquina, en el cual 114 imágenes fueron correctamente etiquetadas, mientras que 41 fotografías adquirieron la categoría contraria.

En otro ejemplo, cuando se entrenó la máquina usando el 80% de los datos del dataset para el entrenamiento, sacando las características nuevamente de la capa fc1000, se obtuvo otra figura.

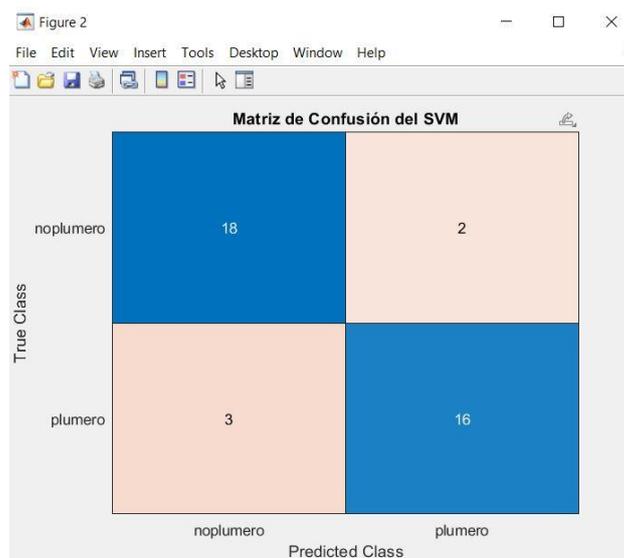


Figura 69. Captura de la carta de confusión usando la nueva configuración

Fuente: Elaboración propia

En este nuevo supuesto, al haber utilizado 4 de cada 5 imágenes para el entrenamiento, se disponen de menos datos para comprobar el correcto funcionamiento de la máquina de vectores de soporte. No obstante, el programa capta mucha más información y se obtienen una precisión superior al 87%, etiquetándose de manera incorrecta únicamente 5 fotografías de 39.

Estos resultados de la máquina de vectores de soporte podrían mejorar en caso de disponer de un dataset mayor, ya que puede darse la circunstancia de que el programa no consiga adquirir nueva información independientemente del número de ejecuciones, al haber analizado todas las imágenes que posee, especificando su uso tanto para entrenamiento como para prueba. Sin embargo, los resultados de la precisión también muestran que el conjunto de datos es tan complejo como variado y, como ya se explicó con anterioridad, el hecho de escoger tanto imágenes que cuenten simultáneamente con ejemplares de cortadería selloana y otras vegetaciones parecidas, como escoger imágenes con gamas cromáticas similares a las encontradas en las fotografías con plumero en la sección “no plumero”, afecta directamente al rendimiento del SVM, ya que el programa comete errores de una manera más fácil.

Además, también se puede apreciar cierta incoherencia porque cuanto mayor es el porcentaje de datos utilizados para el entrenamiento, mayor debería ser la precisión obtenida. No obstante, el motivo de estos desajustes es la selección aleatoria de imágenes que realiza el programa para los procesos de entrenamiento y de validación.

7.6. CONCLUSIONES

A la vista de los resultados obtenidos en los entrenamientos, se pueden extraer diferentes deducciones. En primer lugar, se puede concluir que el desarrollo de las redes neuronales junto con las tecnologías que utilizan, como SqueezeNet y el módulo “Fire” o los bloques residuales en ResNet, persigue la obtención del mejor rendimiento posible independientemente del conjunto de datos con el cual se trabaje. Por otra parte, si los datos de entrada se tratan correctamente sin sufrir grandes modificaciones y se extraen características de alto nivel, una mayor profundidad en la red neuronal no provoca obligatoriamente una bajada del rendimiento. Es más, redes neuronales con arquitecturas tan sencillas como AlexNet son susceptibles de quedar en desuso, más aún si las imágenes a analizar son complejas, como en este caso. Por estos motivos, junto con los resultados obtenidos, se considera recomendable realizar la fase principal de inferencia mediante ResNet50, aunque se probará también AlexNet para comparar las diferencias.

8. FASE DE INFERENCIA

8.1. INTRODUCCIÓN

Como se ha mencionado anteriormente, el proceso de inferencia es inmediatamente posterior al proceso de entrenando de la red neuronal, siendo el tiempo de inferencia una variable a optimizar. En este nuevo proceso, la red neuronal ya entrenada trabaja con datos que no ha usado previamente. Podría considerarse erróneamente que el proceso a realizar es una inferencia por lotes, ya que no se requiere una respuesta inmediata, pero dado que el programa únicamente trata una imagen por ejecución, no se considera en esa categoría.

8.2. INFERENCIA MEDIANTE ALEXNET Y SVM

Para el proceso de inferencia usando la red AlexNet previamente entrenada para el reconocimiento, cargo tanto la imagen como la red entrenada y el SVM entrenado. La imagen es adaptada a las dimensiones necesarias por la red neuronal para que pueda ser analizada. Después, extrayendo las características y usando el clasificador, se realiza la predicción de la etiqueta, es decir, plumero o no plumero.

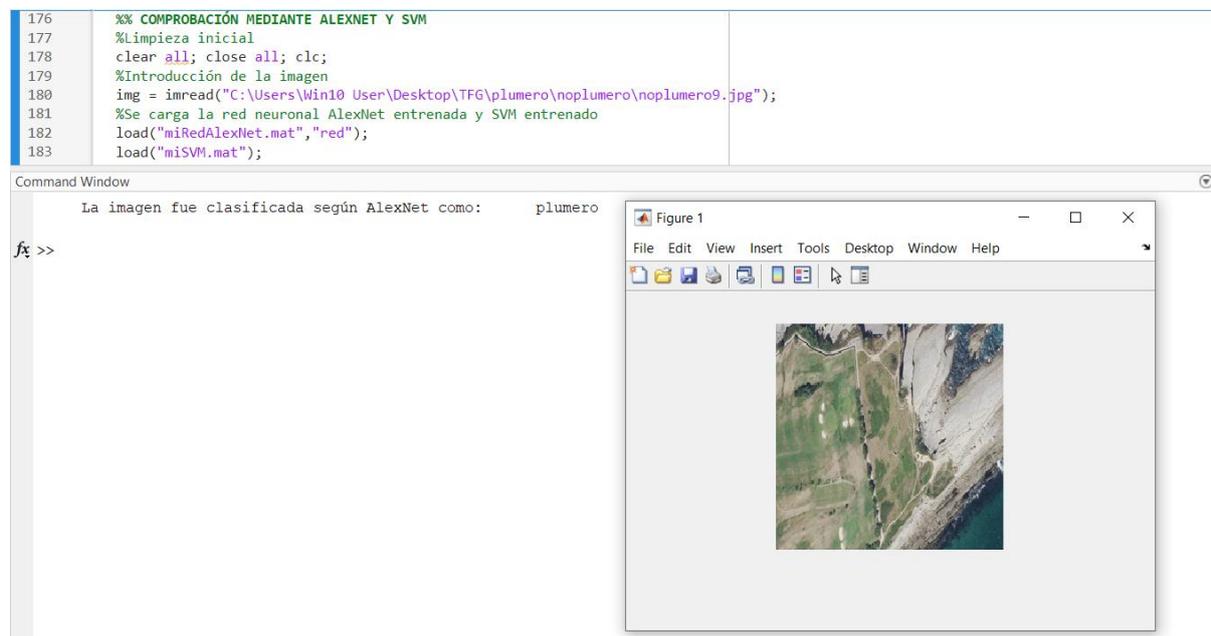


Figura 70. Ejemplo de predicción incorrecta usando AlexNet y SVM

Fuente: Elaboración propia

Este procedimiento se puede repetir con otras imágenes sin plumero de la Pampa, obteniendo el mismo resultado.

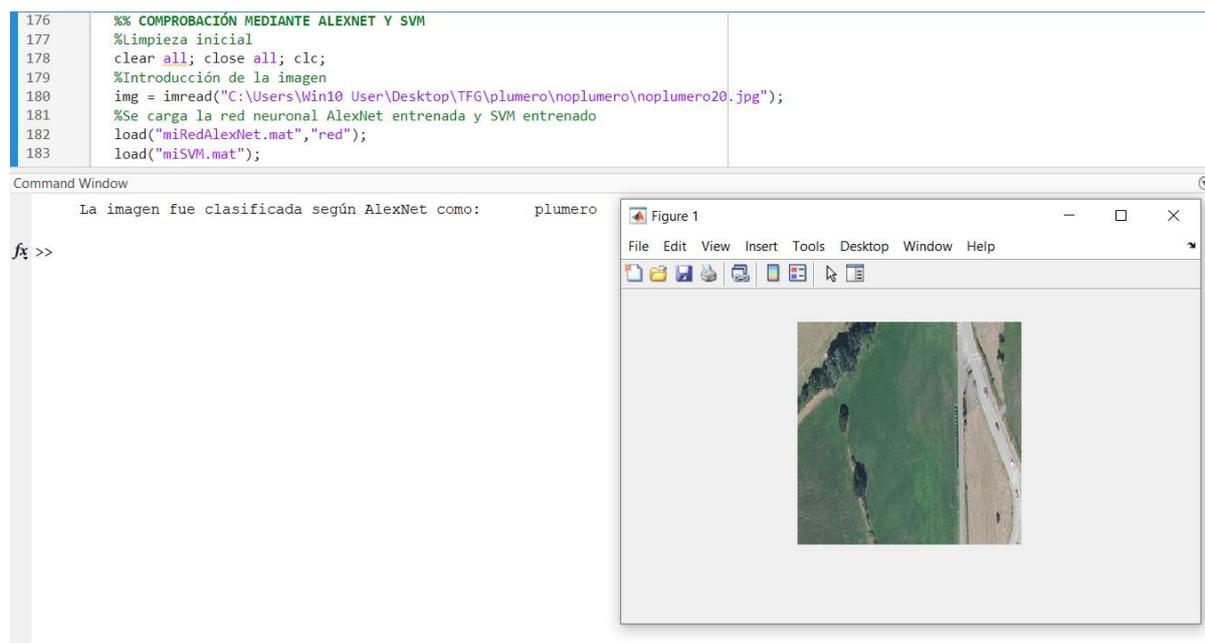


Figura 71. Ejemplo de predicción incorrecta empleando AlexNet y SVM

Fuente: Elaboración propia

La obtención de predicciones erróneas mediante el empleo de SVM y AlexNet en la fase de inferencia se debe a la simplicidad de la red neuronal anterior, que contrasta con la dificultad de los matices en las diferentes imágenes del dataset. A su vez, esta problemática ofrece una mejor visión sobre los impedimentos que tienen redes neuronales de arquitecturas simples y con cierta antigüedad a la hora de utilizar conjuntos de datos complejos. Por consiguiente, se extrae la conclusión de la necesidad de redes neuronales con una mayor profundidad para trabajar con imágenes que cuenten con este tipo de vegetación o con plantas similares. De esta forma, el surgimiento de este problema también sirve para comprender que el desarrollo de nuevas estructuras está enfocado en la resolución de problemas con mayor complejidad, acorde a las necesidades contemporáneas.

8.3. INFERENCIA MEDIANTE RESNET

Por todos estos motivos, en caso de pretender solucionar problemas de detección similares al expuesto, es reseñable la necesidad de utilizar redes neuronales con características similares a ResNet-50 u otras redes residuales.

No obstante, la mejor opción para el proceso de inferencia es hacer uso de ResNet50. Una vez se obtiene un buen rendimiento en ella, se puede pasar al proceso de inferencia. En esta fase del desarrollo y prueba de la red, se puede analizar diferentes cosas, como testear el modelo con cualquier imagen que se desee o probar con una misma imagen extraída a

diferentes alturas. Para ello, se utiliza SigPac, que como se ha explicado con anterioridad, nos permite extraer fotografías de una determinada parcela de terreno modificando el valor del “zoom” gracias a la barra que se encuentra en la izquierda.

Con este objetivo, se debe guardar la red neuronal ResNet-50 ya entrenada en formato “.mat” y dispuesta a catalogar nuevas imágenes en función de la presencia o de la ausencia de cortadería selloana. Junto con la red, se salvan también las clases que es capaz de detectar. En un nuevo script, dedicado exclusivamente al proceso de inferencia, se carga dicha red y una imagen del archivo de datos. Esta imagen debe acondicionarse a la red neuronal residual, por lo que hay que modificar su anchura y longitud, ambos valores a 224 píxeles.

```
imgResized = imresize(img, [224 224]);
```

Figura 72. Línea del código dedicada a la adaptación de los datos

Fuente: Elaboración propia

Luego, mediante un almacén aumentado, se crean los datos a pronosticar y se realiza la predicción. Se devuelve finalmente por pantalla la imagen introducida con los porcentajes predichos en cada una de las clases, plumero y no plumero.

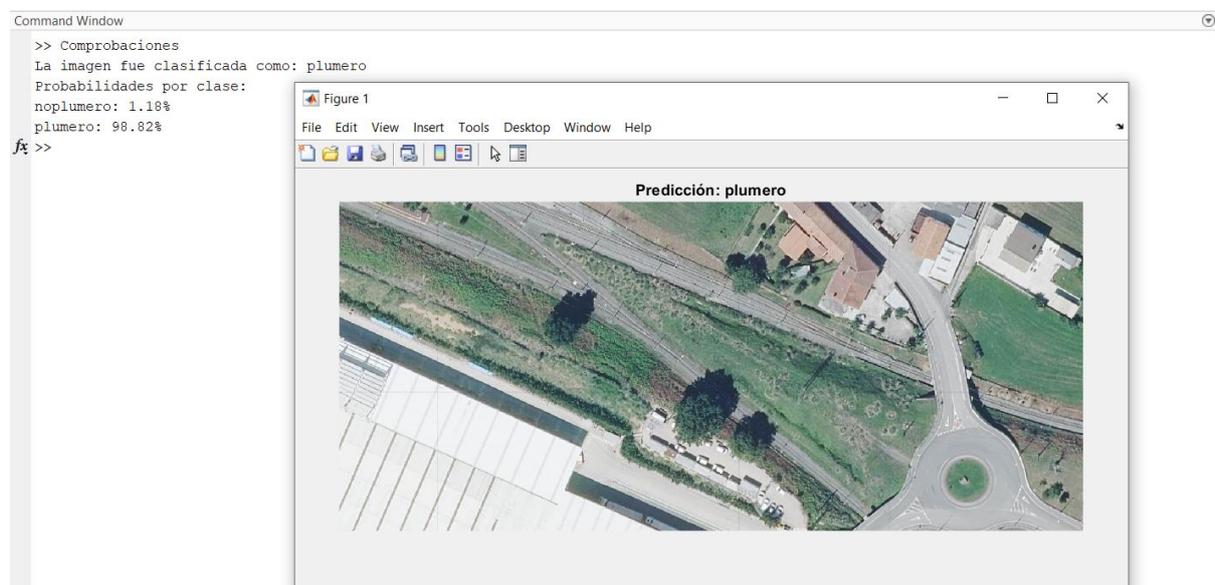


Figura 73. Ejemplo de una predicción realizada junto con los porcentajes resultantes

Fuente: Elaboración propia

A la vista de las imágenes, se puede comprobar que independientemente del ambiente donde se ubiquen, la red neuronal entrenada es capaz de detectar el plumero de la Pampa.

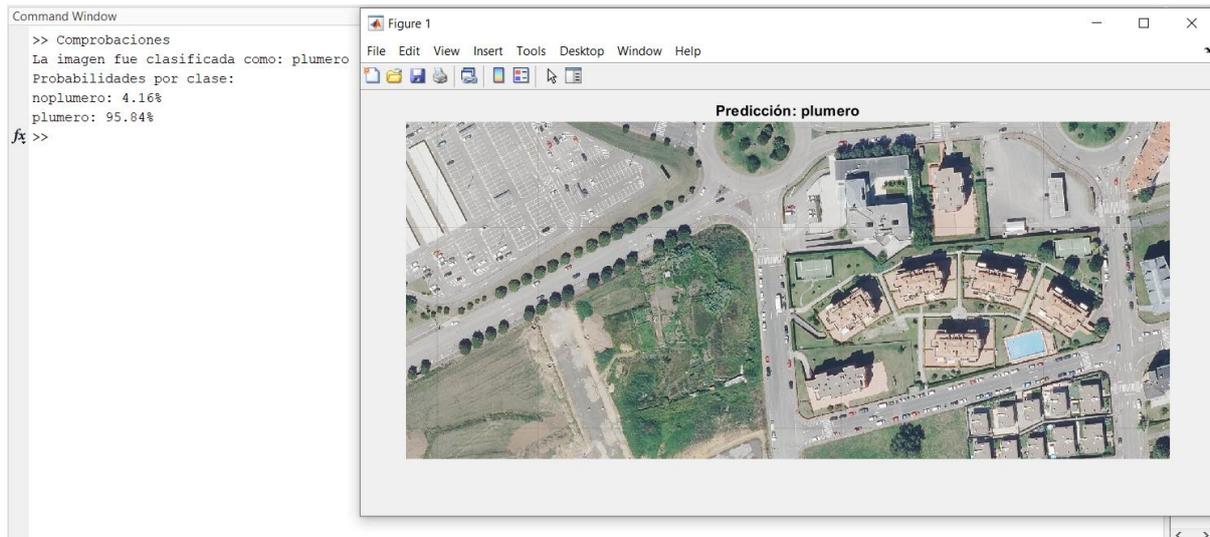


Figura 74. Resultado del proceso de inferencia con una imagen con plumero de la Pampa

Fuente: Elaboración propia

Además, se prueban especialmente las imágenes que poseen diferentes vegetaciones y arbolados, para comprobar que la red no sufre de incoherencias en las detecciones que cuenten con otras plantas.

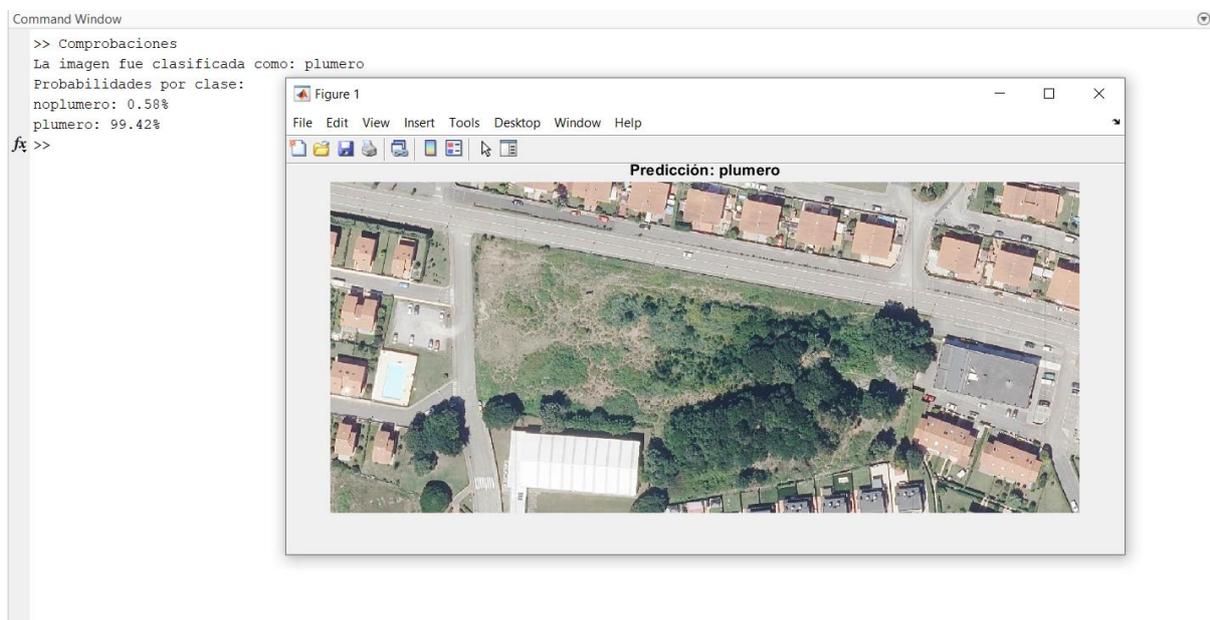


Figura 75. Resultado de una predicción de una imagen urbana con cortadería selloana

Fuente: Elaboración propia

Evidentemente, en los casos en los cuales la imagen tenga menos elementos con los cuales la red neuronal se pueda equivocar en el reconocimiento, el porcentaje de probabilidad del plumero de la Pampa es mayor.

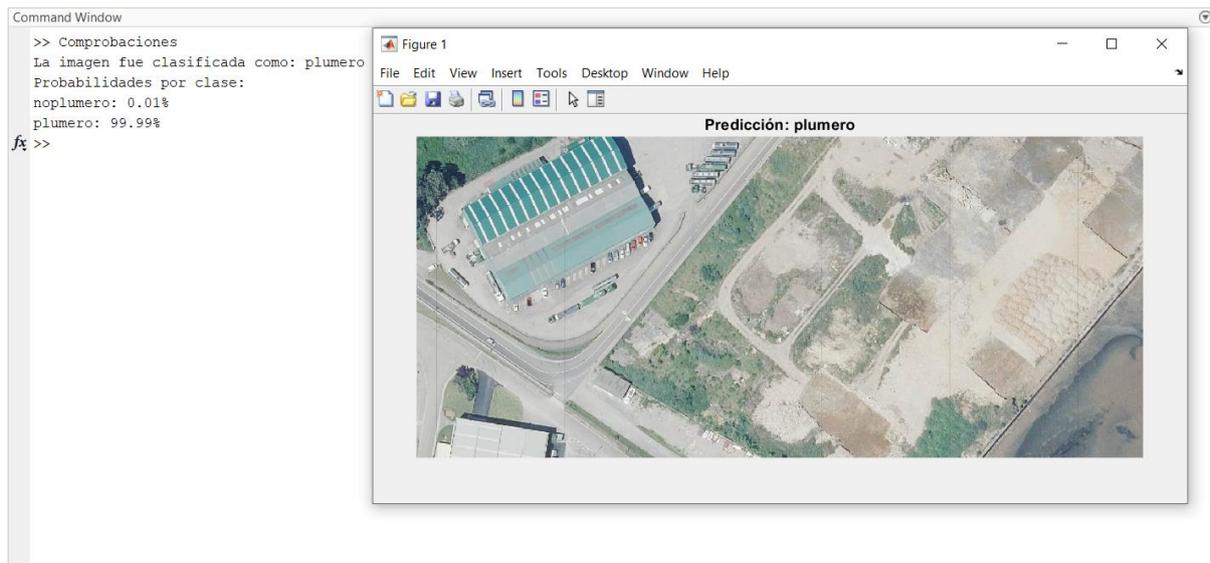


Figura 76. Predicción de una nueva imagen en un ambiente industrial

Fuente: Elaboración propia

Este tipo de predicciones se pueden realizar con otro tipo de imágenes que no posean plumero de la Pampa, de forma que se puede corroborar que la red neuronal es capaz de detectar fotografías sin este tipo de vegetación en la fase de inferencia. Para ello, solamente se debe cambiar la ruta de la imagen deseada.

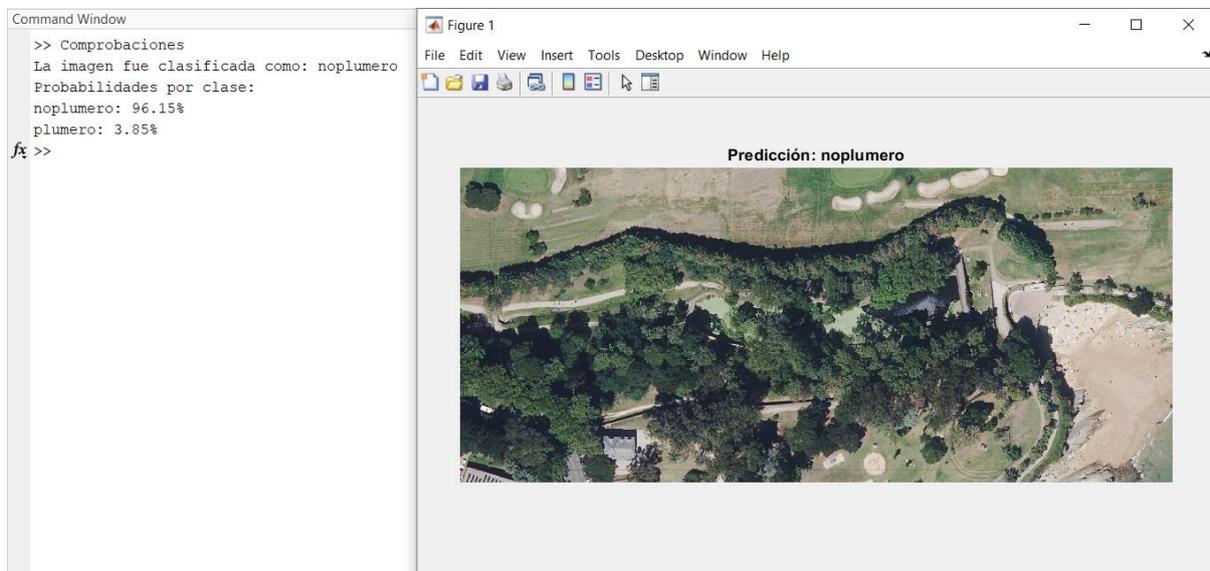


Figura 77. Predicción de una imagen sin cortadería selloana realizada por ResNet-50

Fuente: Elaboración propia

Como también se ha explicado previamente, en el dataset compuesto por imágenes sin plumero existen fotografías que, a pesar de presentar otras vegetaciones o árboles, poseen

formas que se asemejan a los cultivos de cortadería selloana. Como resultado, las predicciones empeoran con respecto a otras imágenes donde se aprecia claramente que no existe plumero de la Pampa.

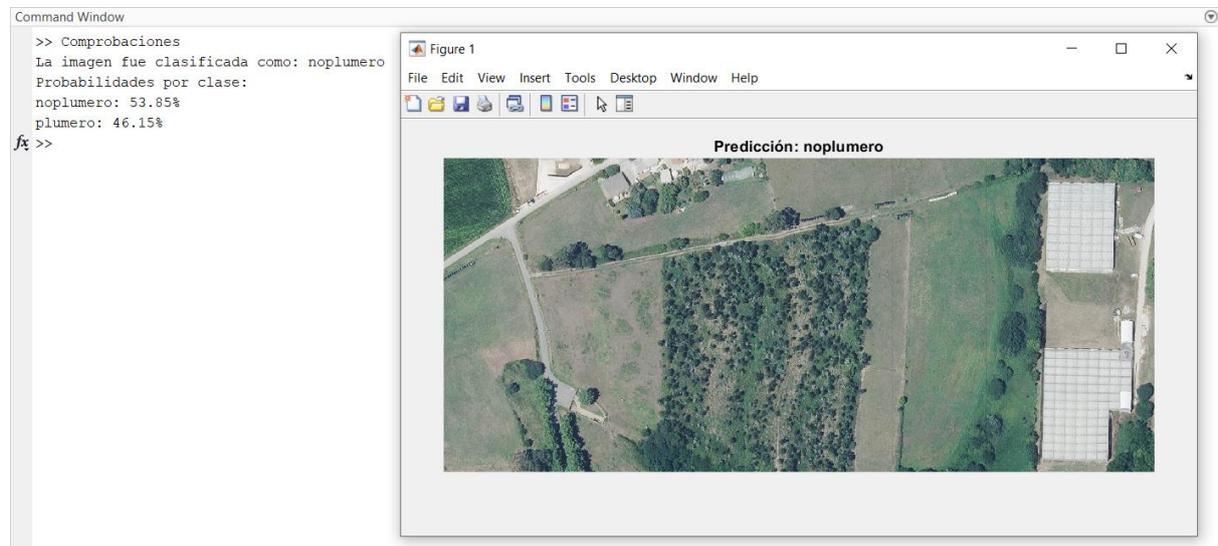


Figura 78. Predicción de una imagen sin plumero de la Pampa según ResNet-50

Fuente: Elaboración propia

A la vista de los diferentes procesos de inferencia realizados, se puede determinar que la red neuronal de ResNet-50 funciona correctamente. No obstante, diferentes factores del dataset son susceptibles de modificar los resultados obtenidos.

Con la finalidad de juzgar con mayor profundidad el desempeño de la red entrenada, se puede valorar su rendimiento usando imágenes extraídas a diferentes alturas, gracias al uso de SigPac. La altura desde la cual se extraen las imágenes está íntimamente relacionada con la complejidad y la robustez del modelo en tareas de reconocimiento y segmentación semántica mediante Deep Learning. Esto es debido a que las imágenes extraídas a menor altura poseen una mayor resolución y los objetos se aprecian con mayor detalle, mientras que las imágenes que son sacadas desde una altura mayor ofrecen más contexto del entorno, pero pierden finura en los detalles. Por otra parte, un mayor zoom en las imágenes también resalta otros elementos que no corresponden con los cultivos de plumero de la Pampa. Esto nos indica que, preliminarmente, no existe una proporcionalidad directa o inversa entre el enfoque y la precisión de la predicción. En añadidura, las imágenes del dataset son independientes entre sí y poseen ambientes diferentes, como paisajes urbanos, rurales, carreteras u otras vegetaciones.

Para ver esta problemática, se usa en primer lugar una imagen extraída desde una altura de 18.

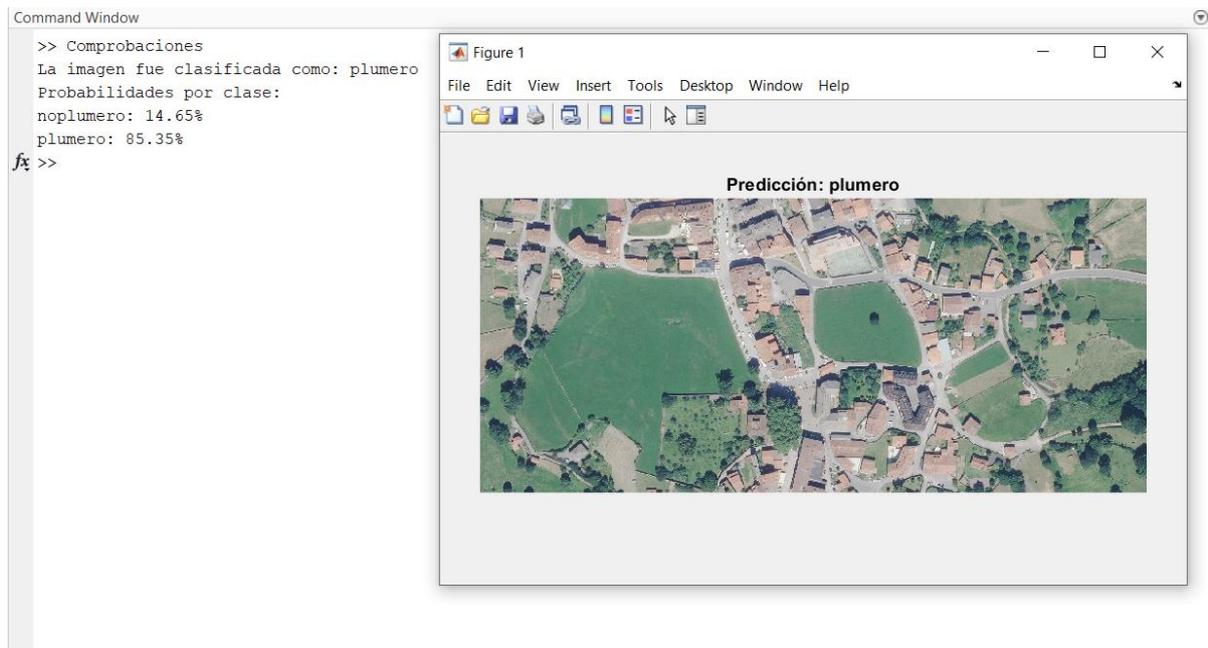


Figura 79. Predicción realizada en la imagen con un enfoque de 18

Fuente: Elaboración propia

La predicción es correcta, pero la existencia de más campos y zonas con árboles hace que el valor de la precisión sea mejorable. A continuación, se repite este mismo proceso con esa misma imagen, pero habiéndose extraído desde una de 19.

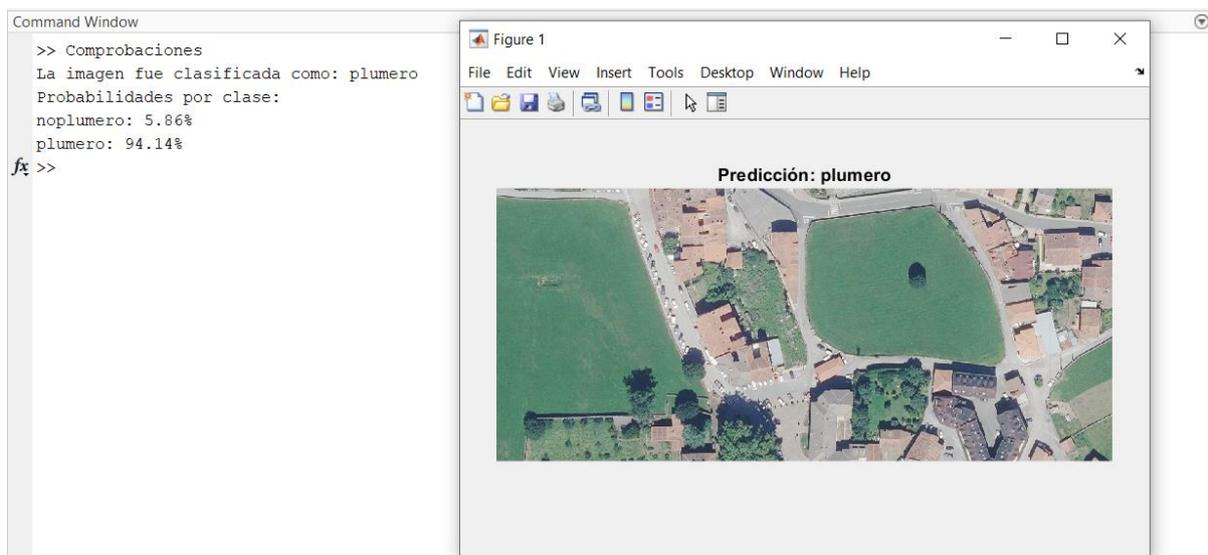


Figura 80. Predicción realizada en la imagen con un enfoque de 19

Fuente: Elaboración propia

Empleando otro ejemplo, también se puede verificar que dependiendo de la zona donde se enfoque la imagen, la precisión puede aumentar.

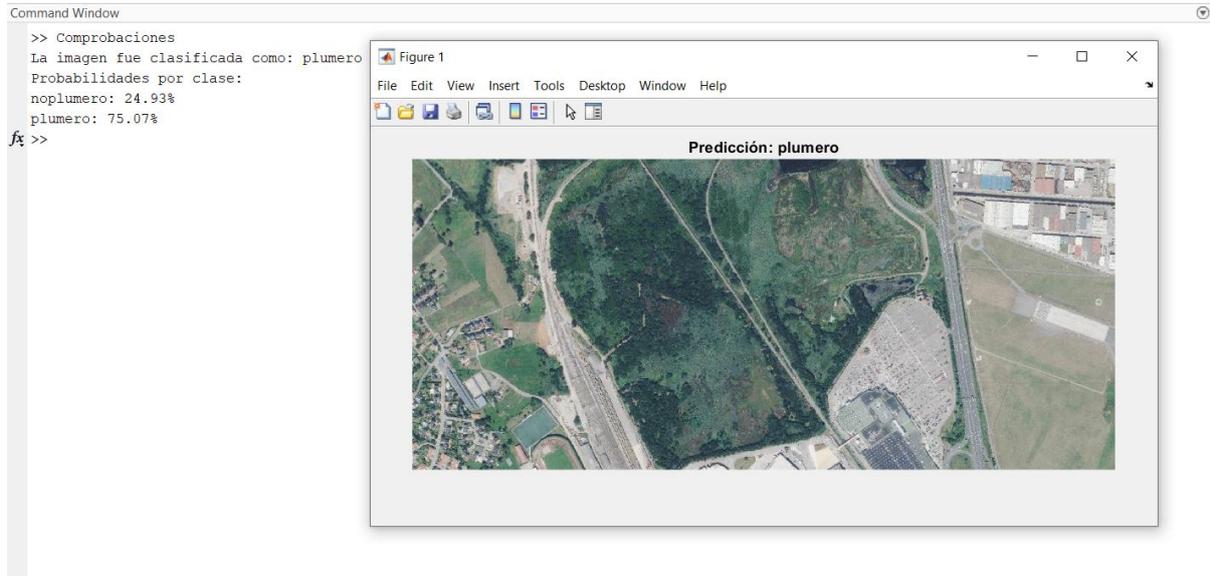


Figura 81. Predicción de una imagen con cortadería selloana utilizando un zoom de 16'5

Fuente: Elaboración propia

En ocasiones, las imágenes procedentes de SigPac al ser obtenidas con zooms mayores, se pueden desajustar, aunque si la región que posee cortadería selloana es grande, no supone un gran inconveniente.

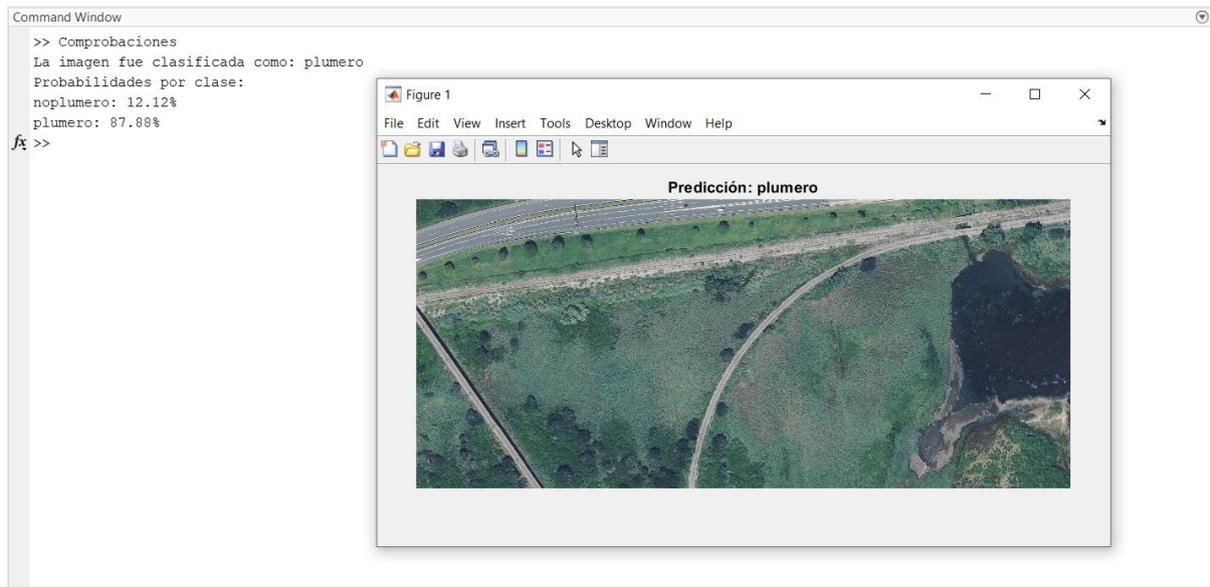


Figura 82. Predicción de una imagen con cortadería selloana usando un zoom de 18'5

Fuente: Elaboración propia

8.4. CONCLUSIONES

Como cabía esperarse, el proceso de inferencia realizado por la red neuronal AlexNet entrenada y la máquina de vectores de soporte entrenada no es correcto, puesto que las imágenes escogidas son complejas y poseen cierto grado de aleatoriedad. Esto se debe a que el plumero de la Pampa crece tanto en zonas urbanas como rurales y en añadidura, ciertas imágenes sin cortaderia selloana pueden provocar confusiones, al tener áreas similares como bosques o cultivos.

Por el contrario, la red neuronal residual es capaz de aprender correctamente qué forma posee la especie invasora en imágenes aéreas. Es más, se puede apreciar que ResNet50 es capaz de determinar si una imagen posee esta clase de vegetación incluso en imágenes extraídas con distintos valores de “zoom”. Gracias a esta capacidad de inferencia, escojo ResNet50 para la detección en imágenes que se realizará después.

9. SEGMENTACIÓN

9.1. INTRODUCCIÓN

La segmentación de imágenes permite al usuario la posibilidad de remarcar regiones de interés dentro de una misma fotografía, con el objetivo de distinguirlas entre sí y ofrecer una información más completa. A su vez, es uno de los problemas generales a los cuales se enfrenta la inteligencia artificial, dada la necesidad de destacar aquello que el usuario o sistema necesita, independientemente del campo de estudio, como la ubicación de vehículos en la calzada en un sistema de conducción autónoma o posibles lesiones o derrames en imágenes tomográficas.

La segmentación semántica asigna una etiqueta de clase a los píxeles de una imagen mediante el uso de un algoritmo de Deep Learning. De esta forma, se marcan los límites de los objetos y regiones de las imágenes, clasificándolas a través de diferentes características, como el color, la forma o el contraste. Aparte de esta clase de segmentación, también están la segmentación de instancias, que está enfocada en el reconocimiento de objetos contables como personas o animales y la segmentación panóptica, que combina tanto la segmentación semántica como la segmentación de instancias.

Existen diferentes tipos de procesos para segmentar una instantánea, en función de la información que dan. Principalmente, destacan los modelos de clasificación de imágenes, que determinan los elementos que posee una imagen, pero no sus ubicaciones, mientras que los modelos de detección de objetos aportan una información más detallada porque

determinan en qué parte de la imagen se encuentran los objetos, así como su forma y, por lo tanto, sus límites. Para todos los casos, se genera un mapa de segmentación de una imagen de entrada. En este mapa se reconstruye la imagen original otorgando una codificación para cada píxel referida a su clase semántica, con el objetivo de generar máscaras de segmentación. Dentro de la segmentación de imágenes, destaca el empleo de redes completamente convolucionales (FCN según sus siglas en inglés), U-Nets, redes de análisis piramidal de escenas o SegNet.

9.2. FCN

Las redes FCN es un tipo de arquitectura de redes neuronales especializada en tareas de predicción a nivel de píxel. Este tipo de estructuras sólo emplea capas convolucionales, por lo que la ausencia de capas densas hace que el tamaño de los datos de entrada sea flexible. Para su adaptación, se convierten todas las capas FC en capas convolucionales 1x1. Con el fin de clasificar cada píxel de la imagen, se añade otra nueva capa convolucional 1x1 con $C+1$ canales de salida, siendo C la cantidad de clases y sumando 1, que sería el fondo de la imagen.

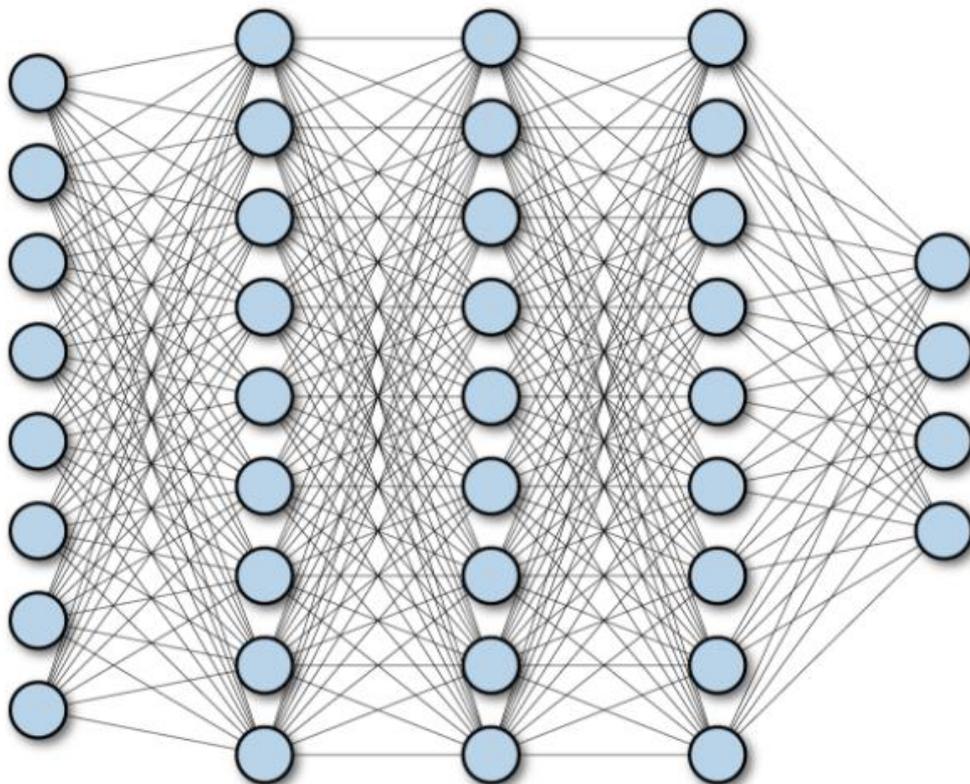


Figura 83. Imagen de una red neuronal completamente conectada

Fuente: [83]

Sin embargo, esta tipología de red neuronal acarrea un problema por la adición de más capas y por las operaciones de pooling que reducen el tamaño de los mapas de características, lo que provoca que el tamaño del mapa de características a la entrada sea diferente del tamaño a la salida. Por consiguiente, la segmentación a nivel de píxel no se podría realizar. Con el objeto de solucionarlo y obtener el sobremuestreo, se realizan convoluciones transpuestas para aumentar la resolución del mapa de características o el empleo de conexiones de salto, gracias a las cuales se pueden adquirir información más detallada. En la práctica, la convolución transpuesta aumenta el tamaño del input a través de la suma de los productos de cada uno de los términos de la matriz por el resto de ellos.

9.3. U-NET

Dentro de las arquitecturas FC, existen también las redes neuronales U-Net. Este tipo de estructuras se caracterizan por dividir su arquitectura en dos partes diferenciadas, una de ellas para contracción y otra para expansión, también llamada arquitectura encoder-decoder. En la etapa de contracción, que es equivalente a la parte del encoder, se reduce la información espacial de las imágenes mientras que la información de las características aumenta. En esta parte de la red U-Net se emplean convoluciones junto con capas ReLU en cada una de ellas, y una operación de pooling para reducir el tamaño de las características. Posteriormente, se desarrolla la etapa de expansión, equivalente a la parte del decoder, donde se reciben los mapas de características procedentes de la etapa de contracción. En esta fase se combina la información espacial mediante convoluciones y concatenaciones. La capa final es una convolución de dimensiones 1×1 con $C+1$ canales, siendo C el número de clases. Originalmente, el uso de U-Net se limitaba a la segmentación de imágenes médicas, lo que provocaba que no tuviera conjuntos de datos amplios y se requiriera de Data Augmentation para su correcto funcionamiento.

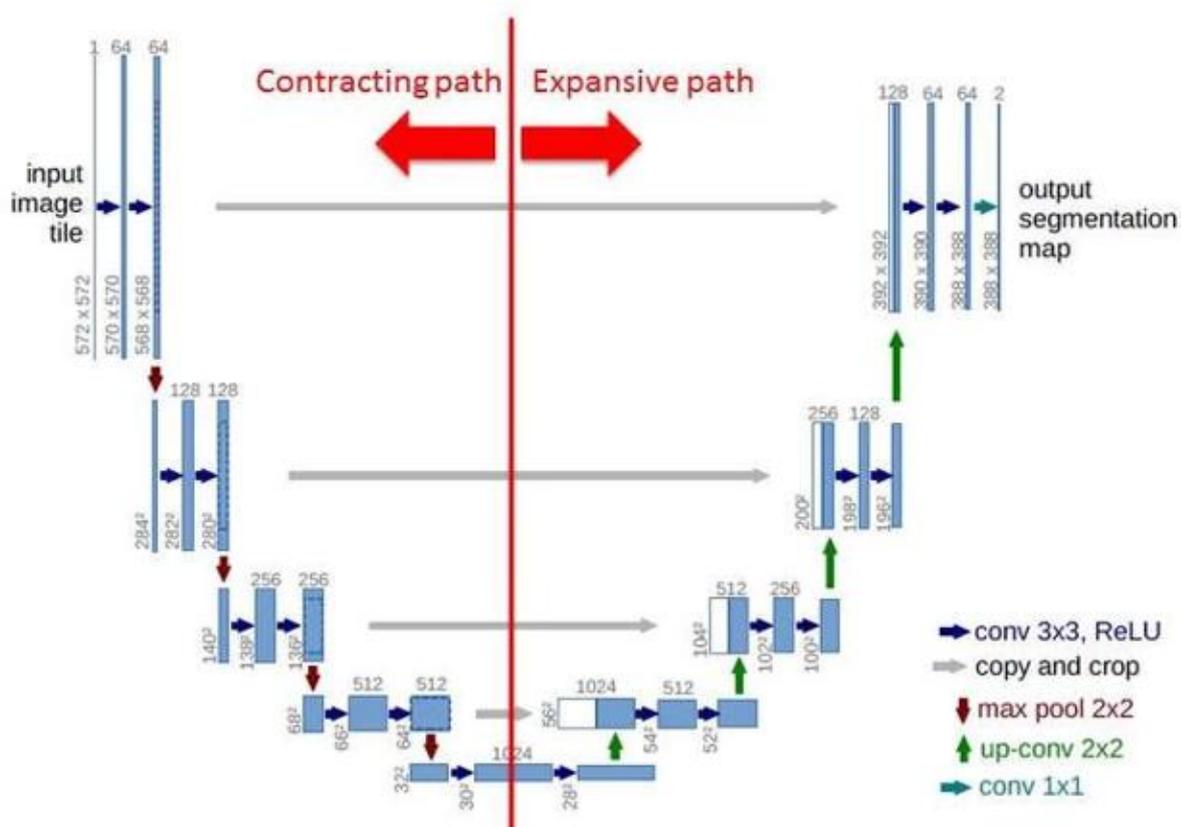


Figura 84. Representación de la arquitectura de la red neuronal U-Net

Fuente: [84]

9.4. PSPNET

Otro ejemplo de FCN es PSPNet, según las siglas en inglés de "Pyramid Scene Parsing Network". Este tipo de red neuronal introducida en 2017 se creó debido a las limitaciones derivadas de la complejidad de ciertas imágenes. En especial, FCN no aportaba un buen rendimiento al trabajar con fotografías con elementos que tuviesen escalas claramente diferentes. Para evitar esta problemática, PSPNet segmenta la imagen en cuestión utilizando diferentes escalas, dando lugar a una arquitectura de capas convolucionales en forma piramidal.

La imagen de entrada atraviesa la red neuronal para obtener el mapa de características resultante de la última capa. Después, mediante una operación de "pool", se envía la información al módulo piramidal, donde posteriormente se sobremuestra con el fin de obtener el mismo tamaño que el mapa de características original y se extraen los datos relevantes a diferentes escalas. Mediante una serie de capas concatenadas se extraen los mapas finales y a través de una capa convolucional final, se realiza la predicción.

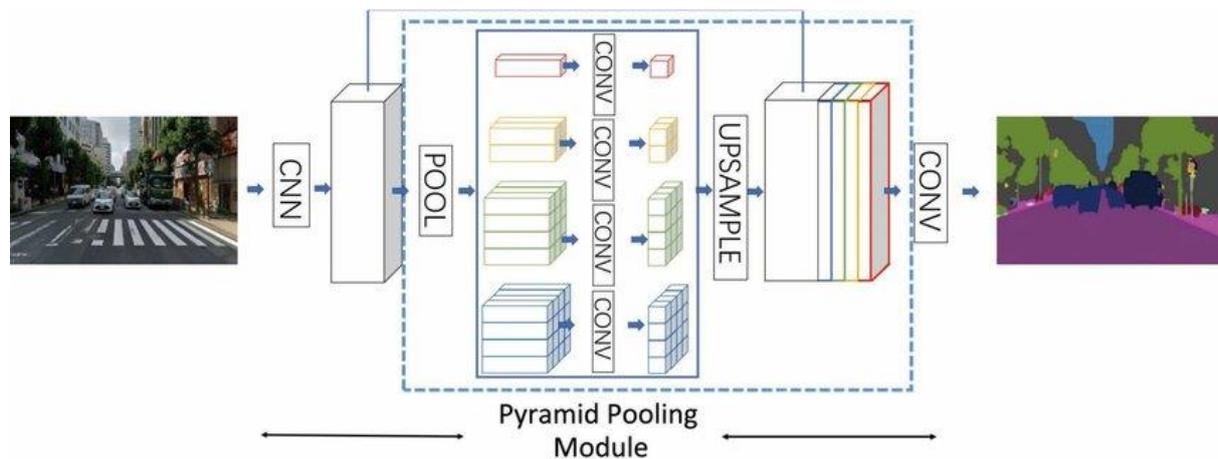


Figura 85. Representación de la arquitectura de PSPNet

Fuente: [85]

9.5. SEGNET

Otro modelo empleado para segmentación semántica es SegNet. Cuenta con una arquitectura convolucional Encoder-Decoder, pero sin usar capas completamente conectadas. Su input es una imagen, mientras que su output es un “pixel-wise label map”.

La primera etapa de su arquitectura, encargada de la codificación, captura características de alto nivel haciendo uso de capas convolucionales y capas “pooling”. No obstante, no almacena los mapas por completo, sino que guarda las localizaciones de los valores máximos de cada mapa de características del encoder. Posteriormente, en la segunda etapa, se sobremuestran los mapas de características mediante los “spatial pooling indices” procedentes del codificador, para la creación de un mapa de características pequeño. Este mapa se hace más grande usando un filtro entrenable. Tras este filtrado, los mapas se llevan a una capa de clasificación “softmax” para realizar la clasificación píxel a píxel.

Se considera a SegNet una buena opción de segmentación semántica gracias a la reutilización de los índices de “pooling” para decodificar, lo que aporta eficiencia computacional. Por otra parte, SegNet se diferencia de otras arquitecturas en su capacidad de entrenamiento con datos etiquetados limitados usando modelos preentrenados y además, los mapas de características obtenidos son de alta resolución.

9.6. OTROS MÉTODOS

Más allá de estas opciones, se generan posibles códigos de segmentación. El primero de ellos consiste en una segmentación semántica mediante el parcheado de la imagen. Cargando la fotografía deseada y la red neuronal ResNet entrenada anteriormente, se

determina el tamaño de los parches. En función de dicho tamaño, la imagen se divide en filas y en columnas.

```
[H, W, ~] = size(img);  
nRows = floor(H / patchSize);  
nCols = floor(W / patchSize);
```

Figura 86. Generación de los parámetros de la cuadrícula de la imagen

Fuente: Elaboración propia

Posteriormente, se crea la máscara de detección original y la perfecciono, de forma que se eliminen posibles huecos y detecciones parciales de las plantas. Después, se recorre toda la imagen etiquetando cada uno de los parches generados en la imagen. Si un determinado parche es detectado como “plumero” por la red neuronal entrenada, permanecerá en la imagen final que contiene la segmentación. Por último, se muestra por pantalla la imagen compuesta únicamente por aquellos píxeles que han sido catalogados como cortadería selloana.

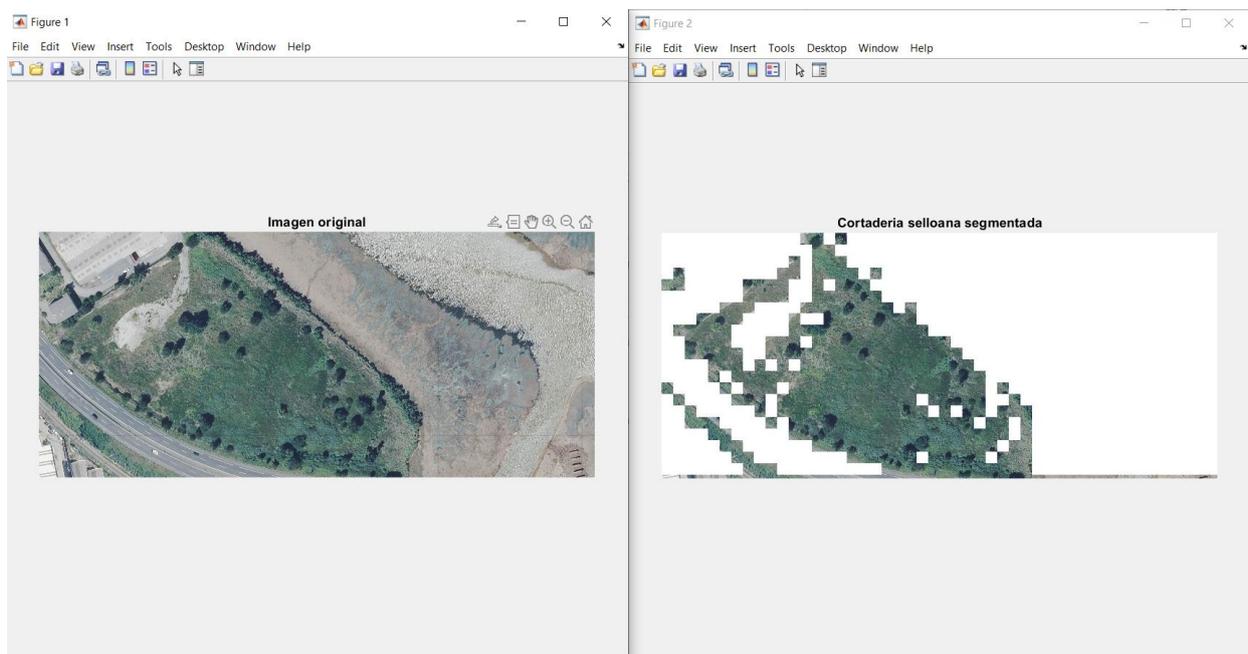


Figura 87. Comparativa entre la imagen original y la imagen con la cortadería selloana segmentada

Fuente: Elaboración propia

La segunda alternativa que se desarrolla utiliza “clústers”, de forma que segmenta la imagen en regiones que poseen propiedades similares. En primer lugar, se cambia la imagen a un espacio $L^*a^*b^*$. En este espacio, “L” corresponde a la luminosidad, mientras que el valor de

a^* abarca desde el color rojo hasta el color verde y el valor de b^* abarca desde el color amarillo hasta el color azul. En este caso, se utilizan únicamente 2 colores al haber sólo 2 clases y se hace la segmentación. La imagen resultante de esta segmentación por clases se devuelve por pantalla.

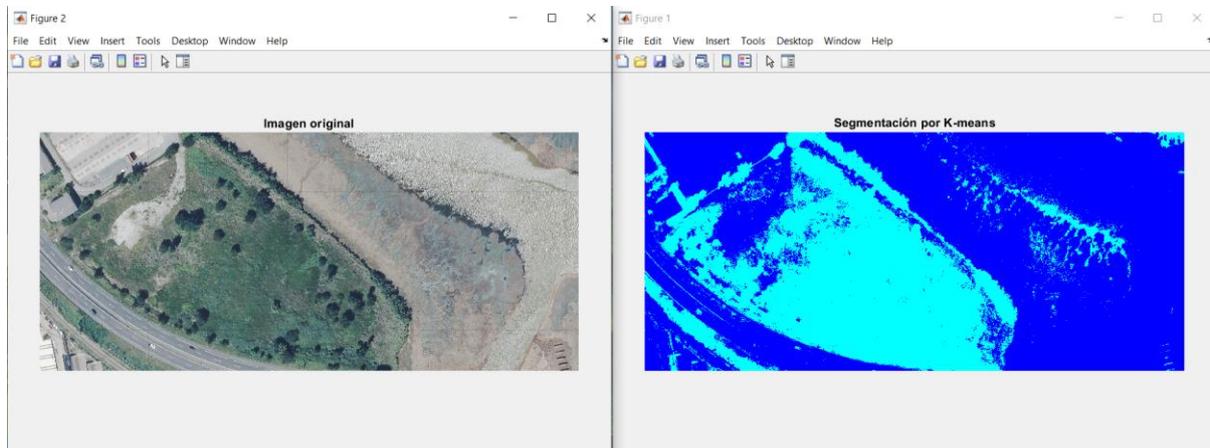


Figura 88. Representación conjunta de la imagen original y la imagen segmentada en clases

Fuente: Elaboración propia

En este caso, las máscaras resultantes de la segmentación semántica tienen un carácter aleatorio, mediante la agrupación de píxeles que poseen características similares como la textura o el color. Esto es debido a que en este caso no se utiliza la red neuronal residual entrenada, a diferencia de la segmentación mediante parcheado. Por este motivo, en el supuesto de tener 2 clases diferentes, se desconoce con anterioridad qué máscara almacena el fondo de la imagen y qué máscara guarda la clase deseada. Por consiguiente, puede ser necesario cambiar de máscara dependiendo de la imagen y representar su máscara inversa.

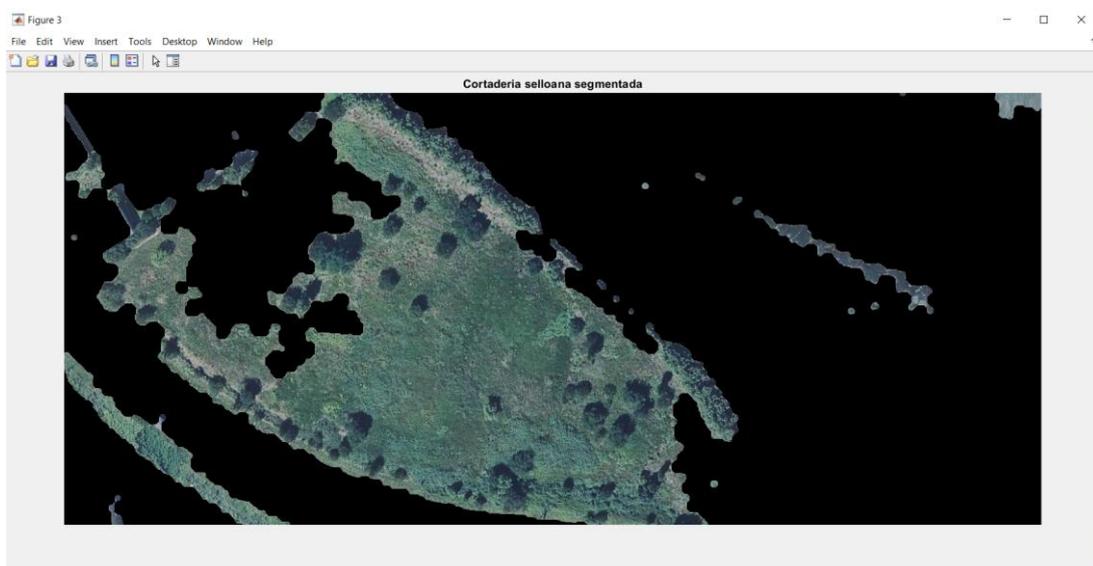


Figura 89. Representación del plumero de la Pampa segmentado en la imagen

Fuente: Elaboración propia

A pesar de que este último método de segmentación es la que mejor se puede ajustar a una correcta identificación de la cortaderia selloana, posee una particularidad que complica su utilización. Al dividir las imágenes en 2 clústeres, el plumero se encuentra en una máscara o en su defecto en su complementario, pero de manera aleatoria. Esto conlleva que no se sabe qué clúster contiene este tipo de vegetación y qué clúster posee el resto de los elementos de la fotografía antes de la ejecución, por lo cual puede ser necesario cambiar la máscara en una imagen determinada y usar su complementaria. Dicho problema dificulta que el proceso de segmentación se realice de manera automática salvo que se devuelvan las imágenes tras aplicar una máscara y su complementaria, lo que supone una desventaja con respecto al primer método.

9.7. CONCLUSIONES

Habiéndose analizado pormenorizadamente las distintas alternativas para realizar la segmentación del plumero de la Pampa, se ha llegado a la conclusión de que la segmentación en las imágenes se realizará mediante parcheado.

Al contrario de otras opciones, en esta técnica de segmentación se sabe con exactitud dónde se encuentra la especie invasora porque se emplea la red neuronal residual entrenada. Las alternativas a este código no poseen redes preentrenadas, de forma que adaptar cualquiera de ellas a esta función de segmentación es complicado. Igualmente, la opción de segmentación por parcheado ayuda a reflejar una nueva utilidad que pueden tener las redes neuronales, en el etiquetado de información en fotografías.

10. DESARROLLO DE LAS HERRAMIENTAS SOFTWARE

10.1 INTRODUCCIÓN

Más allá de los procesos de inferencia y segmentación semántica realizados con las imágenes del conjunto de datos, y antes de desarrollar las aplicaciones, se genera una función en Matlab de forma que se pueda detectar cortaderia selloana a partir de una imagen aérea cualquiera, si se estipulan por teclado la latitud, la longitud y el zoom. Esta función utiliza fotografías procedentes de PNOA, debido a que ofrece las imágenes con mejor resolución posible dentro de las imágenes que pueden ser utilizadas y tratadas con inmediatez. Otra opción es obtener imágenes a partir de Google Maps, pero poseen una

calidad claramente peor, lo cual supone un problema a la hora de detectar y segmentar el plumero de la Pampa. Determinando el encuadre de la nueva imagen mediante la latitud, la longitud y el zoom, es posible descargar la imagen deseada correspondiente a la información introducida.

```
delta = 0.1 / zoom;  
latlim = [lat - delta, lat + delta];  
lonlim = [lon - delta, lon + delta];
```

Figura 90. Ajustes necesarios para obtener la imagen correcta

Fuente: Elaboración propia

Una vez se ejecuta el programa, se devuelve por pantalla la imagen junto con la información introducida.

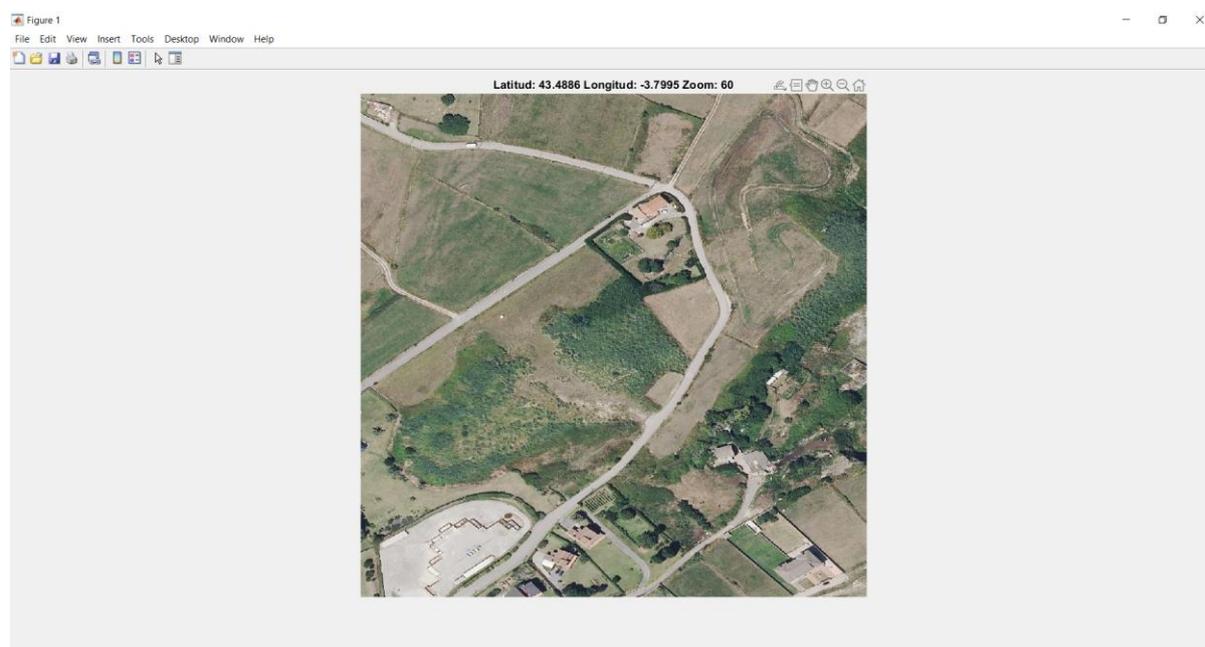


Figura 91. Imagen con cortadería selloana obtenida mediante la función explicada anteriormente

Fuente: Elaboración propia

Esta función permite a continuación obtener las imágenes que serán analizadas por la red neuronal entrenada para detectar si existe plumero de la Pampa y en tal supuesto, segmentarlo.

Además, para ayudar a la representación en la aplicación, se realiza una función que dibuje un punto rojo en un mapamundi a partir de la latitud y de la longitud. Para ello, se deben tener en cuenta 2 factores importantes. El primero de ellos es la utilización de un mapa equirrectangular, gracias al cual se obtiene una relación directa y correcta entre los datos de entrada y el punto obtenido. A su vez, PNOA devuelve las imágenes obtenidas como planos equirrectangulares, lo que facilita la tarea y el código. No obstante, la latitud y la longitud

determinadas al ejecutar el programa son traducidas por código a unas coordenadas cartesianas (x, y), que han de existir dentro del mapa.

```
x = ((longitud + 180) / 360) * ancho;  
y = ((90 - latitud) / 180) * alto;
```

Figura 92. Conversión de la longitud y de la latitud en coordenadas x e y

Fuente: Elaboración propia

Al ejecutar el programa, se devuelve una representación del mapa mundial con un indicador que corresponde al punto determinado por la latitud y la longitud.

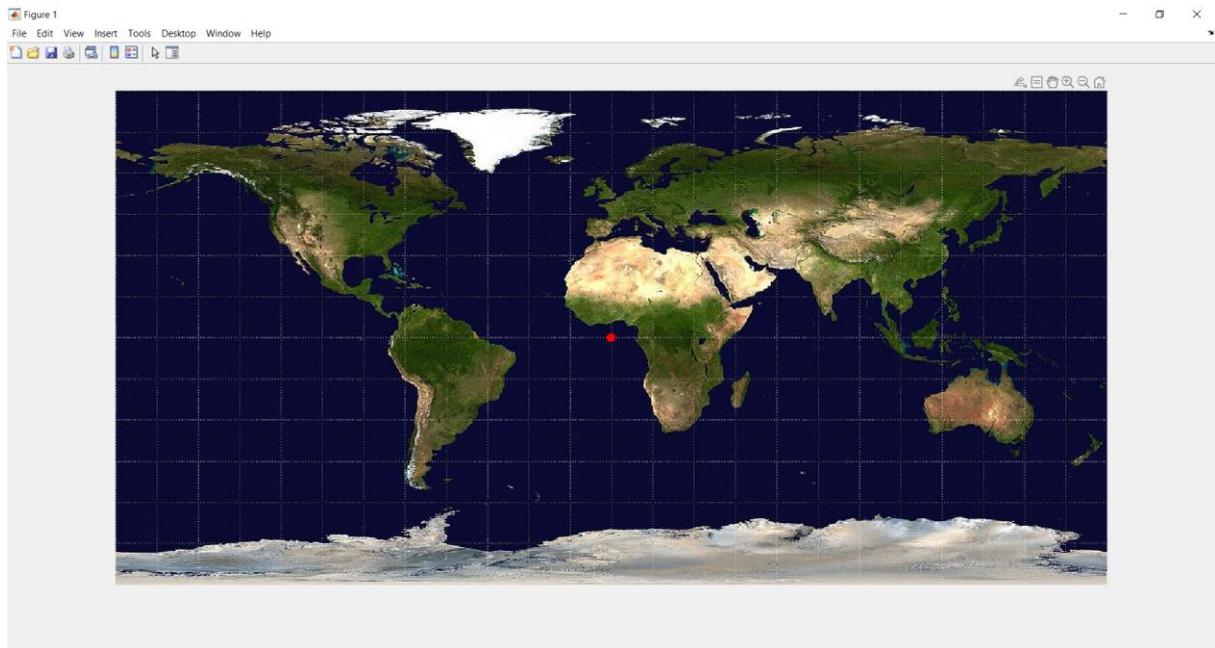


Figura 93. Imagen resultante del anterior código para (0,0) en latitud y longitud

Fuente: Elaboración propia

Para una mayor facilidad para el usuario, se diseña una aplicación que permita detectar cortaderia selloana en un mapa. Con dicho objetivo, utilizo App Designer. Este entorno de Matlab posibilita el diseño y la programación de una aplicación, de una manera mucho más interactiva y fácil para los usuarios, sin necesidad de tener formación previa como desarrollo de software. A su vez, permite migrar otro tipo de proyectos para trabajar con ellos, como GUIDE. Para el diseño, el usuario puede hacer uso de una librería de componentes, los cuales pueden ser arrastrados hasta la vista de diseño y pueden modificarse sus propiedades como el nombre o la fuente.

10.2. HERRAMIENTA SOFTWARE GLOBAL

En una primera versión de la interfaz, se utiliza la función para obtener imágenes desde PNOA, la red neuronal ResNet-50 entrenada para la detección de cortaderia selloana y la parte del código enfocado en la segmentación de dicha especie en las imágenes adquiridas.

Para el diseño de la ventana del programa, App Designer permite el uso de diversos componentes.

La interfaz original consta de 3 campos de "Edit Field (Text)", creados para la adquisición de la longitud, la latitud y el zoom del punto deseado. Por otra parte, existen 3 botones ("Button", según su designación en la librería). Estos botones serán utilizados de manera consecutiva en la ejecución del programa, pero poseen finalidades distintas. El primero de ellos es "Obtener imagen". Dicho botón es clicado después de escribir por teclado los datos de entrada. Después al cargarse la imagen en la interfaz, se ha de pulsar el otro botón llamado "Detectar plumero", mediante el cual, tras un espacio de tiempo, se obtiene la segmentación de la cortaderia selloana en la imagen. En último lugar, tras conocer mediante el indicador visual si la imagen posee plumero de la Pampa, dicha fotografía puede ser descargada en su carpeta correspondiente. Este directorio será elegido por código en función de la presencia o de la ausencia de cortaderia selloana. En este caso, la interfaz posee 2 elementos "Axes", utilizados para cargar imágenes. La primera imagen que se debe cargar procede de PNOA y se carga a la izquierda, mientras que a la derecha se cargará un mapa mundial con un círculo rojo que marca la localización introducida.

Como optimización en este código, se programan además dos indicadores en forma de cartel. El primero de ellos se ilumina de color azul celeste mientras se procesa la imagen en busca de cortaderia selloana. La finalidad de esta señalización es informar al usuario de que, aunque parezca que el programa no está funcionando, está trabajando en el reconocimiento de la imagen. El segundo indicador que, en caso de aparecer, se ilumina de color amarillo, manifiesta que se ha conseguido identificar plumero de la Pampa en el punto escogido, usando el zoom determinado anteriormente. Por el contrario, en el caso de que no exista dicha vegetación en la imagen, se mostrará un mensaje por pantalla.

En lo que respecta al código, se programan 4 funciones. Las 3 primeras funciones corresponden al funcionamiento de los botones, mientras que la última responde a una optimización del código en caso de que se detecte correctamente la planta invasora.

En primer lugar se encuentra la función encargada de obtener la imagen procedente de PNOA. En el comienzo, además de reinicializar el indicador de presencia de cortaderia selloana, se hace la conversión de los valores introducidos. Esto se debe a que inicialmente son strings, pero para poder trabajar con ellos deben ser valores numéricos. En el supuesto de que algún dato sea incorrecto, se mostrará un mensaje al usuario por pantalla.

```

%Obtención de los datos de entrada: latitud, longitud y zoom
lat = str2double(app.LatitudEditField.Value);
lon = str2double(app.LongitudEditField.Value);
zoom = str2double(app.ZoomEditField.Value);
%Si se ha introducido algún punto de manera incorrecta, lo
%informo
if isnan(lat) || isnan(lon) || isnan(zoom)
    uialert(app.UIFigure, 'Las coordenadas o el zoom son inválidos. Vuelve a introducir la información', 'Error');
    return;
end

```

Figura 94. Parte del código correspondiente a la entrada de datos

Posteriormente, se llama a las funciones que se encargan de obtener la imagen de PNOA y la representación de un punto en el mapamundi. Una vez que se realiza estas llamadas, se muestra cada una de ellas en sus correspondientes figuras de la interfaz.

```

%Obtención de la imagen
img = Descargadeimagenes(lat, lon, zoom);
%Marcar lugar en mapa mundi
img2 = puntomapa(lat,lon);
if size(img, 3) == 1
    img = cat(3, img, img, img); % Conversión a RGB
end
%Muestro las imagenes
setappdata(app.UIFigure, 'ImageData', img);
% Mostrar imagen en UIAxes de forma compatible
cla(app.Mapa); % Limpia contenido previo
image(app.Mapa, img); % Muestra imagen correctamente
axis(app.Mapa, 'off'); % Oculta ejes
app.Mapa.Visible = 'off'; % (opcional) Oculta bordes

setappdata(app.MapaMundi, 'ImageData', img2);
% Mostrar imagen en UIAxes de forma compatible
cla(app.MapaMundi); % Limpia contenido previo
image(app.MapaMundi, img2); % Muestra imagen correctamente
axis(app.MapaMundi, 'off'); % Oculta ejes
app.MapaMundi.Visible = 'off'; % (opcional) Oculta bordes
catch ME
uialert(app.UIFigure, ME.message, 'Error');
end

```

Figura 95. Parte del código correspondiente a la descarga y a la visualización de las imágenes

Después, se ejecuta la función encargada de la detección del plumero de la Pampa. En este nuevo procedimiento, se utiliza la segmentación semántica mediante parchado. Determinando las dimensiones de los parches y siendo etiquetados por la red neuronal

residual entrenada, se debe realizar la clasificación mediante estructuras condicionales para discernir qué cantidad de parches se considera como plumero de la Pampa, de forma que se consigan el menor número de reconocimientos incorrectos, tanto falsos positivos como falsos negativos. Después de una serie de pruebas buscando la mejor forma de programar esta problemática, se llega a una conclusión compuesta por el número de parches junto con el zoom de la imagen.

```
if numerodeparches > 1000 || numerodeparches < 20
    uialert(app.UIFigure, 'La imagen no posee cortaderia selloana', 'Error');

elseif numerodeparches < 1000 && numerodeparches > 670 && zoom < 70
    mostrarExito();

elseif numerodeparches <= 670 && numerodeparches > 200
    mostrarExito();

elseif numerodeparches <= 200 && numerodeparches > 20 && zoom > 80 && zoom < 100
    mostrarExito();

else
    uialert(app.UIFigure, 'La imagen no posee cortaderia selloana', 'Error');
end
```

Figura 96. Serie de condicionales que garantizan el correcto funcionamiento del reconocimiento

Fuente: Elaboración propia

En cada uno de los condicionales satisfactorios, se llama a la función “mostrarExito()”. Primeramente, esta función muestra el indicador en el supuesto de que el reconocimiento de cortaderia selloana sea satisfactorio. En todos estos casos, si la imagen resultante de la introducción de los datos de entrada posee esa vegetación, se descargará la imagen procedente de PNOA con el objetivo de aumentar el conjunto de datos. Para optimizar este proceso, dicha fotografía tendrá como formato “latitud_longitud.jpg”. Este método uniformiza el nombre de los archivos y facilita la futura representación que se hará en el mapa. Si la imagen no se puede descargar, se informará al usuario mediante un aviso.

Una vez se ejecuta el programa, aparece la interfaz en blanco a la espera de la introducción de la información.

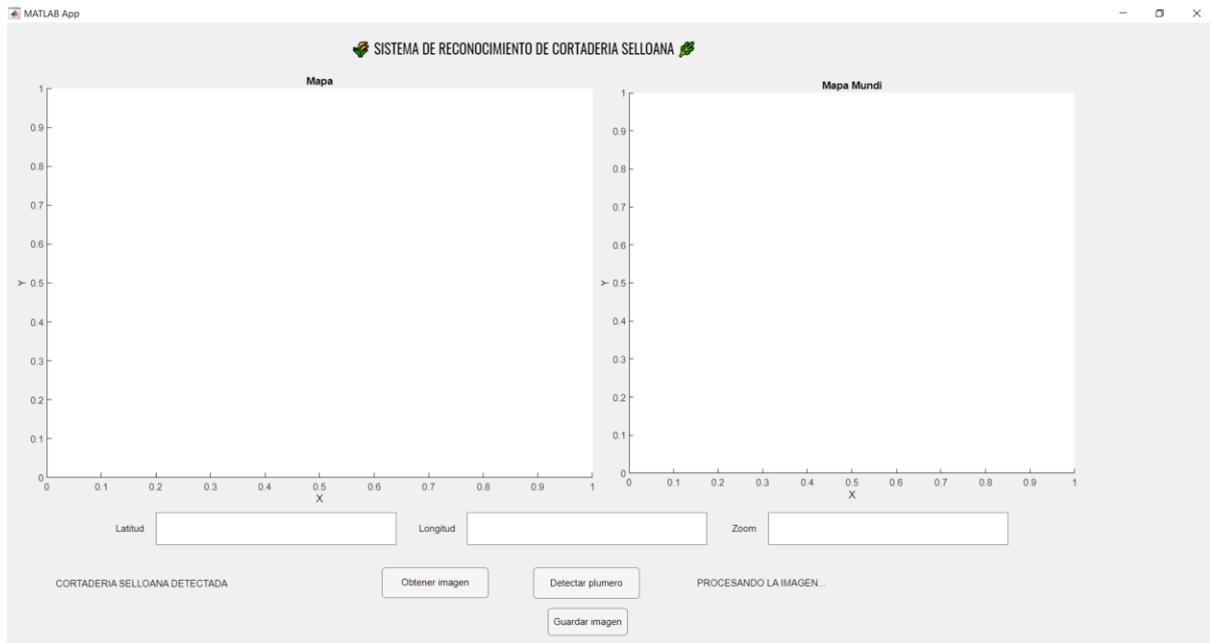


Figura 97. Captura de la interfaz antes de reconocer el plumero de la imagen

Fuente: Elaboración propia

Tras introducir los valores de latitud, longitud y zoom, si se pulsa en “Obtener imagen”, cargará tanto la fotografía procedente de PNOA como la ubicación señalada en el mapamundi.

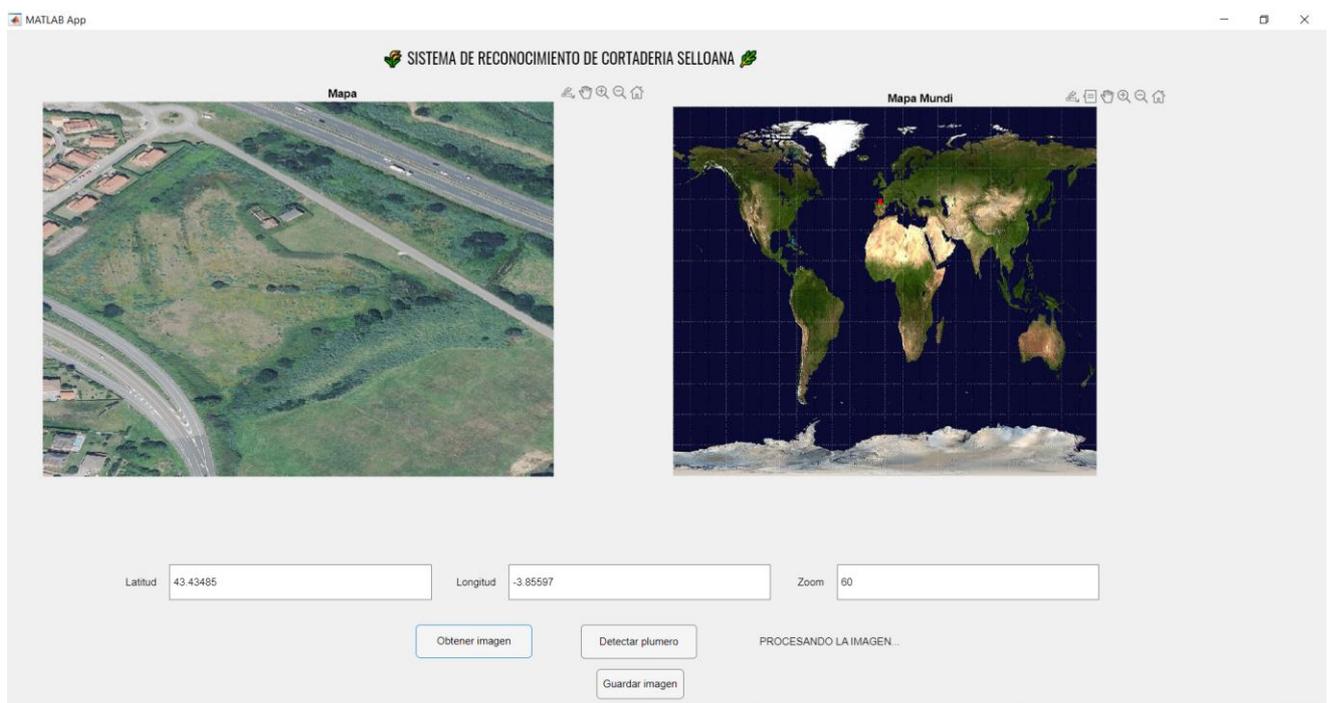


Figura 98. Captura de la inferfaz tras la carga de la imagen satelital

Fuente: Elaboración propia

Posteriormente, clicando en “Detectar plumero”, se ilumina el indicador que informa al usuario de que el programa está procesando la imagen deseada.

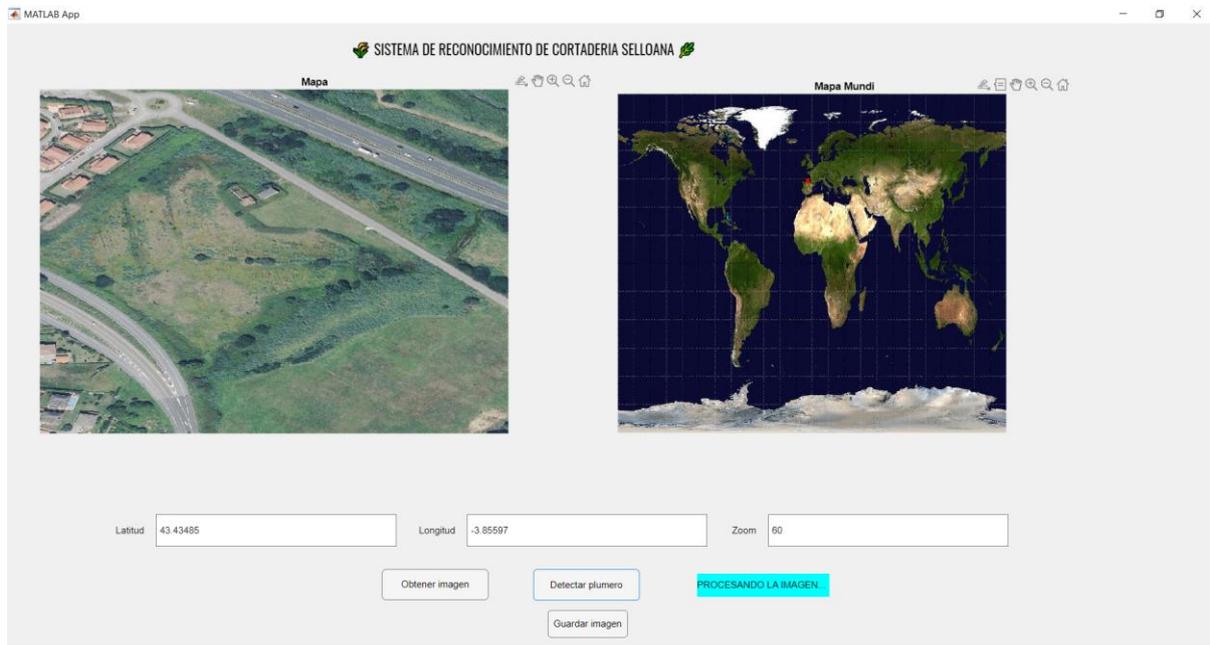


Figura 99. Procesamiento de la segmentación de la cortadería selloana en la imagen

Fuente: Elaboración propia

Por último, se devuelve la imagen con la segmentación. En el caso de que exista cortadería selloana en la imagen, se iluminará el indicativo para informar al usuario.

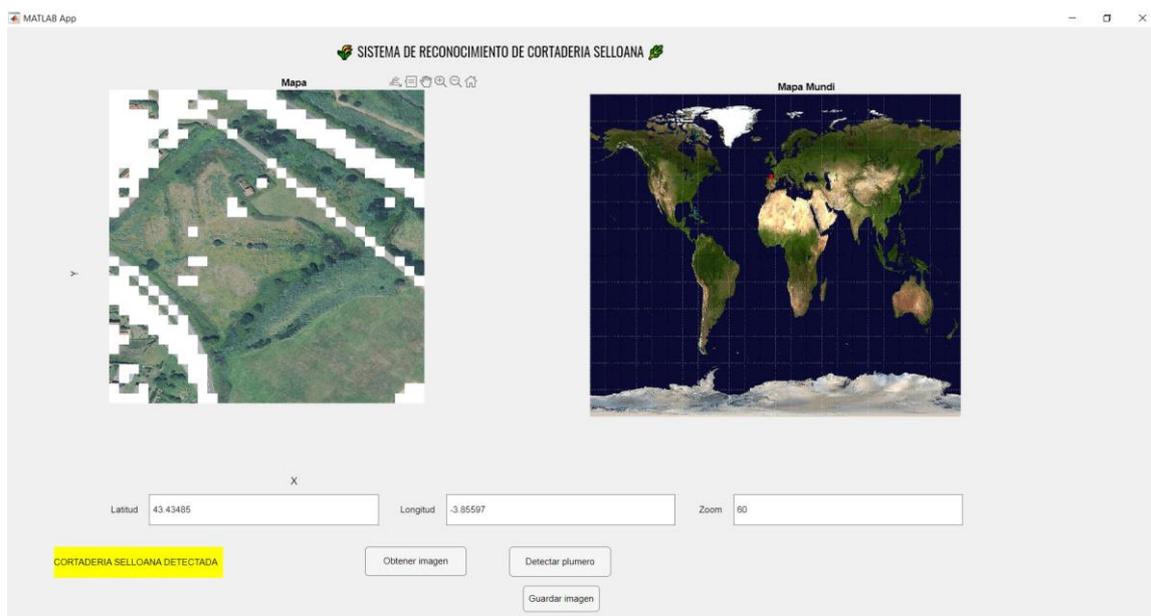


Figura 100. Devolución de la imagen con la segmentación de plumero de la Pampa

Fuente: Elaboración propia

Una vez se obtiene la segmentación, se pulsa en “Guardar imagen”, para almacenar la fotografía en su carpeta correspondiente. Este procedimiento se puede repetir el número de ocasiones que se desee.

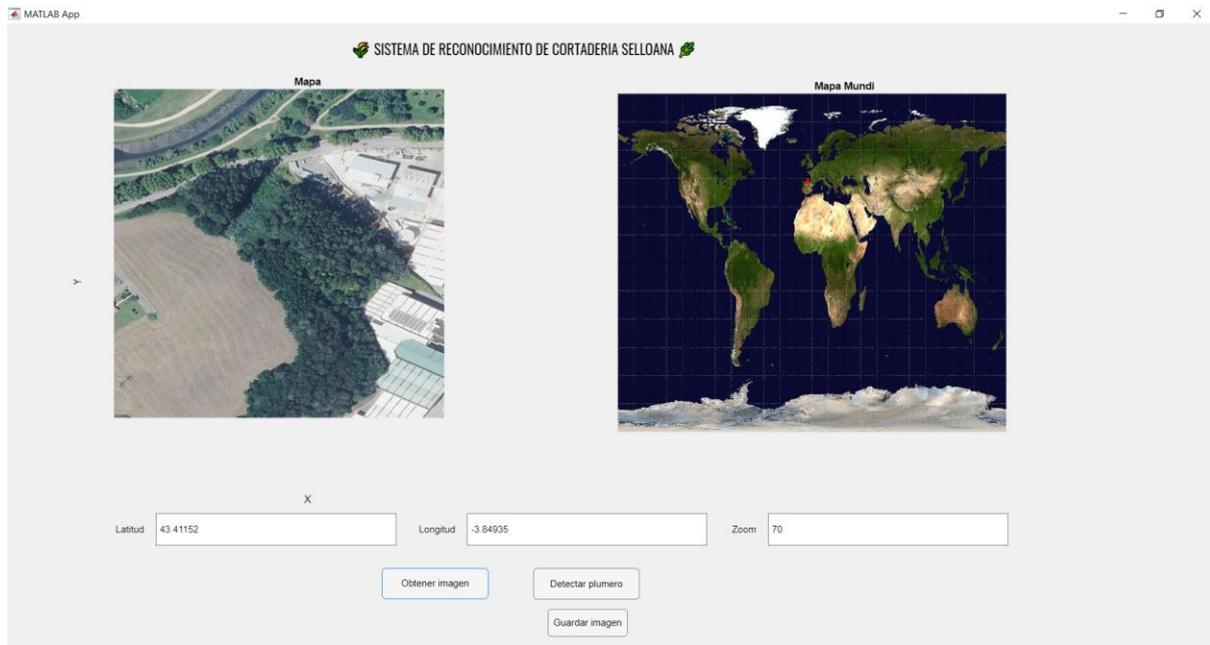


Figura 101. Carga de una imagen sin cortadería selloana

Fuente: Elaboración propia

Independientemente de que sea conocido que la imagen seleccionada no posee plumero de la Pampa, el procedimiento realizado por el programa no varía.

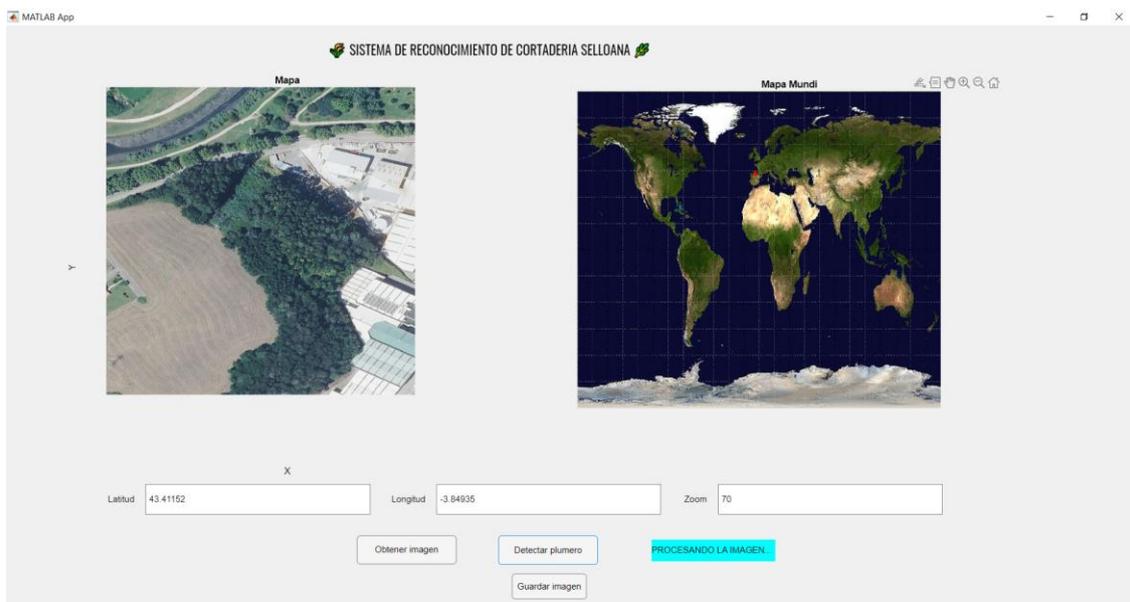


Figura 102. Procesamiento de la imagen anterior

Fuente: Elaboración propia

En este caso, en lugar de iluminarse ningún indicador en la pantalla, el programa devuelve al usuario un mensaje de error, informando de que no existe la especie vegetal en la imagen.

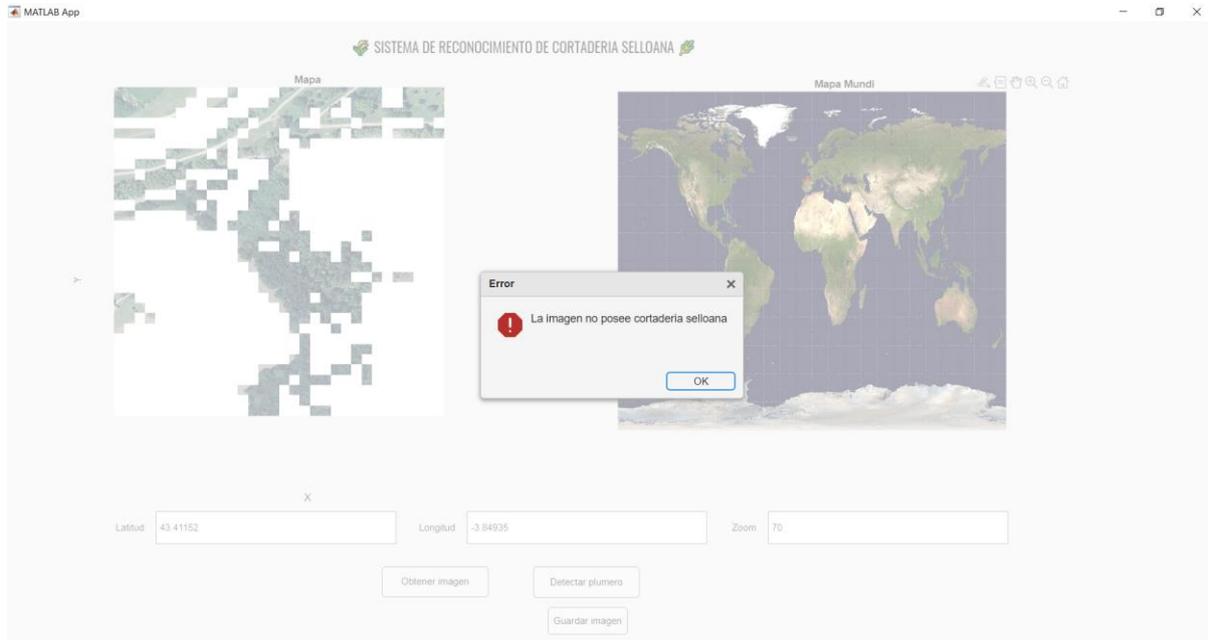


Figura 103. Reconocimiento correcto en caso de ausencia de cortadería selloana

Fuente: Elaboración propia

10.3. HERRAMIENTA SOFTWARE PARA CANTABRIA

Con el objetivo de centrar y prevenir la expansión de cortadería selloana únicamente en el territorio cántabro, se genera otra aplicación en la cual solamente se puedan cargar localizaciones de Cantabria. En este nuevo caso, para la admisión exclusiva de coordenadas pertenecientes a esta comunidad autónoma, se realiza un condicional que permita delimitar el territorio, a partir de los datos introducidos referentes a la latitud y a la longitud.

```
if lat>43.5124 || lat<42.76212 || lon<-4.85064 || lon>-3.15135
    uialert(app.UIFigure, 'Las coordenadas no corresponden al territorio cántabro', 'Error');
end
```

Figura 104. Condicional donde se verifica que la localización se encuentra en Cantabria

Fuente: Elaboración propia

En lo referido al código es idéntico al código original, incluyendo un botón para poder descargar las imágenes que se almacenarán en las mismas carpetas que en el programa anterior dependiendo de su caso particular. No obstante, para una mayor complejidad, se

avisará al usuario de que el proceso de detección de la vegetación está en curso mediante un indicador amarillo, mientras que si el reconocimiento realizado es positivo, se encenderá otro indicador de color verde.

```
% Encendido de la lámpara de color amarillo
app.Enproceso.Visible = 'on';
app.Enproceso.Color = [1 1 0];
drawnow;
```

Figura 105. Parte del código donde se determina el encendido del indicador amarillo

Fuente: Elaboración propia

En este caso, realizo el mismo procedimiento en caso de que se identifique plumero de la Pampa, llamando a una función con el mismo nombre que en el caso anterior, “mostrarExito()”.

```
if numerodeparches > 1000 || numerodeparches < 20
    uialert(app.UIFigure, 'La imagen no posee cortadería selloana', 'Error');

elseif numerodeparches < 1000 && numerodeparches > 670 && zoom < 70
    mostrarExito();

elseif numerodeparches <= 670 && numerodeparches > 200
    mostrarExito();

elseif numerodeparches <= 200 && numerodeparches > 20 && zoom > 80 && zoom < 100
    mostrarExito();

else
    uialert(app.UIFigure, 'La imagen no posee cortadería selloana', 'Error');
end
```

Figura 106. Conjunto de supuestos en los cuales se enciende el indicador verde

Fuente: Elaboración propia

Esta función difiere de la que se explicó en el otro programa, puesto que en este caso, el mapa situado a la derecha irá marcando las diferentes localizaciones donde se encuentre cortadería selloana. Estas localizaciones se leen gracias al formato de guardado de las imágenes. Por otra parte, las imágenes que se guarden, se almacenarán en una carpeta distinta a la carpeta utilizada en el anterior programa. De esta manera, se evita que se puedan leer localizaciones no correspondientes a Cantabria. Por consiguiente, las imágenes que corresponden con la cortadería selloana y son de esta comunidad autónoma no se deben juntar con las otras imágenes de plumero de la Pampa obtenidas en el otro programa.

```
if app.DeteccionDeCortaderia.Visible == 'on'  
    carpeta = 'C:\Users\Win10 User\Desktop\TFG\plumero\nuevasimagenesdecortaderiaselloanacantabria';
```

Figura 107. Líneas de código correspondientes al guardado de las imágenes en este programa

Fuente: Elaboración propia

Después de la programación, se pasa a probar el programa. En primer lugar, si las coordenadas introducidas no pertenecen a Cantabria, se debe mostrar un mensaje por pantalla al usuario, para que corrija los datos de entrada.



Figura 108. Captura del programa en caso de que la localización no pertenezca a Cantabria

Fuente: Elaboración propia

Si las coordenadas son correctas, se carga la imagen a la izquierda, y pulsando el botón “Detectar plumero”, se enciende una luz amarilla, que indica que la imagen se está procesando en busca de cortadería selloana. Cuando el programa detecte un reconocimiento positivo de plumero de la Pampa, se enciende el indicador de color verde y la ubicación se rodea en el mapa.

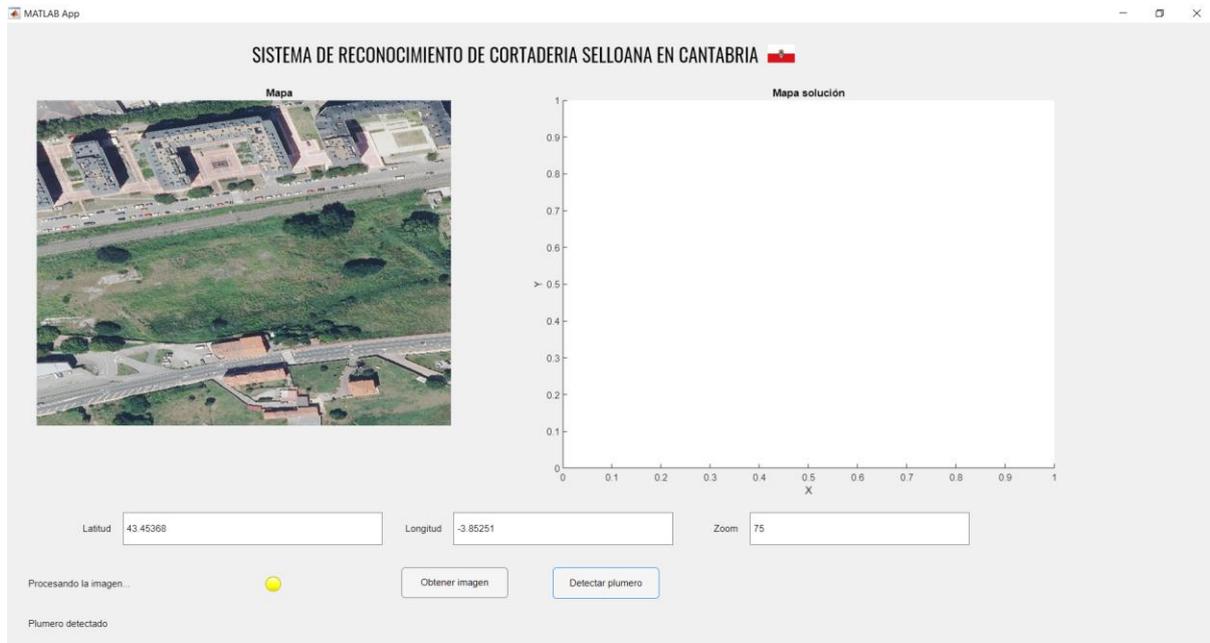


Figura 109. Procesamiento de una imagen durante el transcurso del programa

Fuente: Elaboración propia

El indicador luminoso amarillo también es útil porque en el transcurso del reconocimiento de plumero de la Pampa, el programa está trabajando aunque no se aprecie de manera directa.

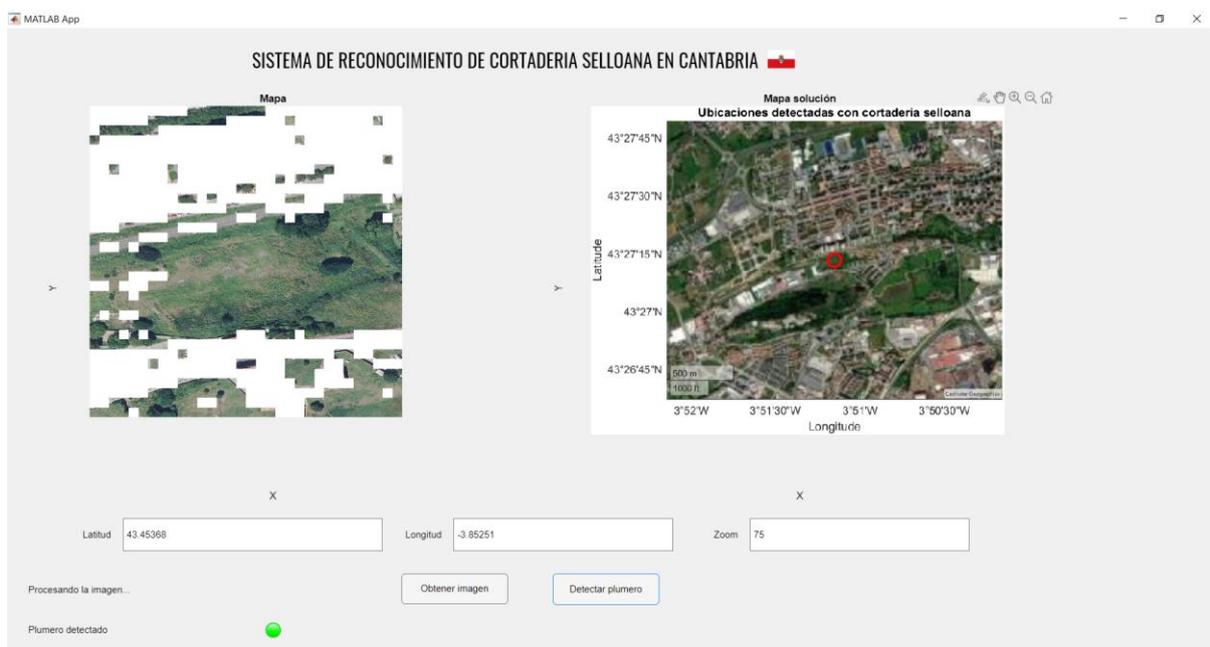


Figura 110. Representación de la segmentación de la cortadería selloana en la imagen y marcado de la ubicación en el mapa

Fuente: Elaboración propia

Tras la obtención de la primera localización, repito el proceso cargando nuevas imágenes.

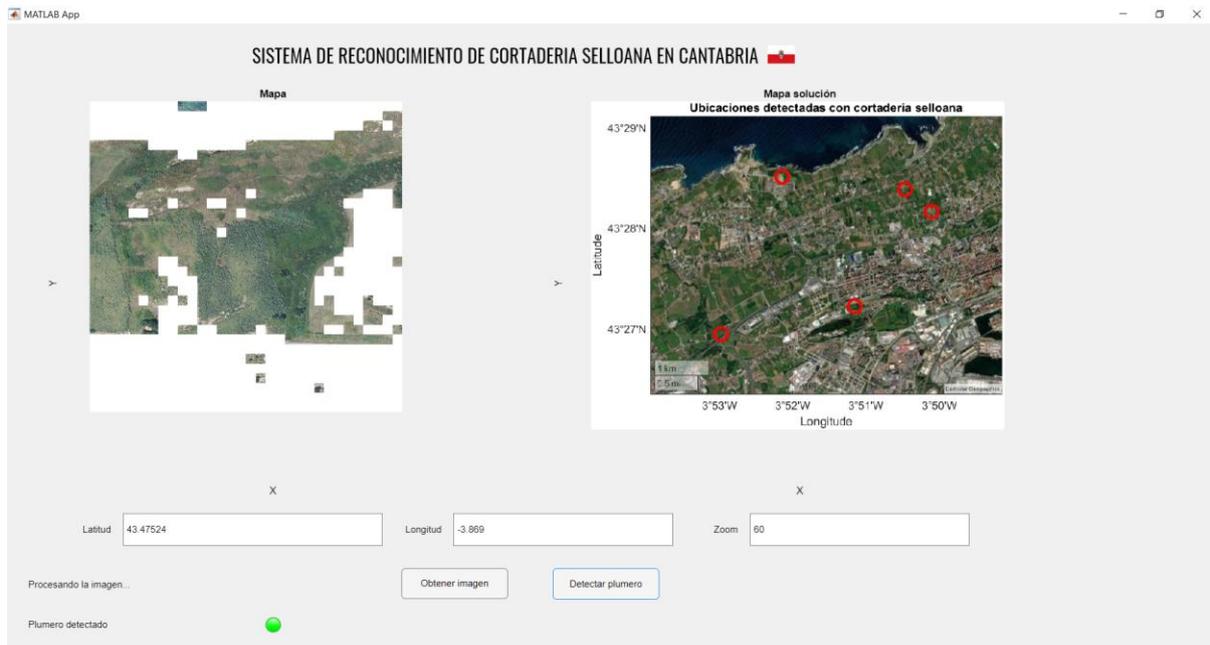


Figura 111. Segmentación de los ejemplares de plumero de la Pampa en la anterior fotografía y representación del reconocimiento en el mapa

Fuente: Elaboración propia

A medida que el número de identificaciones de cortadería selloana válidas aumenta, se va modificando el mapa con el objetivo de encuadrar todas las localizaciones en una misma representación.

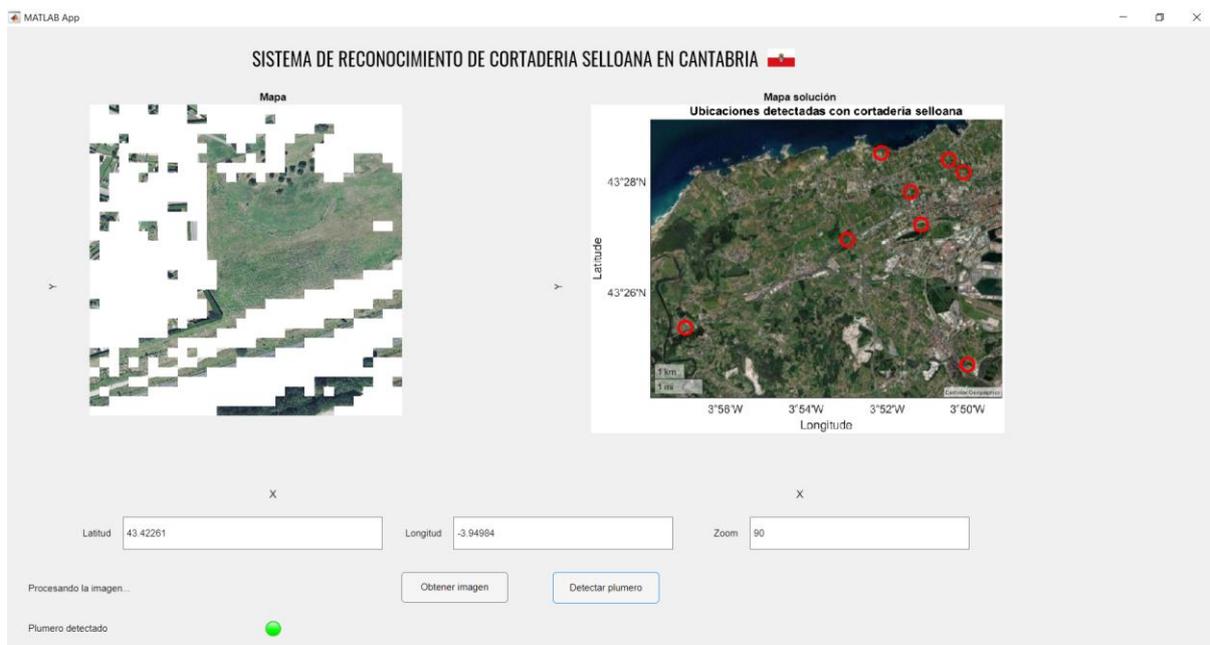


Figura 112. Segmentación de los ejemplares de plumero de la Pampa y representaciones del mapa

Fuente: Elaboración propia

En último lugar, es reseñable que, en caso de que se encuentre la planta invasora en una determinada imagen, dicha instantánea se descarga automáticamente con el zoom correspondiente. Esto ayuda a que el conjunto de datos aumente, por lo que el rendimiento de la red neuronal podría mejorar de una manera más fácil al disponer de más datos.

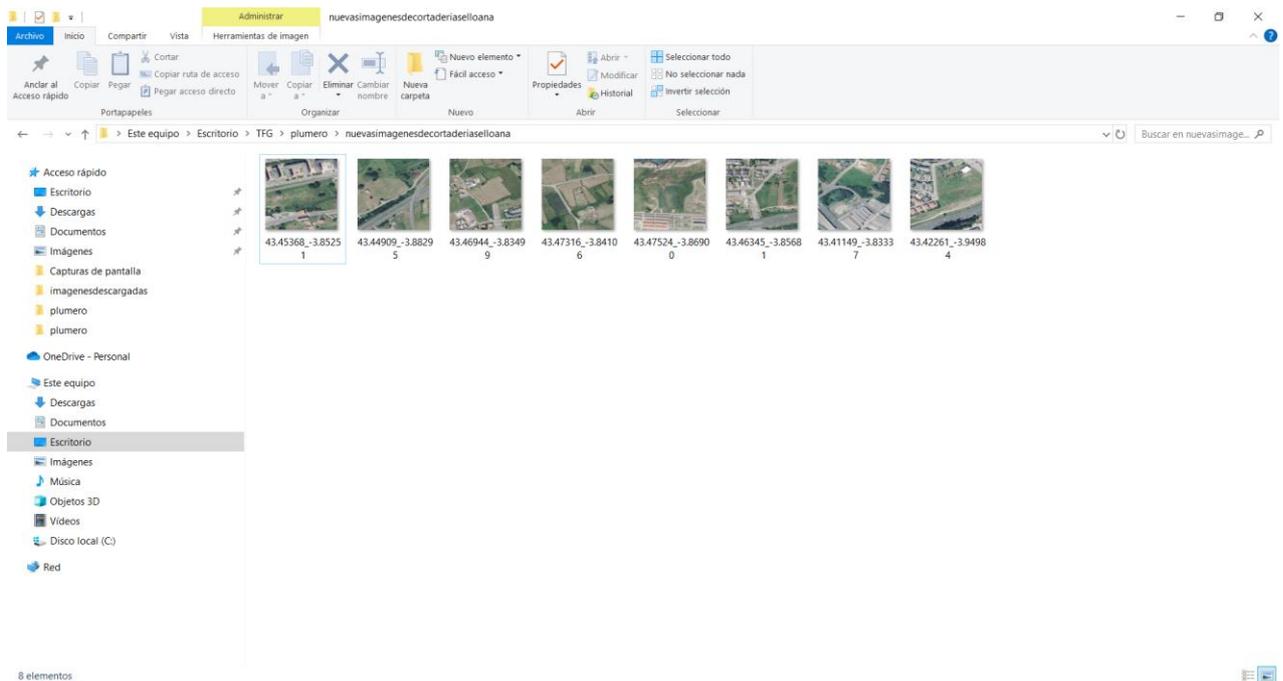


Figura 113. Conjunto de imágenes descargadas, procedentes de reconocimientos positivos

Fuente: Elaboración propia

Utilizando una zona concreta, se emplea la página web de stopcortaderia.org para valorar el rendimiento de la aplicación al identificar las plantas. Por este motivo, se busca una serie de localizaciones que estén señalizadas en un espacio no muy amplio.



Figura 114. Captura del mapa de stopcortaderia.org

Fuente: [114]

Después, se introducen los valores de las latitudes y las longitudes de cada una de esos lugares, probando diferentes valores de zoom, gracias a lo cual el reconocimiento puede perder cierta precisión en algunas imágenes. Tras escribir los diferentes valores en el programa, se va generando progresivamente el mapa correspondiente a esa zona concreta, que encaja con el mapa anterior. En el supuesto de que se produzca una ausencia de cortadería selloana en el proceso, el usuario puede continuar con sus localizaciones, sin perder el mapa que estaba construyendo hasta ese momento.



Figura 115. Respuesta del programa en caso de producirse un reconocimiento negativo

Fuente: Elaboración propia

Seleccionando las latitudes y las longitudes del resto de puntos que aparecían en el mapa de stopcortaderia.org, se comprueba el correcto funcionamiento del programa, puesto que los resultados son idénticos.

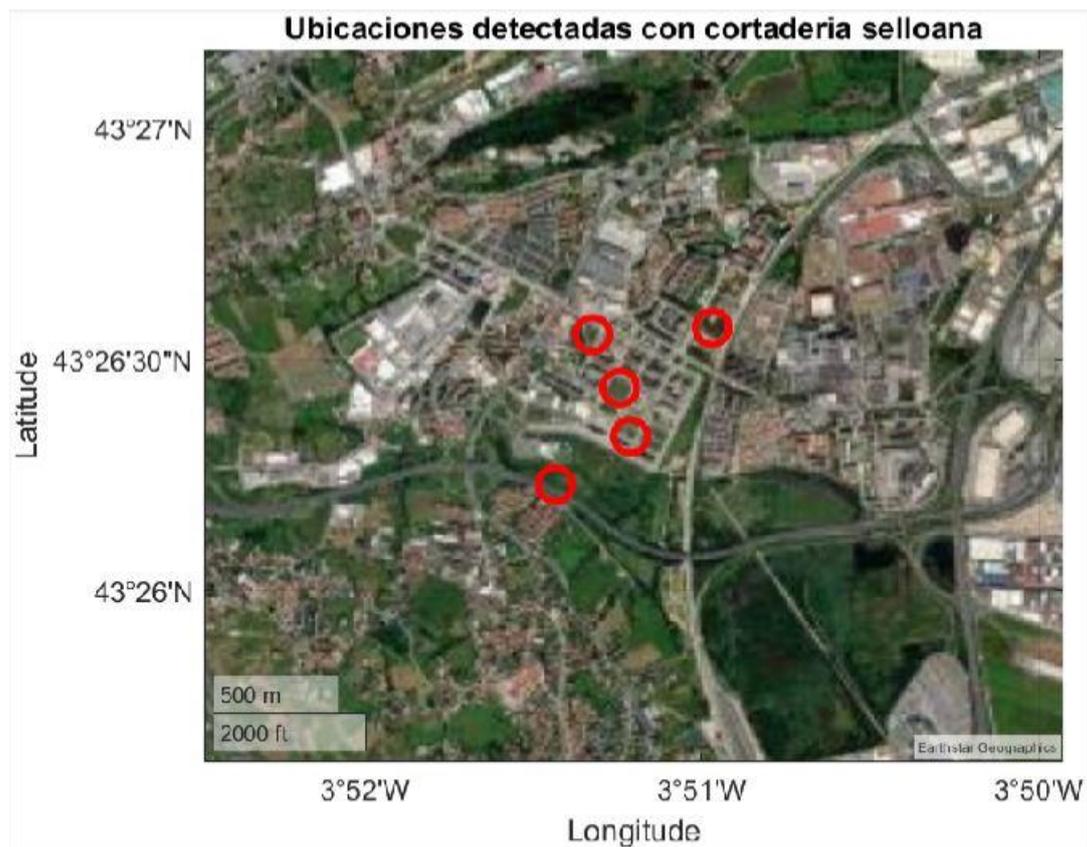


Figura 116. Captura del mapa de la aplicación tras los reconocimientos

Fuente: Elaboración propia

10.4. CONCLUSIONES

Se puede considerar que los programas software diseñados cumplen correctamente con sus cometidos, aunque la complejidad del criterio impuesto para la consideración del reconocimiento como cortadería selloana pueda provocar tanto falsos positivos como falsos negativos. Este error no es lo suficientemente importante para considerar que la solución es inválida.

La obtención de estos programas es una conclusión correcta, de acuerdo a la envergadura del proyecto realizado, con diferentes opciones de gran utilidad. En primer lugar, la posibilidad de obtener un mapa actualizado de los reconocimientos favorece a valorar el avance en los reconocimientos. En segundo lugar, es reseñable la opción por parte del usuario de descargar las imágenes analizadas, ya que enriquecen el conjunto de datos de entrenamiento. Para finalizar, la posesión de aplicaciones fáciles de utilizar ayuda al usuario a poder analizar el territorio en tiempo real cómodamente y por ende, ayuda a frenar la proliferación del plumero de la Pampa.

11. COMPARATIVA DE LOS RESULTADOS

Conociendo los resultados adquiridos, se puede obtener una comparativa entre las opciones utilizadas. De esta manera, se ve de una manera clara y sencilla la calidad de cada una de las opciones tanto en el proceso de entrenamiento como en el proceso de inferencia.

Es importante valorar que, en este caso, la tarea de reconocimiento no es fácil, lo que conlleva unos resultados peores que en otros tipos de reconocimientos utilizándose estas mismas estructuras de aprendizaje profundo.

Programa	Calidad	Precisión obtenida	Inferencia
AlexNet	Baja	Media	Nula
SVM	Media	Media/Alta	Nula
ResNet50	Alta	Alta	Correcta

Figura 117. Tabla comparativa

Fuente: Elaboración propia

12. BIBLIOGRAFÍA

[1]. Goodfellow, Ian. Bengio, Yoshua. Courville, Aaron (2016). <<Deep Learning>>. Editorial MIT Press.

[2]. Charu C. Aggarwal (2018). <<Neural Networks and Deep Learning: A Textbook>>. Editorial Springer.

[3]. Shigeo Abe (2010). <<Support Vector Machines for Pattern Classification>>. Editorial Springer.

[4] dive.tech. <<Premios Nobel en Física y Química 2024: Un Triunfo para la IA y la Ciencia Molecular>>. Fecha de acceso: 24 de abril de 2025. Disponible en:

<https://dive.tech/es/news-and-knowledge/nobel-prize-winners-in-physics-and-chemistry-2024-a-triumph-for-ai-and-molecular-science/>

[5] villaviciosa. <<Guía para el control de la cortadería selloana>>. Fecha de acceso: 2 de noviembre de 2024. Disponible en: <https://www.villaviciosa.es/plumero-de-la-pampa>

[6]. Zoumana Keita, <<Introducción a las redes neuronales convolucionales>>, 15 de abril de 2024. Fecha de acceso: 5 de noviembre de 2024. Disponible en:

https://www.datacamp.com/es/tutorial/introduction-to-convolutional-neural-networks-cnns?dc_referrer=https%3A%2F%2Fwww.google.com%2F

[7]. Google, LLC. Google Maps. Fecha de acceso: 29 de mayo. Disponible en:

<https://www.google.es/maps/preview>

- [8]. Natacha Payà, <<El “plumero de la Pampa”, un peligro para la salud y el ecosistema>>, 2 de enero de 2022. Fecha de acceso: 10 de noviembre de 2024. Disponible en: <https://www.tiempo.com/noticias/actualidad/el-plumero-de-la-pampa-peligro-para-la-salud-y-el-ecosistema.html#:~:text=El%20polen%20%22prohibido%22%20para%20los%20al%C3%A9rgicos&text=Los%20resultados%20surgieron%20fuertemente%20que,provocar%20alergias%20respiratorias%20muy%20graves.>
- [9]. Nicolas Demoulin, <<ResNet 50: un modelo preentrenado para el reconocimiento de imágenes>>, 17 de agosto de 2024. Fecha de acceso: 28 de marzo de 2025. Disponible en: <https://es.innovatiana.com/post/discover-ResNet-50>
- [10]. EITCA, <<¿Cómo optimiza el optimizador de Adam el modelo de red neuronal?>>, 8 de agosto de 2023. Fecha de acceso: 3 de abril de 2025. Disponible en: <https://es.eitca.org/artificial-intelligence/eitc-ai-dltf-deep-learning-with-tensorflow/tensorflow/neural-network-model/examination-review-neural-network-model/how-does-the-adam-optimizer-optimize-the-neural-network-model/>
- [11]. Zoumana Keita, <<Explicación de la detección de objetos YOLO>>, 29 de enero de 2024. Fecha de acceso: 26 de marzo de 2025. Disponible en: <https://www.datacamp.com/es/blog/yolo-object-detection-explained>
- [12]. Ivan Belcic, Cole Stryker, <<¿Qué es el aprendizaje supervisado?>>, 28 de diciembre de 2024. Fecha de acceso: 10 de abril de 2025. Disponible en: <https://www.ibm.com/es-es/topics/supervised-learning#:~:text=El%20aprendizaje%20supervisado%2C%20tambi%C3%A9n%20conocido, predicen%20los%20resultados%20con%20precisi%C3%B3n.>
- [13]. Enrique A. <<Detección de objetos con YOLO: implementaciones y cómo usarlas>>, 12 de mayo de 2018. Fecha de acceso: 17 de marzo de 2025. Disponible en: <https://medium.com/@enriqueav/detecci%C3%B3n-de-objetos-con-yolo-implementaciones-y-como-usarlas-c73ca2489246>
- [14]. César Chique Rodríguez. <<Maquina de soporte vectorial (SVM)>>, 3 de septiembre de 2020. Fecha de acceso: 19 de marzo de 2025. Disponible en: <https://medium.com/@csarchiquerodriguez/maquina-de-soporte-vectorial-svm-92e9f1b1b1ac>
- [15]. Jacob Murel, Eda Kavlakoglu. <<¿Qué es una matriz de confusión?>>, 19 de enero de 2024. Fecha de acceso: 10 de abril de 2025. Disponible en: <https://www.ibm.com/es-es/topics/confusion-matrix>
- [16]. Dave Bergmann, Cole Stryker. <<¿Qué es una función de pérdida?>>, 12 de julio de 2024. Fecha de acceso: 25 de marzo de 2025. Disponible en: <https://www.ibm.com/es-es/think/topics/loss-function>
- [17]. Manu Duque. <<AlexNet>>. Fecha de acceso: 26 de marzo de 2025. Disponible en: <https://www.manuque.com/enciclopedia-ia/AlexNet/>
- [18]. Vila Montserrat. <<Cortadería sellosa (pampas grass)>>, 15 de julio de 2009. Fecha de acceso: 26 de marzo de 2025. Disponible en: <https://www.cabidigitallibrary.org/doi/10.1079/cabicompendium.11872>

- [19]. Luis González. <<Visualización de NDVI en Sentinel Hub>>, 11 de diciembre de 2018. Fecha de acceso: 28 de marzo de 2025. Disponible en: <https://medium.com/luis-gonzalez/visualizaci%C3%B3n-de-ndvi-en-sentinel-hub-f022db4ae939>
- [20]. Mesh Flinders, Stephanie Susnjara, Ian Smalley. <<¿Qué es una unidad de procesamiento gráfico (GPU)?>>, 6 de enero de 2025. Fecha de acceso: 27 de marzo de 2025. Disponible en: [https://www.ibm.com/es-es/topics/gpu#:~:text=AI%20realizar%20c%C3%A1lculos%20matem%C3%A1ticos%20r%C3%A1pido%2C%20una%20GPU,la%20inteligencia%20artificial%20\(IA\)%20y%20el%20blockchain](https://www.ibm.com/es-es/topics/gpu#:~:text=AI%20realizar%20c%C3%A1lculos%20matem%C3%A1ticos%20r%C3%A1pido%2C%20una%20GPU,la%20inteligencia%20artificial%20(IA)%20y%20el%20blockchain).
- [21]. Anmol Malhotra, Rithik Agarwal, Hardey Khandhar, Rohan Mathur. <<SqueezeNet: The Key to Unlocking the Potential of Edge Computing>>, 9 de febrero de 2023. Fecha de acceso: 27 de marzo de 2025. Disponible en: <https://medium.com/sfu-csmp/SqueezeNet-the-key-to-unlocking-the-potential-of-edge-computing-c8b224d839ba>
- [22]. Universidad UNIE. <<¿Qué son las redes neuronales y cómo se aplican a la Inteligencia Artificial?>>, 9 de julio de 2024. Fecha de acceso: 31 de marzo de 2025. Disponible en: <https://www.universidadunie.com/blog/que-son-redes-neuronales#:~:text=La%20IA%20es%20el%20campo,hacer%20predicciones%20basadas%20en%20datos>.
- [23]. Jeffrey Erickson. <<¿Qué es la inferencia de IA?>>, 2 de abril de 2024. Fecha de acceso: 31 de marzo de 2025. Disponible en: <https://www.oracle.com/es/artificial-intelligence/ai-inference/>
- [24]. Sandra Navarro. <<Forward y back propagation en Deep Learning>>, 18 de abril de 2024. Fecha de acceso: 1 de abril de 2025. Disponible en: <https://keepcoding.io/blog/forward-back-propagation-deep-learning/>
- [25]. Ángel Villazón Trabanco. <<Entrenamiento de Redes Neuronales>>, 17 de marzo de 2024. Fecha de acceso: 1 de abril de 2025. Disponible en: <https://www.angelvillazon.com/inteligencia-artificial-robotica/entrenamiento-de-redes-neuronales/>
- [26]. Casco Christopher. <<Estimación adaptativa del momento: entender a Adam y utilizarlo correctamente>>, 8 de mayo de 2025. Fecha de acceso: 1 de abril de 2025. Disponible en: <https://konfuzio.com/es/estimacion-adaptativa-de-momentos/>
- [27]. Saba Hesarakí. <<SegNET>>, 20 de octubre de 2023. Fecha de acceso: 11 de abril de 2025. Disponible en: <https://medium.com/@saba99/segnet-a139ce77b570>
- [28]. Lakshmi Narayana Santha. <<Semantic Segmentation in PSPNet with implementation in Keras>>, 18 de noviembre de 2019. Fecha de acceso: 11 de abril de 2025. Disponible en: <https://medium.com/analytics-vidhya/semantic-segmentation-in-pspnet-with-implementation-in-keras-4843d05fc025>
- [29]. Kirill Bondarenko. <<Understanding UNET>>, 2 de julio de 2019. Fecha de acceso: 14 de abril de 2025. Disponible en: <https://bond-kirill-alexandrovich.medium.com/understanding-unet-27de538e08d8>

13. REFERENCIAS DE FIGURAS

Figura 1. Captura de una página web que vende ejemplares de cortadería selloana

Figura 2. Población de cortaderia selloana cercana a una carretera

Figura 3. Operarios eliminando ejemplares de plumero de la Pampa

Figura 4. Representación del Arco Atlántico invadido por la cortaderia selloana

Figura 5. Imagen del mapa de stopcortaderia.org

Figura 6. Interfaz gráfica de la web de SigPac

Figura 7. Información ofrecida en SigPac sobre un cultivo declarado

Figura 8. Fotografía aérea de una parcela con plumero obtenida mediante SigPac

Figura 9. Fotografía aérea de una parcela sin plumero mediante SigPac

Figura 10. Imagen del satélite Sentinel-2

Figura 11. Ejemplo de medición de distancias en Sentinel Hub

Figura 12. Información disponible en la web sobre el satélite Sentinel-2

Figura 13. Reconstrucción del curso del río Nilo usando imágenes de diferentes años

Figura 14. Ejemplo de clasificación de puntos mediante la tecnología LiDAR

Figura 15. Imagen de las subcategorías dentro de “Especies Invasoras”

Figura 16. Ejemplares de cortaderia selloana detectados en septiembre de 2024

Figura 17. Captura de los ejemplares de plumero de la Pampa detectados con mayor detalle

Figura 18. Localización de un ejemplar de plumero de la Pampa en stopcortaderia.org

Figura 19. Captura del cultivo en SigPac anteriormente encontrado en stopcortaderia.org

Figura 20. Representación de una neurona con sus diferentes partes señaladas

Figura 21. Representación de las distintas capas de una red neuronal

Figura 22. Representación teórica de una red neuronal artificial

Figura 23. Representación gráfica de la red neuronal AlexNet

Figura 24. Representación gráfica del funcionamiento de la vista humana

Figura 25. Representación del proceso de Dropout en una red neuronal

Figura 26. Ejemplos de GPUs de Intel

Figura 27. Representación del empleo de la tecnología YOLO en una imagen

Figura 28. Representación del funcionamiento de un algoritmo YOLO

Figura 29. Ilustración teórica del mezclado de canales

Figura 30. Ejemplos de aplicaciones de sistemas embebidos

Figura 31. Representación de un módulo Fire

Figura 32. Representación de la red neuronal resultante después del proceso de poda

Figura 33. Ilustración de la detección de objetos y personas en la conducción

Figura 34. Ejemplo del uso de SqueezeNet y YOLO en el uso correcto de mascarillas

Figura 35. Representación de un hiperplano y sus partes más características

Figura 36. Ejemplo de capas de una red residual

Figura 37. Representación de una conexión de salto

Figura 38. Modificaciones realizadas en la capa convolucional

Figura 39. Captura de las transformaciones realizadas a las imágenes

Figura 40. Captura de las gráficas del proceso de entrenamiento de la red neuronal

Figura 41. Predicción realizada por la red de una parcela sin plumero

Figura 42. Predicción realizada por la red de una parcela con plumero

Figura 43. Captura de las gráficas de rendimiento y de pérdida con la anterior configuración

Figura 44. Captura de la predicción de 2 imágenes con la configuración anterior

Figura 45. Imagen de las gráficas de rendimiento y pérdida usando esta configuración

Figura 46. Representación de imágenes predichas por la red neuronal con esta configuración

Figura 47. Gráficas de la precisión y de la pérdida de SqueezeNet con esta configuración

Figura 48. Diferentes predicciones realizadas por la red en esta simulación concreta

Figura 49. Captura de los cambios realizados en el código

Figura 50. Captura de las gráficas de precisión y de pérdida de la red neuronal de AlexNet

Figura 51. Representación de las gráficas usando la anterior configuración de imágenes

Figura 52. Captura de algunas imágenes detectadas con la anterior configuración

Figura 53. Imagen de las gráficas de precisión y pérdida usando la anterior configuración

Figura 54. Representación de algunas de las predicciones realizadas

Figura 55. Gráficas de precisión y pérdida usando la anterior configuración

Figura 56. Predicciones de la red con esta configuración, tanto de plumero como sin plumero

Figura 57. Representación de las gráficas de precisión y pérdida de la red neuronal con esta configuración

Figura 58. Representación de 4 imágenes de la base de datos con sus etiquetas teórica y predicha

Figura 59. Gráficas resultantes del anterior entrenamiento

Figura 60. Predicciones realizadas por la red al término del entrenamiento previo

Figura 61. Captura de las gráficas de precisión y de pérdida usando la anterior configuración

Figura 62. Pareja de imágenes predichas por la red neuronal

Figura 63. Descompresión del archivo de imágenes y creación del almacén de datos

Figura 64. Dos de las líneas más importantes de la red neuronal

Figura 65. Extracción de características usando ambos conjuntos y obtención de las etiquetas

Figura 66. Tabla con los resultados obtenidos del SVM

Figura 67. Representación lineal del rendimiento frente al porcentaje de datos de entrenamiento

Figura 68. Matriz de confusión obtenida con la configuración anterior

Figura 69. Captura de la carta de confusión usando la nueva configuración

Figura 70. Ejemplo de predicción incorrecta usando AlexNet y SVM

Figura 71. Ejemplo de predicción incorrecta empleando AlexNet y SVM

Figura 72. Línea del código dedicada a la adaptación de los datos

Figura 73. Ejemplo de una predicción realizada junto con los porcentajes resultantes

Figura 74. Resultado del proceso de inferencia con una imagen con plumero de la Pampa

Figura 75. Resultado de una predicción de una imagen urbana con cortadería selloana

Figura 76. Predicción de una nueva imagen en un ambiente industrial

Figura 77. Predicción de una imagen sin cortadería selloana realizada por ResNet-50

Figura 78. Predicción de una imagen sin plumero de la Pampa según ResNet-50

Figura 79. Predicción realizada en la imagen con un enfoque de 18

Figura 80. Predicción realizada en la imagen con un enfoque de 19

Figura 81. Predicción de una imagen con cortaderia selloana utilizando un zoom de 16'5

Figura 82. Predicción de una imagen con cortaderia selloana usando un zoom de 18'5

Figura 83. Imagen de una red neuronal completamente conectada

Figura 84. Representación de la arquitectura de la red neuronal U-Net

Figura 85. Representación de la arquitectura de PSPNet

Figura 86. Generación de los parámetros de la cuadrícula de la imagen

Figura 87. Comparativa entre la imagen original y la imagen con la cortaderia selloana segmentada

Figura 88. Representación conjunta de la imagen original y la imagen segmentada en clases

Figura 89. Representación del plumero de la Pampa segmentado en la imagen

Figura 90. Ajustes necesarios para obtener la imagen correcta

Figura 91. Imagen con cortaderia selloana obtenida mediante la función explicada anteriormente

Figura 92. Conversión de la longitud y de la latitud en coordenadas x e y

Figura 93. Imagen resultante del anterior código para (0,0) en latitud y longitud

Figura 94. Parte del código correspondiente a la entrada de datos

Figura 95. Parte del código correspondiente a la descarga y a la visualización de las imágenes

Figura 96. Serie de condicionales que garantizan el correcto funcionamiento del reconocimiento

Figura 97. Captura de la interfaz antes de reconocer el plumero de la imagen

Figura 98. Captura de la interfaz tras la carga de la imagen satelital

Figura 99. Procesamiento de la segmentación de la cortaderia selloana en la imagen

Figura 100. Devolución de la imagen con la segmentación de plumero de la Pampa

Figura 101. Carga de una imagen sin cortaderia selloana

Figura 102. Procesamiento de la imagen anterior

Figura 103. Reconocimiento correcto en caso de ausencia de cortadería selloana

Figura 104. Condicional donde se verifica que la localización se encuentra en Cantabria

Figura 105. Parte del código donde se determina el encendido del indicador amarillo

Figura 106. Conjunto de supuestos en los cuales se enciende el indicador verde

Figura 107. Líneas de código correspondientes al guardado de las imágenes en este programa

Figura 108. Captura del programa en caso de que la localización no pertenezca a Cantabria

Figura 109. Procesamiento de una imagen durante el transcurso del programa

Figura 110. Representación de la segmentación de la cortadería selloana en la imagen y marcado de la ubicación en el mapa

Figura 111. Segmentación de los ejemplares de plumero de la Pampa en la anterior fotografía y representación del reconocimiento en el mapa

Figura 112. Segmentación de los ejemplares de plumero de la Pampa y representaciones del mapa

Figura 113. Conjunto de imágenes descargadas, procedentes de reconocimientos positivos

Figura 114. Captura del mapa de stopcortaderia.org

Figura 115. Respuesta del programa en caso de producirse un reconocimiento negativo

Figura 116. Captura del mapa de la aplicación tras los reconocimientos

Figura 117. Tabla comparativa

Figura 1. <https://www.floresfrescaonline.com/plumeros-de-cortaderia-selloana-pack-6-uds-be51/>

Figura 2. <https://www.lne.es/asturias/2024/10/09/increible-incongruencia-politica-ministerio-aconseja-109089099.html>

Figura 3. <https://seo.org/urge-un-marco-normativo-para-cooperar-frente-al-plumero/>

Figura 4. <http://stopcortaderia.org/localizacion/>

Figura 5. <http://stopcortaderia.org/alerta/>

Figura 6. <https://sigpac.mapama.gob.es/fega/visor/>

Figura 7. <https://sigpac.mapama.gob.es/fega/visor/>

Figura 8. <https://sigpac.mapama.gob.es/fega/visor/>

Figura 9. <https://sigpac.mapama.gob.es/fega/visor/>

Figura 10. <https://es.wikipedia.org/wiki/Sentinel-2>

Figura 11. <https://apps.sentinel-hub.com/eo-browser/?zoom=6&lat=50.08245&lng=11.38184&themeld=DEFAULT-THEME#search>

Figura 12. <https://apps.sentinel-hub.com/eo-browser/?zoom=6&lat=50.08245&lng=11.38184&themeld=DEFAULT-THEME#search>

Figura 13. <https://apps.sentinel-hub.com/eo-browser/?zoom=6&lat=50.08245&lng=11.38184&themeld=DEFAULT-THEME#search>

Figura 14. <https://pnoa.ign.es/pnoa-lidar/tecnologia-lidar#:~:text=LIDAR%20es%20el%20acr%C3%B3nimo%20de,tambi%C3%A9n%20denominados%20ecos%20o%20rebotes>

Figura 15. <https://mapas.cantabria.es/>

Figura 16. <https://mapas.cantabria.es/>

Figura 17. <https://mapas.cantabria.es/>

Figura 18. <http://stopcortaderia.org/alerta/>

Figura 19. <https://sigpac.mapama.gob.es/fega/visor/>

Figura 20. <https://espanol.nichd.nih.gov/salud/temas/neuro/informacion/partes>

Figura 21. <https://futuroelectrico.com/tag/redes-neuronales-artificiales/>

Figura 23. <https://viso.ai/deep-learning/alexnet/>

Figura 24. <https://despresdelcancer.cat/index.php/tag/revisiones/?lang=es>

Figura 25. <https://www.kaggle.com/code/blurredmachine/alexnet-architecture-a-complete-guide>

Figura 26. <https://www.techspot.com/article/2125-intel-last-graphics-card/>

Figura 27. <https://viso.ai/deep-learning/yolov7-guide/>

Figura 28. <https://blog.naver.com/sogangori/220993971883>

Figura 29. <https://medium.com/syncedreview/shufflenet-an-extremely-efficient-convolutional-neural-network-for-mobile-devices-72c6f5b01651>

Figura 30. <https://www1.utec.edu.pe/blog-de-carreras/ingenieria-electronica/sabias-que-los-sistemas-embadidos-estan-en-casi-todo-lo-que-usamos>

Figura 31. <https://medium.com/data-science/review-squeezenet-image-classification-e7414825581a>

Figura 32. <https://konfuzio.com/es/optimizacion-de-modelos-ai/>

Figura 33. <https://ignaciogavilan.com/la-percepcion-en-vehiculos-autonomos/>

Figura 34. <https://www.mdpi.com/2504-2289/6/4/106>

Figura 35. <https://brainscience518485518.wordpress.com/2018/08/12/toda-la-verdad-sobre-la-mentira/>

Figura 36. https://datascience.eu/es/aprendizaje-automatico/una-vision-general-de-resnet-y-sus-variantes/#google_vignette

Figura 37. <https://chautuankien.medium.com/skip-connection-and-explanation-of-resnet-afabe792346c>

Figura 83. <https://medium.com/swlh/fully-connected-vs-convolutional-neural-networks-813ca7bc6ee5>

Figura 84. <https://bond-kirill-alexandrovich.medium.com/understanding-unet-27de538e08d8>

Figura 85. https://www.researchgate.net/figure/The-basic-structure-of-PSPNet_fig3_351104978

Figura 114. <http://stopcortaderia.org/alerta/>