



universidad  
de león



# Escuela de Ingenierías Industrial, Informática y Aeroespacial

## GRADO EN INGENIERÍA INFORMÁTICA

Trabajo de Fin de Grado

Reconocimiento de imágenes obtenidas con dron  
mediante técnicas de Deep Learning

Recognition of images obtained with drone using Deep  
Learning techniques

Autor: Daniel Fernández Alonso  
Tutor: María Teresa García Ordás y Javier Fernández Lozano

(Febrero, 2022)

**UNIVERSIDAD DE LEÓN**  
**Escuela de Ingenierías Industrial, Informática y**  
**Aeroespacial**

**GRADO EN INGENIERÍA INFORMÁTICA**  
**Trabajo de Fin de Grado**

**ALUMNO:** Daniel Fernández Alonso

**TUTOR:** María Teresa García Ordás || Javier Fernández Lozano

**TÍTULO:** Reconocimiento de imágenes obtenidas con dron mediante técnicas de Deep Learning

**TITLE:** Recognition of images obtained with drone using Deep Learning techniques

**CONVOCATORIA:** Febrero, 2022

**RESUMEN:**

En este trabajo se plantea el uso de técnicas de **Deep Learning** en combinación con los **drones** para la identificación de distintos patrones sobre las imágenes capturadas. Para ello, se ha desarrollado una **aplicación web**, accesible para que cualquier usuario, que permite identificar sobre las imágenes los patrones que mediante **redes neuronales convolucionales** puede reconocer el sistema. El funcionamiento consistirá en que el usuario registrado, pueda introducir sus imágenes y buscar una serie de patrones con los que ha sido entrenado el modelo de red neuronal, para clasificar imágenes.

El uso de dron permitió la captura de imágenes en los que quedan representados distintos patrones, para posteriormente crear el conjunto de datos, divididos en subconjuntos, que fueron introducidos en la red neuronal convolucional para su entrenamiento, como **cruces de caminos**, y elementos arqueológicos, como **canales para el transporte de agua** hacia las antiguas explotaciones mineras romanas, y las propias **minas de oro romanas**. El estudio se ha realizado en la provincia de León, España.

**ABSTRACT:**

Through the use of **Deep Learning** techniques working together, with technologies such as **drones**, we are going to identify different patterns within the captured images. For this purpose, a **web application** will be built, accessible so that any user can identify in their images the patterns that this system will recognize through convolutional neural networks. The operation will consist in that any user who is registered, can enter their images and test if they contain any of the patterns with which we have trained the neural network model, functioning as an image classifier.

With the drone we are going to take the images of different patterns, to later create the data set that we will introduce to the convolutional neural network,

such as crossroads and archaeological sites, which will be water transport channels to the old Roman mining operations, and finally the Roman gold mines themselves. This data set will be carried out in the province of León, Spain.

**Palabras clave:** Deep Learning, Dron, web application, dron recognition

**Firma del alumno:**

**VºBº Tutor/es:**

## Índice de contenidos

1	Introducción .....	12
1.1	Objetivos .....	12
2	Trabajos previos .....	15
3	Drones .....	16
3.1	Características del dron. ....	16
3.2	Toma de imágenes.....	17
3.2.1	Vuelo 1: Patrón de cruce de caminos.....	18
3.2.2	Vuelo 2: Patrón de canales de agua romanos. ....	19
3.2.3	Vuelo 3 Patrón de explotación minera de oro romana.....	20
4	Patrimonio.....	22
4.1	Canales de agua .....	22
4.2	Minas auríferas .....	23
5	Redes Neuronales Convolucionales .....	25
5.1	¿Qué son las redes neuronales convolucionales? .....	25
5.2	¿Cómo funcionan? .....	25
6	Conjuntos de datos .....	30
7	Implementación de red neuronal .....	34
7.1	Fase de entrenamiento.....	34
7.1.1	Creación y compilación del modelo .....	34
7.1.2	Carga de imágenes .....	34
7.1.3	Entrenamiento, test y guardado del modelo .....	34
7.1.4	Aumento de datos .....	35
7.2	Fase de predicción .....	36
7.2.1	Carga del modelo.....	36
7.2.2	Carga de la imagen y predicción.....	36
8	Aplicación Web.....	38
8.1	Interfaz del usuario .....	39
8.1.1	Portada .....	39
8.1.2	Inicio de sesión .....	40
8.1.3	Registro de nuevo usuario.....	41
8.1.4	Perfil.....	42

8.1.4	Cambio de contraseña.....	43
8.1.5	Actualización del perfil .....	43
8.1.6	Contacto .....	44
8.1.7	Subida y comprobación de imágenes.....	45
8.1.8	Enrutamiento con VUE .....	45
8.1.5	Menú de navegación.....	46
8.1.6	Pie de página .....	48
8.1.7	Mis imágenes .....	48
8.2	Parte del servidor.....	49
8.2.1	Conexión a base de datos y modelos tablas. ....	49
8.2.2	Subida de imágenes.....	49
8.2.3	Envío de correos electrónicos .....	50
8.2.4	Comprobación de patrones.....	51
9	Optimización de motores de búsqueda (SEO) .....	52
9.1	¿Qué es SEO? .....	52
9.2	SEO en la web .....	52
9.3	Informe SEO .....	53
10	Control de versiones .....	57
10.1	Seguimiento de la red neuronal .....	57
10.2	Seguimiento de la aplicación web .....	58
11	Resultados.....	60
11.1	Entrenamiento inicial.....	61
11.2	Mejoras y correcciones del entrenamiento inicial. ....	62
11.3	Solución de errores de predicción .....	65
12	Conclusiones y recomendaciones .....	67
12.1	Trabajo realizado .....	67
12.2	Trabajos futuros.....	67
12.3	Problemas encontrados.....	67
12.4	Opinión personal.....	68
13	Análisis de prototipo. ....	69
13.1	Presupuesto .....	69
13.2	Análisis de mercado .....	70
13.2.1	Análisis de empresas .....	70

13.2.2	Comparativa con el prototipo .....	71
13.2.3	Análisis de costes.....	71
14	Manual del usuario.....	72
14.1	Página de login.....	74
14.2	Página de perfil .....	76
14.3	Página de Mis Imágenes .....	77
14.4	Página de Cambio de contraseña .....	78
14.5	Página de actualización de datos personales .....	79
14.6	Página de contacto .....	79
15	Agradecimientos .....	81
16	Bibliografía .....	82

## Índice de figuras

Figura 1.1 Ejemplo de la zona de estudio donde se presentan caminos y cruces en el área de Villimer.....	13
Figura 3.1 Imagen del dron con el que se han tomado las imágenes .....	16
Figura 3.2 Logotipo de la aplicación utilizada para el vuelo del dron. ....	17
Figura 3.3 Vuelo autónomo de rejilla. ....	18
Figura 3.4 Picado en la roca para la construcción del canal de transporte de agua en Corporales.....	19
Figura 3.5 Forma de las minas auríferas romanas.....	20
Figura 4.1 Canales existentes en la cabrera leonesa. (Fuente: <a href="http://canalesromanos.es/canales.html">http://canalesromanos.es/canales.html</a> ).....	22
Figura 4.2 Picado en el canal de transporte de agua. ....	23
Figura 4.3 Mina romana aurífera de Castrocontrigo.....	24
Figura 5.1 Obtención del mapa de características inicial (Fuente: <a href="https://www.juanbarrios.com/redes-neurales-convolucionales/#Definiciones">https://www.juanbarrios.com/redes-neurales-convolucionales/#Definiciones</a> ).....	26
Figura 5.2 Separación de la imagen de RGB, en 3 mapas de características. (Fuente: <a href="https://www.juanbarrios.com/redes-neurales-convolucionales/#Definiciones">https://www.juanbarrios.com/redes-neurales-convolucionales/#Definiciones</a> ).....	26
Figura 5.3 Filtro de la convolución, y su paso por la imagen original (Fuente: <a href="https://www.juanbarrios.com/redes-neurales-convolucionales/#Definiciones">https://www.juanbarrios.com/redes-neurales-convolucionales/#Definiciones</a> ).....	27
Figura 5.4 Ejemplo de pooling del mapa de características (Fuente: <a href="https://www.juanbarrios.com/redes-neurales-convolucionales/#Definiciones">https://www.juanbarrios.com/redes-neurales-convolucionales/#Definiciones</a> ).....	28
Figura 5.5 Reducción de los mapas de características a un vector unidimensional. (Fuente: <a href="https://frogames.es/la-guia-definitiva-de-las-redes-neuronales-convolucionales-3-4/">https://frogames.es/la-guia-definitiva-de-las-redes-neuronales-convolucionales-3-4/</a> ) .....	28
Figura 5.6 Ejemplo de una red neuronal convolucional completa (Fuente: <a href="https://riull.ull.es/xmlui/bitstream/handle/915/10422/Usando%20redes%20neuronales%20convolucionales%20para%20convertir%20caracteristicas%20visuales%20en%20estimulos%20sonoros.pdf?sequence=1&amp;isA">https://riull.ull.es/xmlui/bitstream/handle/915/10422/Usando%20redes%20neuronales%20convolucionales%20para%20convertir%20caracteristicas%20visuales%20en%20estimulos%20sonoros.pdf?sequence=1&amp;isA</a> ) .....	29
Figura 6.1 Separación del dataset en directorios .....	31
Figura 6.2 Ejemplo de imágenes de cruces de nuestro conjunto de datos personalizado.31	
Figura 6.3 Ejemplo de imágenes de sin cruce del conjunto de datos .....	32
Figura 6.4 Ejemplo de las imágenes de los canales de transporte de agua hacia explotaciones mineras romanas.....	32
Figura 6.5 Ejemplo de las imágenes de las minas de oro romanas de Castrocontrigo .....	33
Figura 7.1 Visualización del aumento de datos (Fuente: <a href="https://enmilocalfunciona.io/tratamiento-de-imagenes-usando-imagedatagenerator-en-keras/">https://enmilocalfunciona.io/tratamiento-de-imagenes-usando-imagedatagenerator-en-keras/</a> ).....	36
Figura 7.2 Ejemplo de redimensión y obtención de los 3 canales de entrada a la red neuronal convolucional (Fuente: <a href="https://anderfernandez.com/blog/que-es-una-red-neuronal-convolucional-y-como-crearla-en-keras/">https://anderfernandez.com/blog/que-es-una-red-neuronal-convolucional-y-como-crearla-en-keras/</a> ) .....	37

Figura 8.1 Logotipos de los lenguajes usados en la aplicación web (Fuente: <a href="https://grokonez.com/frontend/vue-js/vue-js-nodejs-express-restapis-sequelize-orm-mysql-crud-example">https://grokonez.com/frontend/vue-js/vue-js-nodejs-express-restapis-sequelize-orm-mysql-crud-example</a> ) .....	38
Figura 8.2 Explicación de cliente servidor con las tecnologías usadas.....	39
Figura 8.3 Muestra de los patrones reconocidos en la pantalla de inicio.....	40
Figura 8.4 Muestra del inicio de sesión de esta aplicación web .....	41
Figura 8.5 Pantalla de registro de un nuevo usuario.....	42
Figura 8.6 Pantalla del perfil del usuario. ....	42
Figura 8.7 Interfaz de cambio de contraseña, para el usuario. ....	43
Figura 8.8 Interfaz para la actualización de los datos del usuario .....	44
Figura 8.9 Pantalla de contacto de la web, para notificar errores, mejoras, etc. ....	44
Figura 8.10 Interfaz para la subida y comprobación de las imágenes. ....	45
Figura 8.11 Fragmento del .htaccess para que sean visibles todas las rutas de la aplicación. (Fuente: <a href="https://router.vuejs.org/guide/essentials/history-mode.html#example-server-configurations">https://router.vuejs.org/guide/essentials/history-mode.html#example-server-configurations</a> ) .....	46
Figura 8.12 Menú sin iniciar sesión con la cuenta de usuario.....	46
Figura 8.13 Menú con la sesión iniciada.....	47
Figura 8.14 Menú visto en dispositivo móvil, diseño responsive.....	47
Figura 8.15 Pie de página de la web. ....	48
Figura 8.16 Interfaz de mis imágenes, propia de cada usuario.....	48
Figura 8.17 ORM de Sequelize con conexión a base de datos MYSQL mediante Node (Fuente: <a href="https://medium.com/@ahmedcharef/sequelize-orm-with-node-js-koa-mysql-foreign-keys-a3097ab91a8">https://medium.com/@ahmedcharef/sequelize-orm-with-node-js-koa-mysql-foreign-keys-a3097ab91a8</a> ).....	49
Figura 8.18 Icono de la librería utilizada para envío de correos. (Fuente: <a href="https://nodemailer.com/about/">https://nodemailer.com/about/</a> ) .....	50
Figura 8.19 Aplicaciones poco seguras para permitir el envío de correos.....	51
Figura 9.1 Infografía sobre el contenido del SEO (Fuente: <a href="https://marketing4food.com/como-encontrar-palabras-clave-para-seo/">https://marketing4food.com/como-encontrar-palabras-clave-para-seo/</a> ).....	52
Figura 9.2 Reporte de SEO de Lighthouse. ....	54
Figura 9.3 Reporte de la accesibilidad de la aplicación. ....	55
Figura 9.4 Reporte de buenas prácticas. ....	56
Figura 10.1 Control de versiones GitHub usado en el proyecto (Fuente: <a href="https://medium.com/swlh/how-to-auto-sync-update-from-one-github-repository-to-other-repository-using-github-workflow-6492f4830d45">https://medium.com/swlh/how-to-auto-sync-update-from-one-github-repository-to-other-repository-using-github-workflow-6492f4830d45</a> ) .....	57
Figura 10.2 Seguimiento de commits de la red neuronal. ....	58
Figura 10.3 Seguimiento de las ramas de la red neuronal. ....	58
Figura 11.1 Red convolucional planteada al inicio del proyecto.....	60
Figura 11.2 Gráfica de entrenamiento de la red donde podemos ver la precisión y la pérdida de nuestro modelo.....	61
Figura 11.3 Ejemplo de error a la hora de predecir un nuevo elemento en una imagen..	62
Figura 11.4 Nueva red neuronal convolucional, mejorada para la solución del problema	63
Figura 11.5 Gráfica del entrenamiento de 1000 periodos con loss y accuracy .....	64
Figura 11.6 Imagen de demostración de corrección de la clase a una correcta .....	65



Figura 14.1	Página principal de la aplicación web.....	72
Figura 14.2	Segunda parte de la página principal. ....	73
Figura 14.3	Fin de la página principal. ....	73
Figura 14.4	Interfaz de inicio de sesión .....	74
Figura 14.5	Alerta de ¿Has olvidado tu contraseña?.....	75
Figura 14.6	Interfaz del registro del usuario. ....	75
Figura 14.7	Interfaz de perfil del usuario. ....	76
Figura 14.8	Muestra de interfaz de Mis Imágenes.....	77
Figura 14.9	Menús y botones de la interfaz de Mis Imágenes.....	78
Figura 14.10	Interfaz de cambio de contraseña. ....	78
Figura 14.11	Interfaz de actualización de los datos personales.....	79
Figura 14.12	Página de contacto con la empresa.....	80

## Índice de tablas

Tabla 3.1 Características del dron .....	16
Tabla 3.2 Datos de vuelo de caminos de Villimer. ....	18
Tabla 3.3 Datos de vuelo de canales de agua romanos de Corporales.....	20
Tabla 3.4 Datos de vuelo de minas de oro romanas de Castrocontrigo. ....	21
Tabla 6.1 Separación de las imágenes en las diferentes clases .....	30
Tabla 8.1 Tabla de campos del registro de usuario.....	41
Tabla 11.1 Aumento y reestructuración del conjunto de datos. ....	65
Tabla 13.1 Costes hardware del proyecto.....	69
Tabla 13.2 Costes de personal del proyecto. ....	69
Tabla 13.3 Otros costes relacionados con el proyecto .....	69
Tabla 13.4 Coste global del proyecto .....	70
Tabla 13.5 Gastos de producción del prototipo.....	71

## Glosario

**Dataset:** Es el conjunto de datos que se utilizará para entrenar la red neuronal.

**Kernel:** Es el filtro que se aplicará en las capas de convolución o de reducción para la búsqueda de características.

**Pooling:** Capas de reducción de las características de la convolución.

**CNN:** Sigla de las redes neuronales convolucionales, en inglés Convolutional Neural Network.

**TensorFlow y Keras:** librerías de Deep Learning del lenguaje de programación de Python.

**ORM:** modelo de programación que nos permite el mapeo de las distintas tablas de bases de datos relacionales.

**Responsive:** En diseño web es considerado un sinónimo de diseño adaptable, es decir, que se puede ver en cualquier dispositivo independientemente de su pantalla.

**SPA:** aplicaciones de una sola página, como las que se pueden crear con frameworks como Vue, React, etc.

**Mapa de características:** imágenes generadas tras el proceso de convolución.

**Frontend:** Parte visual de la aplicación web o página web, lo que el usuario puede ver.

**Backend:** Es la parte del servidor de cualquier aplicación web.

**Accuracy:** Es la función que sirve para medir el número de casos en que el modelo acierta.

**Loss:** Es la función de pérdida utilizada para saber cuánto se desvía un elemento de la clase original.

**Error 404:** Es un código de error, el cual nos indica que una página web no se puede encontrar en la ruta especificada o que no existe.

**.htaccess:** Es un fichero con mandamientos para el servidor Apache. Este sirve para optimizar la carga del sitio web, mejorar las redirecciones, etc.

**Commits:** nombres que se da a las entradas en el controlador de versiones de git.

## 1 Introducción

Con el auge de las nuevas tecnologías, como los drones y la visión por computador, podemos ver algunos de los usos que se les está dando. Por una parte, los drones se están usando cada vez más para realizar trabajos técnicos de precisión, acceder a lugares de difícil acceso que de otro modo serían inaccesibles o remotos. Algunos campos de aplicación de esta tecnología pueden ser la topografía, minería, arquitectura y patrimonio o, la agricultura de precisión, entre otros muchos. Por otro lado, la visión por computador que cada vez está más instaurada en un gran número de ámbitos de actuación científica y civil para distintos usos, como pueden ser la seguridad, clasificación de imágenes, reconocimientos de patrones.

### 1.1 Objetivos

Para este proyecto se ha combinado ambas tecnologías con el fin de generar una aplicación web automatizada para el reconocimiento de patrones en imágenes aéreas. Esta aplicación puede ser útil para profesionales de distintos ámbitos como son los investigadores, las administraciones públicas, los técnicos y el público en general.

En este proyecto, se van a detectar tres patrones en imágenes aéreas, las cuales serán para la planificación del territorio de cruces de caminos y, para el estudio del patrimonio, los canales de transporte de agua hacia explotaciones auríferas romanas y, las propias explotaciones mineras.

Todo el conjunto de datos de este trabajo se ha tomado con un dron dentro de la provincia de León en las zonas rurales de Villimer, Corporales y Castrocontigo.

El DeepLearning en este estudio aportará la identificación de los patrones sobre estas imágenes. Además, se plantea la detección de elementos antiguos mineros de patrimonio por sus estructuras lineales, que podrían ser confusos en imágenes aéreas con los caminos o cruces de estos. Todo ello aportará una mayor fiabilidad a la hora de reconocer los patrones en imágenes que no se hayan visto durante el proceso de entrenamiento de la red.



*Figura 1.1 Ejemplo de la zona de estudio donde se presentan caminos y cruces en el área de Villimer.*

Tras la realización de estos tres vuelos vamos a tener que reenumerar las imágenes para evitar solapamiento de nombres y posteriormente separarlas en los diferentes patrones. Esto ayudará a crear nuestro conjunto de datos personalizado, que más tarde tendremos que dárselo a nuestra red neuronal convolucional para que logre aprender las características.

Desde el punto de vista informático, se implementará una aplicación web con su correspondiente servicio Restful API. De esta forma, todos los usuarios registrados en el servicio podrán tener acceso al reconocimiento de sus imágenes para saber si tienen o no esos patrones, siempre mostrándoselo por pantalla. Para desarrollar la técnica de Deep Learning se implementará una red neuronal convolucional (CNN) con el objetivo de reconocer los patrones en las imágenes. Esta red va a ser desarrollada desde cero para este proyecto, creando un modelo de red que vamos a tener que ajustar hasta dar con el punto exacto que permita obtener buenos resultados a la hora de clasificar las imágenes que se introduzcan en la web.

Por lo tanto, el objetivo principal de este proyecto, es la implementación de un modelo de Deep Learning para el reconocimiento de patrones dentro de las imágenes aéreas. De este objetivo principal se derivan:

1. Creación del conjunto de datos: se realizarán vuelos con dron para la obtención de las imágenes con las que crear el dataset para entrenar a la red neuronal convolucional.
2. Implementación de la red: Tras la creación del conjunto de datos se desarrollará el modelo de la red neuronal, con sus fases de entrenamiento y test.
3. Desarrollo de aplicación web: Con el fin de que este modelo de red neuronal sea lo más accesible posible, se pretende crear una aplicación web.

## 2 Trabajos previos

En visión por computador tenemos varios trabajos de investigación de las redes neuronales convolucionales y Deep Learning como pueden ser los siguientes que se van a exponer:

En el ámbito de la medicina se encuentra AlphaFold2 [1], un sistema capaz de resolver el plegamiento de las proteínas siendo un sistema bastante eficiente para el estudio de las mismas. En este trabajo se emplean las redes neuronales convolucionales para resolver el problema del plegamiento, tratándose de imágenes de las proteínas.

En el campo de la seguridad, nos encontramos con un sistema de reconocimiento facial en personas [2]. Con este sistema se pueden reconocer las caras estimando su edad y género. Se pudo ver como haciendo varios grupos de edades, se consiguieron distintos resultados entre géneros. En este caso de estudio se obtuvieron resultados bastante altos de entre un 70% y 82%, aunque en algunos casos fallase entre cinco y seis años de estimación de edad. Para este caso se emplearon redes neuronales convolucionales un poco más avanzadas que las del proyecto que se realiza en el presente trabajo.

Otros artículos de investigación con cierta relevancia que unen las dos tecnologías (drones e inteligencia artificial) son:

Estudios para emergencias sanitarias [3] en la cual se ha utilizado Python para conseguir que el dron vuele desde una posición origen hasta un destino utilizando el sistema de posicionamiento GPS. Con esto y la cámara del dron, utilizando Deep learning, se pueden utilizar los drones para emergencias sanitarias. Este sistema se basa en la detección de objetos de forma rápida y todo procesado en el mismo dron.

El uso de drones también se ha implementado para reconocimiento de imágenes aéreas[4]. En este trabajo comentan las diferentes arquitecturas y en concreto, la de redes neuronales convolucionales, además de algunas más avanzadas. En el reconocimiento de objetos que no sean rígidos comentan que es más complicado que se detecten bien.

El empleo de reconocimiento de objetos mediante los drones [5] , especialmente humanos en bosques, utilizan diferentes técnicas de aprendizaje profundo como pueden ser las redes convolucionales para diferenciar un objeto de otro. Gracias a unas cámaras colocadas en los drones y, a estos sistemas inteligentes, podemos identificar personas en el bosque.

### 3 Drones

#### 3.1 Características del dron.

El dron utilizado para este proyecto es un vehículo de ala rotatoria que tiene un despegue vertical y que cuenta con cuatro rotores. La aeronave está controlada de forma remota o totalmente autónoma para los distintos tipos de vuelos que se pueden realizar, equipado con una cámara fotográfica RGB como podemos ver en la **Figura 3.1**. Además, es un dron tipo Phantom 4 de DJI cuyas características principales pueden consultarse en la **Tabla 3.1**.



*Figura 3.1 Imagen del dron con el que se han tomado las imágenes*

Peso	Longitud de alas	Tipo de ala	Autonomía	Tipo de cámara
1 388 g	350 mm	Rotatoria	30 min	RGB

*Tabla 3.1 Características del dron*



### 3.2 Toma de imágenes

Para la captura de imágenes se realizaron tres vuelos, uno para cada patrón que se quería detectar con este sistema inteligente y cuya descripción, se realizará en los siguientes puntos.

Con la intención de realizar estos vuelos se ha empleado la aplicación móvil Pix4DCapture, la cual nos permite seleccionar el tipo de vuelo que deseamos hacer. Con esta aplicación, podemos ver en tiempo real la cámara del dron, independientemente del tipo de vuelo que estemos realizando.

Antes de comenzar los vuelos de captura del dataset, tenemos que indicar qué tipo de vuelo se utilizará, cada cuanto se tomará la imagen y, ajustar diferentes parámetros, entre ellos la altura y la velocidad de vuelo.

En un vuelo autónomo tenemos que indicarle: la altura de vuelo para que sea constante, la velocidad de vuelo, cada cuantos metros se tomará la imagen y, el área delimitada, como podemos ver en la **Figura 3.2**.



*Figura 3.2 Logotipo de la aplicación utilizada para el vuelo del dron.*

En la **Figura 3.3** podemos ver el área delimitada en el que se va a volar, donde está el principio y el fin de esta rejilla, y según está configurado, volaría a una altura constante de 50 metros según el modelo de elevación del terreno.

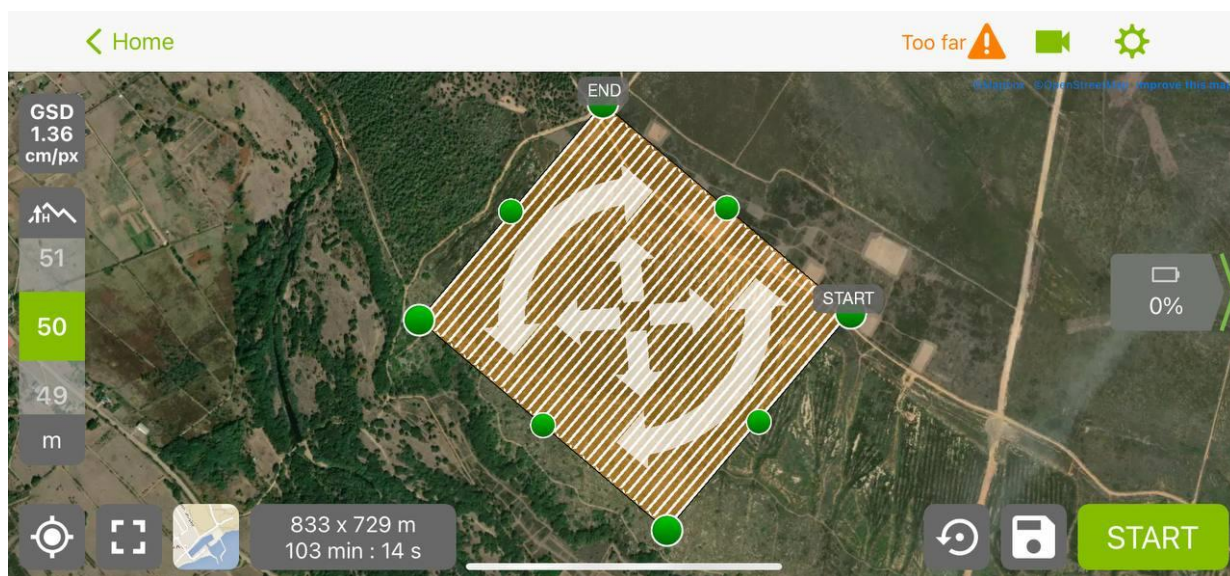


Figura 3.3 Vuelo autónomo de rejilla.

En conclusión, el uso de esta aplicación permite la captura de imágenes de una forma automática y sistemática para poder tener un mayor rango de cobertura del objeto que deseamos estudiar.

### 3.2.1 Vuelo 1: Patrón de cruce de caminos.

En este vuelo realizado el 11 de noviembre de 2020, en la zona de Villimer (León) se obtuvieron 2209 imágenes con cruces y sin cruces de caminos, volando el dron a alturas de 55 y 100 metros de forma autónoma. Además, se hizo un vuelo de forma manual a distintas alturas.

En la **Tabla 3.2**, podemos ver los datos de vuelo, la superficie cubierta, el número total de imágenes, las coordenadas GPS de la ubicación seleccionada y, el tiempo de vuelo.

Tiempo de vuelo	Superficie	Número de imágenes	Coordenadas GPS
4h 30min	0.643 km <sup>2</sup> / 64.2681 ha / 0.25 sq. mi. / 158.8921 acres	2209	[ 42.58996922 , -5.40431961 ]

Tabla 3.2 Datos de vuelo de caminos de Villimer.

### 3.2.2 Vuelo 2: Patrón de canales de agua romanos.

Este vuelo fue realizado el 8 de marzo de 2021, en la zona de Corporales, perteneciente al ayuntamiento de Truchas, en León. Aquí se obtuvieron 1646 imágenes de los canales de transporte de agua hasta las explotaciones mineras romanas. Para la toma de imágenes se utilizó el vuelo manual, recorriendo estos canales y capturando las imágenes desde distintas perspectivas.

Estos canales tienen un ancho de 1.20 metros de ancho, y en las curvas aumenta para reducir la velocidad del agua. A parte con este sistema, estos canales, tienen una inclinación menor al 1% del desnivel. Con estos 2 sistemas explicados, los romanos querían que el agua no tuviese mucha fuerza al llegar a las zonas de lavado donde les podría arrastrar todo el mineral sedimentado.

En la **Figura 3.4** en la parte baja se observan las marcas de picado realizadas por los romanos para la construcción del canal, indicativo de su antigüedad y uso.



*Figura 3.4 Picado en la roca para la construcción del canal de transporte de agua en Corporales.*

En la **Tabla 3.3**, podemos ver los datos de vuelo, la superficie cubierta, el número total de imágenes, las coordenadas GPS de la ubicación seleccionada, y el tiempo de vuelo.

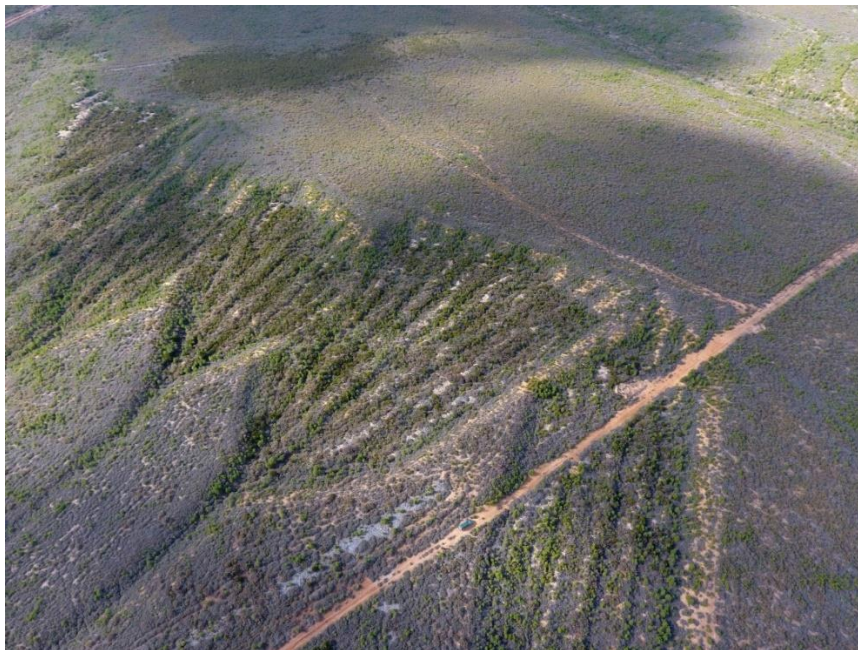
Tiempo de vuelo	Superficie	Número de imágenes	Coordenadas GPS
2h	0.064 km <sup>2</sup> / 6.3723 ha / 0.02 sq. mi. / 15.7545 acres	1646	[ 42.30921656 , -6.46551792]

*Tabla 3.3 Datos de vuelo de canales de agua romanos de Corporales*

### 3.2.3 Vuelo 3 Patrón de explotación minera de oro romana.

En el vuelo de la explotación minera realizado el 8 de marzo del 2021, en la zona de Castrocontrigo (León) se compone de 2082 imágenes obtenidas con un vuelo manual formando círculos realizándose a una altura adecuada para capturar todas las minas de oro.

En estas minas, se puede ver una estructura similar a una escoba, donde los romanos iban refinando el material. En las zonas más alejadas, era un material más grueso, y en la zona en la que convergen estas, denominada agoga, era una zona de lavado donde ya podría obtener el oro más fino.



*Figura 3.5 Forma de las minas auríferas romanas*

En la **Tabla 3.4**, podemos ver los datos de vuelo, la superficie cubierta, el número total de imágenes, las coordenadas GPS de la ubicación seleccionada, y el tiempo de vuelo.

Tiempo de vuelo	Superficie	Número de imágenes	Coordenadas GPS
2h	1.520 km <sup>2</sup> / 152.0437 ha / 0.59 sq. mi. / 375.9026 acres	2082	[ 42.19963375 , -6.19946847]

*Tabla 3.4 Datos de vuelo de minas de oro romanas de Castrocontrigo.*

## 4 Patrimonio

### 4.1 Canales de agua

En la situación geográfica de León se pueden encontrar un gran conjunto de canales que construyeron los romanos para transportar el agua hasta las explotaciones mineras, para la realización del lavado del oro en ellas.

En la **Figura 4.1** podemos ver los canales existentes en la zona de “La Cabrera”.



*Figura 4.1 Canales existentes en la Cabrera leonesa. (Fuente: <http://canalesromanos.es/canales.html> )*

Como se puede ver en la **Figura 4.2**, los romanos picaban la roca para la construcción de estos canales.



*Figura 4.2 Picado en el canal de transporte de agua.*

Estos canales tenían un ancho de 90 cm, que aumentaban hasta 1.2 metros en las curvas para reducir la velocidad del agua. A parte de estos, tenían un desnivel menor a un 1% con el mismo fin que el aumento del ancho de estos canales.

#### 4.2 Minas auríferas

Dentro de la provincia de León nos encontramos una gran cantidad de explotaciones mineras de oro que dejaron los romanos.

Pero en este caso de estudio vamos a emplear las minas auríferas a cielo abierto. En este tipo de minas, los romanos iban refinando el material de más grueso a más fino con ayuda del agua.

Estas minas o zonas de lavado de oro, como la que se puede ver en la **Figura 4.3**, podemos ver tres sectores, las cuales se van a explicar, de acuerdo a Fernández-Lozano et al [X].



*Figura 4.3 Mina romana aurífera de Castrocontrigo.*

En la parte superior se puede ver la zona por donde introducían el agua para la limpieza del material.

La zona intermedia o parte considerada “la escoba” es donde va clasificando el material de más grueso a más fino.

Por último, la zona donde convergen todos estos canales de la escoba es donde realizaban el bateo del oro, dado que es la parte donde ya se encuentra el material más fino.



## 5 Redes Neuronales Convolucionales

### 5.1 ¿Qué son las redes neuronales convolucionales?

Este tipo de redes neuronales están encapsuladas en el Deep Learning y forman parte de la visión por computador.

Esta red neuronal sirve para obtener patrones dentro de grandes dataset de imágenes. De esta manera, se obtienen características de cada imagen pudiendo generalizar en el patrón o forma que se desee obtener.

### 5.2 ¿Cómo funcionan?

Este tipo de redes neuronales funcionan con múltiples capas convolucionales, en las cuales se van aumentando el número de mapas de características y reduciendo la dimensión de la imagen.

Las capas convolucionales trabajan con distintos tipos de filtros, que pueden hacer distintas variaciones a la imagen, como pueden ser desenfocarla, diferencia de contrastes para encontrar bordes, etc. Sin embargo, todos estos filtros no se configuran de forma manual, sino que es la propia red neuronal convolucional, la que va a ir ajustando hasta llegar al valor más óptimo a la hora de reconocer los patrones.

En la **Figura. 5.1**, se puede ver la imagen inicial. Esta es pasada a la red neuronal convolucional, pero primero sufrirá una normalización de los colores de sus píxeles que pasa a ser de 0 a 255 a ser entre 0 y 1.



Una imagen...

		0.6	0.6	
	0.6			0.6
	0.6	0.6	0.6	0.6
	0.6			0.6

...es una matriz de pixeles.

El valor de los pixeles va de 0 a 255 pero se normaliza para la red neuronal de 0 a 1

Figura 5.1 Obtención del mapa de características inicial (Fuente: <https://www.juanbarrios.com/redes-neurales-convolucionales/#Definiciones> )

Si a la red le introduce una imagen en RGB, significa que tiene tres canales. Esta imagen va a ser separada, como si fuesen tres imágenes completamente diferentes a la hora de pasar por las capas convolucionales, aportando mapas de características, que van a ser diferentes, aunque con similitudes, este caso podemos verlo en la **Fig. 5.2**.

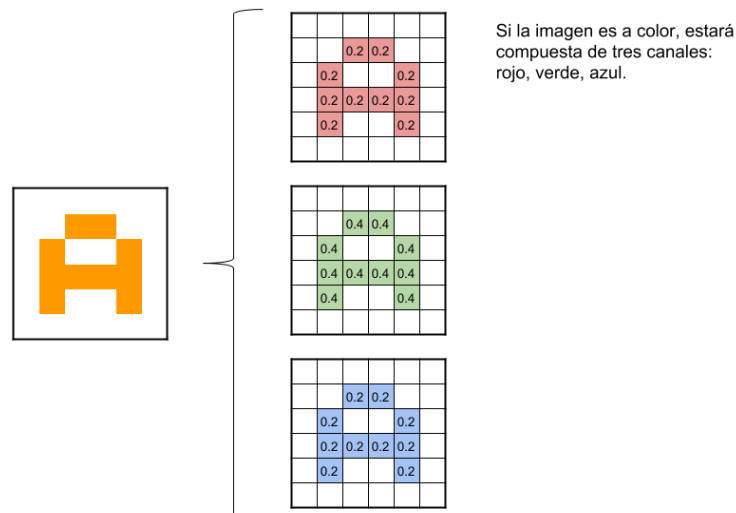
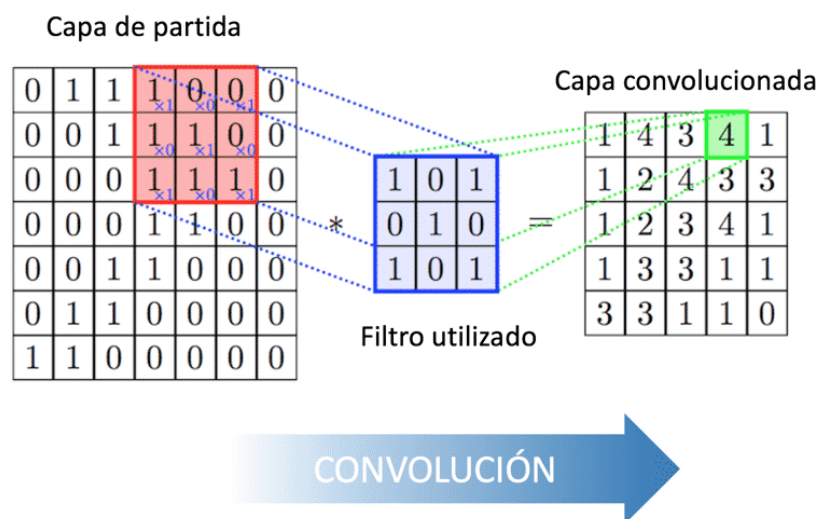


Figura 5.2 Separación de la imagen de RGB, en 3 mapas de características. (Fuente: <https://www.juanbarrios.com/redes-neurales-convolucionales/#Definiciones> )

En la **Figura 5.3**, se puede ver que la capa de partida es la imagen original. El kernel (filtro utilizado en la figura) que se está utilizando es una matriz de tamaño 3 x 3.

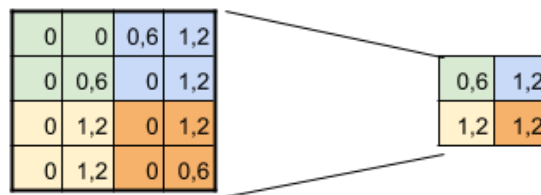
Este kernel se va pasando por los píxeles de la imagen original, donde cada valor de salida, es el correspondiente a la multiplicación de pixel por valor correspondiente de kernel y después se suman estos valores. De tal forma que van saliendo tantos mapas de características como neuronas tenga la capa convolucional, que posteriormente será reducida.



*Figura 5.3 Filtro de la convolución, y su paso por la imagen original (Fuente: <https://www.juanbarrios.com/redes-neurales-convolucionales/#Definiciones> )*

Tras haber realizado la operación de convolución, los mapas de características obtenidos se tienen que reducir. En este caso concreto, será eligiendo el máximo de cada kernel que se defina. Un ejemplo de esta explicación es **Figura 5.4**, con un kernel de tamaño 2 x 2.

Después de realizar estos 2 pasos de convolución y pooling, se puede repetir varias veces logrando que patrones simples, tras cada capa de convolución, sean más complejos, ya que las nuevas capas trabajaran con los mapas de características de salida de las capas anteriores.



**SUBSAMPLING:**  
 Aplico Max-Pooling de 2x2  
 y reduzco mi salida a la mitad

Figura 5.4 Ejemplo de pooling del mapa de características (Fuente: <https://www.juanbarrios.com/redes-neurales-convolucionales/#Definiciones> )

Al terminar las convoluciones, se reducirán todos los mapas de características, para que quede como una única entrada para la red neuronal multicapa, a esto se le denomina “flatening”. Esto sirve para reducir la dimensionalidad, como se puede observar en la **Figura 5.5**.

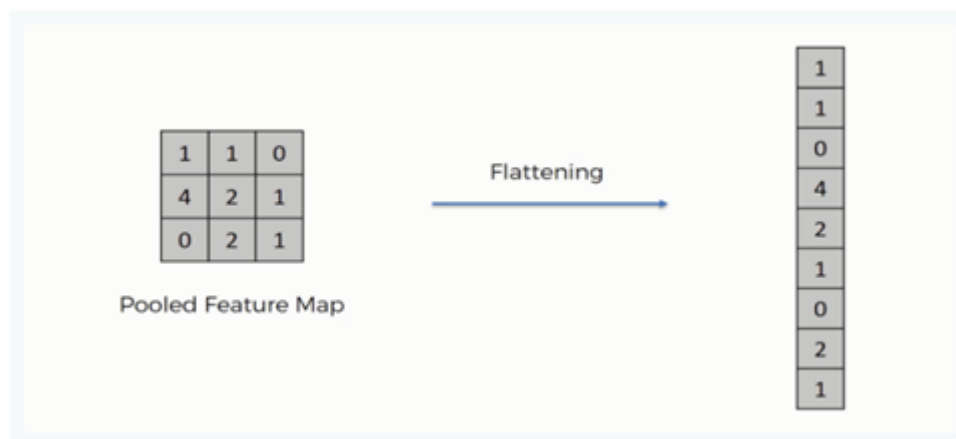


Figura 5.5 Reducción de los mapas de características a un vector unidimensional. (Fuente: <https://froqames.es/la-qui-definitiva-de-las-redes-neurales-convolucionales-3-4/> )

El resultado que se obtiene es el que se introduzca a la red neuronal multicapa, la cual en su capa de salida tendrá tantas neuronas como patrones queramos detectar. En el estudio

a realizar serán cuatro elementos a identificar y tendrá cuatro neuronas. La estructura final de la red será algo similar a la **Figura 5.6**

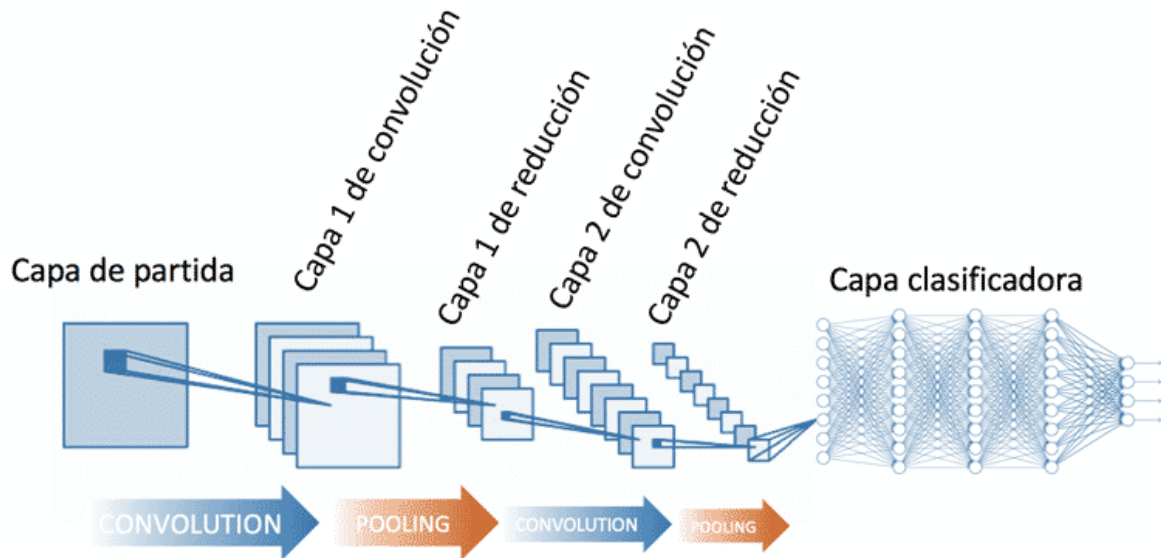


Figura 5.6 Ejemplo de una red neuronal convolucional completa (Fuente: <https://riull.ull.es/xmlui/bitstream/handle/915/10422/Usando%20redes%20neuronales%20convolucionales%20para%20convertir%20caracteristicas%20visuales%20en%20estimulos%20sonoros.pdf?sequence=1&isA>)

## 6 Conjuntos de datos

Gracias a las imágenes obtenidas anteriormente con los distintos vuelos del dron, se ha podido construir el dataset. Al tratarse de un aprendizaje supervisado hay que dividir el dataset en dos subconjuntos, entrenamiento y test, con la distribución de **Tabla 6.1**.

Categoría	Entrenan	Test	Total
Canales	1035	611	1646
Cruces	899	518	1417
Minas	1500	582	2082
No cruces	527	220	747
	Total		5892

*Tabla 6.1 Separación de las imágenes en las diferentes clases*

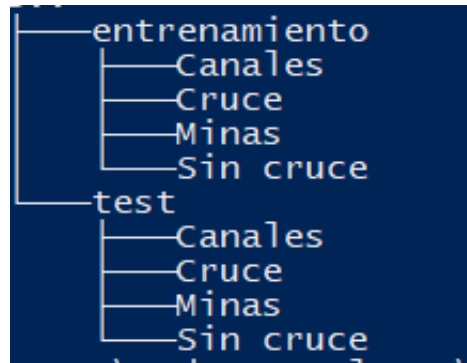
Para todo esto vamos a crear dos directorios base, entrenamiento y test. Dentro de estos mismos crearemos los subdirectorios correspondientes a las categorías a reconocer.

Con tal de evitar problemas de que dos imágenes tengan el mismo nombre, lo cambiaremos todas de forma automatizada con un script de Python. Este cogerá todas las imágenes de un directorio y las numerará antes de introducirlas a nuestro nuevo dataset por correspondiente categoría.

La división de categorías será de la siguiente forma:

- Cruce de caminos
- Sin cruces
- Canales de transporte de agua romanos
- Explotaciones mineras auríferas romanas.

Esta separación comentada se puede ver en la **Figura 6.1**



*Figura 6.1 Separación del dataset en directorios*

Los dos primeros patrones corresponden al mismo vuelo (como se indica en el apartado de drones). En estas imágenes se han tenido que separar las que contenían cruces de las que no tenían cruce, denominadas: sin cruce.

En las **Figura 6.3**, se pueden ver los distintos tipos de cruces de caminos que se han detectado en las imágenes aéreas tomadas.



*Figura 6.2 Ejemplo de imágenes de cruces de nuestro conjunto de datos personalizado.*

En las imágenes denominadas sin cruces son todas caminos y la mayoría de ellos son tramos rectos como se observa en la **Figura 6.3**.



*Figura 6.3 Ejemplo de imágenes de sin cruce del conjunto de datos*

Dentro de la **Figura 6.4**, se pueden ver algunos ejemplos de los canales de transporte de agua que los romanos utilizaban para llevar el agua hasta las explotaciones mineras. Estos canales son de la zona de Corporales.



*Figura 6.4 Ejemplo de las imágenes de los canales de transporte de agua hacia explotaciones mineras romanas*

En el último vuelo, en la zona de Castrocontrigo, donde las explotaciones mineras de oro romanas, se pueden observar las estructuras de surcos convergentes que se asemejan a escobas. En la **Figura 6.5** se puede apreciar perfectamente.





*Figura 6.5 Ejemplo de las imágenes de las minas de oro romanas de Castrocontrigo*

## 7 Implementación de red neuronal

### 7.1 Fase de entrenamiento

#### 7.1.1 Creación y compilación del modelo

Para la implementación de la red neuronal se han usado las librerías de Python, Keras que funcionan sobre la librería de más bajo nivel de TensorFlow. Se han empleado la mayoría de las funciones de estas librerías para el entrenamiento o predicción los elementos de la red.

La construcción de la red neuronal convolucional o CNN por sus siglas en inglés, se han utilizado varias capas de convolución y de “pooling”, para reducir la dimensionalidad de los mapas de características, teniendo un kernel de 3x3 en la capa convolucional y de 2x2 en la de “pooling”.

Tras estas capas, se reducirán los mapas de características a uno solo por imagen, para tener una sola entrada válida hacia nuestra red neuronal multicapa. En esta se van a desactivar la mitad de las neuronas en cada paso para evitar el overfitting o sobreajuste de nuestra red, obligándola a generalizar los resultados.

Una vez se haya construido la red neuronal por completo, se tiene que compilar para comprobar que todo está correcto.

#### 7.1.2 Carga de imágenes

A la hora de cargar el enorme conjunto de datos para evitar el colapso de la memoria RAM del ordenador se ha tenido que usar el ImageDataGenerator. Con esto se consigue poder cargar por lotes de 64 las imágenes a procesar. Además, se ha usado para aumentar el conjunto de datos realizando distintas transformaciones como pueden ser rotar, invertir vertical y horizontalmente, y normalizar los colores entre 0 y 1 (los colores que aparecen en la imagen).

A parte de las transformaciones comentadas, también redimensionará en memoria las imágenes, pasando de un tamaño de 4000 x 3000 píxeles a 150 x 150 píxeles.

#### 7.1.3 Entrenamiento, test y guardado del modelo

En este paso se introduce el tensor creado para la carga de imágenes del punto anterior a la red, indicándole el número de períodos y el número de pasos para cada período. Aquí

tardará bastante en procesar todo, dado que se tiene un gran conjunto de datos del que obtener características para entrenar la red.

Al acabar este entrenamiento, se guardarán los datos producidos de “accuracy” y “loss” obtenidos, de cada período de entrenamiento y los producidos en el test en un archivo txt.

Se guarda el modelo de red con todos los pesos en un fichero .h5. Con esto se puede más tarde cargarlo y predecir nuevas imágenes. Este modelo es guardado en un nuevo directorio nombrado: modelo. Si este ya existe no tenemos que hacer nada, dado que se guardará el solo. Sin embargo, sino se encuentra este directorio se creará y almacenará en estos dos ficheros.

#### 7.1.4 Aumento de datos

El aumento de datos sirve para aumentar el número de casos de ejemplo para que la red se entrene con muchos más casos. Se va a emplear esta técnica antes de pasarle el dataset a la red.

Esto nos sirve para voltear la imagen ya sea horizontalmente o verticalmente, rotarla o, realizar oclusión de cierta parte de la misma. Con esto también normalizaremos los colores de la imagen entrante de 1-255 entre los valores de 0 y 1.

Como podemos ver en la **Figura 7.1**, de una única imagen podemos obtener nueve diferentes. En ese caso concreto, podrían ser muchas más si invirtiésemos la imagen en vertical.

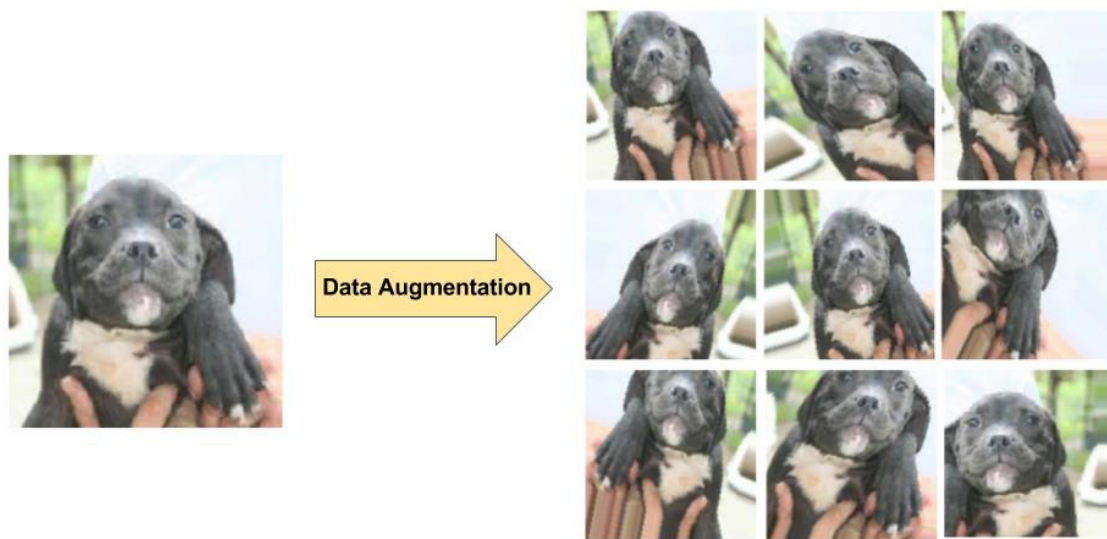


Figura 7.1 Visualización del aumento de datos (Fuente: <https://enmilocalfunciona.io/tratamiento-de-imagenes-usando-imagegenerator-en-keras/>)

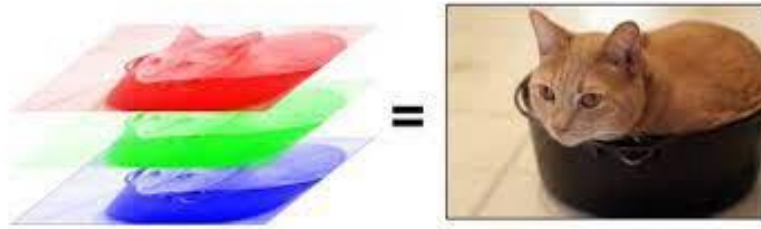
## 7.2 Fase de predicción

### 7.2.1 Carga del modelo

Para este apartado, se van a utilizar los ficheros generados en el entrenamiento, donde se han guardado los pesos de las diferentes neuronas y el modelo de la red. Si no se encuentran estos ficheros saltará un error, en el que dirá que no existen los ficheros que se están buscando.

### 7.2.2 Carga de la imagen y predicción.

A la hora de predecir la clase a la que pertenece la imagen introducida, se preprocesará primero la imagen, dándole un tamaño de 150 x 150 píxeles, y una forma de tres canales, como si fuese una imagen RGB como la **Figura 7.2**. Tras este paso, ya se puede introducir la imagen a la red neuronal convolucional, y que la procese. A partir de ahí, devolverá un array de salida con los porcentajes a los que pertenecen las etiquetas aportadas en el entrenamiento.



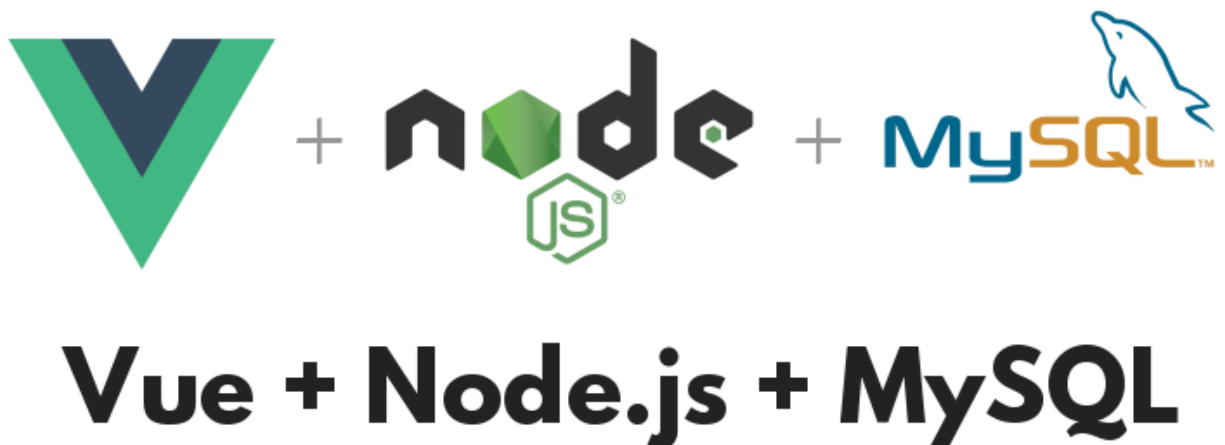
*Figura 7.2 Ejemplo de redimensión y obtención de los 3 canales de entrada a la red neuronal convolucional (Fuente: <https://anderfernandez.com/blog/que-es-una-red-neuronal-convolucional-y-como-crearla-en-keras/>)*

Para finalizar este paso, se encapsula la imagen, como un JSON, donde se introduce la imagen, y se determinará a qué clase pertenece la misma.

## 8 Aplicación Web

La aplicación web se implementará usando las siguientes tecnologías. En la parte de Frontend el framework de Javascript de Vuejs, en la que engloba los tres elementos en un fichero: HTML, CSS, y JavaScript. Esta tecnología crea webs de una única página trabajando los cambios de interfaz el controlador de JavaScript.

La parte del Backend, se va a trabajar con Nodejs mediante el ORM de Sequelize que conectará con la base de datos MySQL.



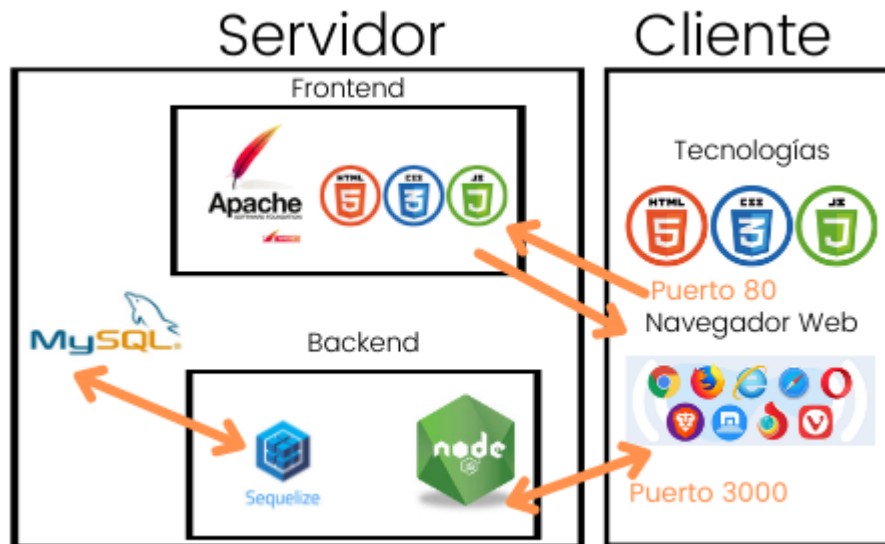
*Figura 8.1 Logotipos de los lenguajes usados en la aplicación web (Fuente: <https://grokonez.com/frontend/vue-js/vue-js-nodejs-express-restapis-sequelize-orm-mysql-crud-example> )*

Para la puesta en marcha de esta aplicación, y poder funcionar de forma correcta cuando este en producción se empleará la arquitectura cliente-servidor, donde se utilizarán dos servidores:

- Apache: que gestionará las peticiones del usuario, devolviendo los ficheros necesarios para la correcta visualización de la aplicación web, como son los ficheros HTML, CSS y JavaScript. Este servidor trabajará sobre el puerto 80.
- Node: este servidor controlara las peticiones que haga el usuario a la hora de navegar por la aplicación web tratando los accesos a la base de datos, la subida de imágenes, y la comprobación de las imágenes contra la red neuronal convolucional. Todas estas peticiones serán enviadas al usuario en formato estandarizado JSON para así trabajar de una forma más fácil y eficiente en el período del desarrollo. Este

servidor también se puede llamar servicio RestFull API o API. Se encontrará funcionando en el puerto 3000.

Esta explicación comentada se puede ver de forma más visual en la **Figura. 8.2**.



*Figura 8.2 Explicación de cliente servidor con las tecnologías usadas.*

## 8.1 Interfaz del usuario

En este punto se especifican las diferentes interfaces con las que cuenta la aplicación así como sus funcionalidades y requisitos. Toda la interfaz de usuario está orientada a ser visible en cualquier dispositivo con conexión a internet teniendo así, un diseño completamente adaptable o responsive.

Por otro lado, las peticiones realizadas hacia el servicio Rest desde la interfaz, se han realizado mediante la librería de axios, permitiendo hacer peticiones de cuatro tipos sobre el protocolo http como son get, post, put y delete.

### 8.1.1 Portada

La portada formará parte de la página principal que cualquier usuario verá cuando acceda al servicio. Desde esta interfaz, mediante una barra superior de menú, puede acceder a todo el mapa de la web con ciertas restricciones que posteriormente se comentarán.

Además, desde el pie de página se tendrán referencias a las páginas que siempre serán visibles para cualquier usuario que acceda a esta web.



*Figura 8.3 Muestra de los patrones reconocidos en la pantalla de inicio.*

Aquí, se ve una pequeña parte de la portada. En ella indicamos al usuario qué patrones se reconocen, y también si quiere acceder a reconocer sus imágenes. A este último punto, el usuario tiene que iniciar sesión para poder tener acceso.

### 8.1.2 Inicio de sesión

Desde esta interfaz cualquier usuario que se encuentre registrado podrá acceder a todos los contenidos si hace el inicio de sesión de forma correcta. Si se dejan todos los campos vacíos, o se introducen mal los campos de correo y contraseña, este inicio de sesión resultará inválido, y saltará un mensaje de error, para que el usuario se dé cuenta.





Figura 8.4 Muestra del inicio de sesión de esta aplicación web

### 8.1.3 Registro de nuevo usuario

La interfaz básica desde la cual un nuevo usuario de la aplicación se registrará para poder acceder a todos los servicios que se ofrecen a través de esta web.

Aquí, se le pedirá al usuario los datos de nombre, apellidos, contraseña, correo electrónico, y su teléfono móvil. De estos campos comentados, son todos obligatorios, excepto el teléfono y los apellidos. Estas restricciones se pueden ver en la **Tabla 8.1**

Campo	Tipo de dato	Requerido
Nombre	Cadena de caracteres	Sí
Apellidos	Cadena de caracteres	No
Contraseña	Cadena de caracteres	Sí
Correo electrónico	Cadena de caracteres	Sí
Teléfono	Numérico	No

Tabla 8.1 Tabla de campos del registro de usuario.

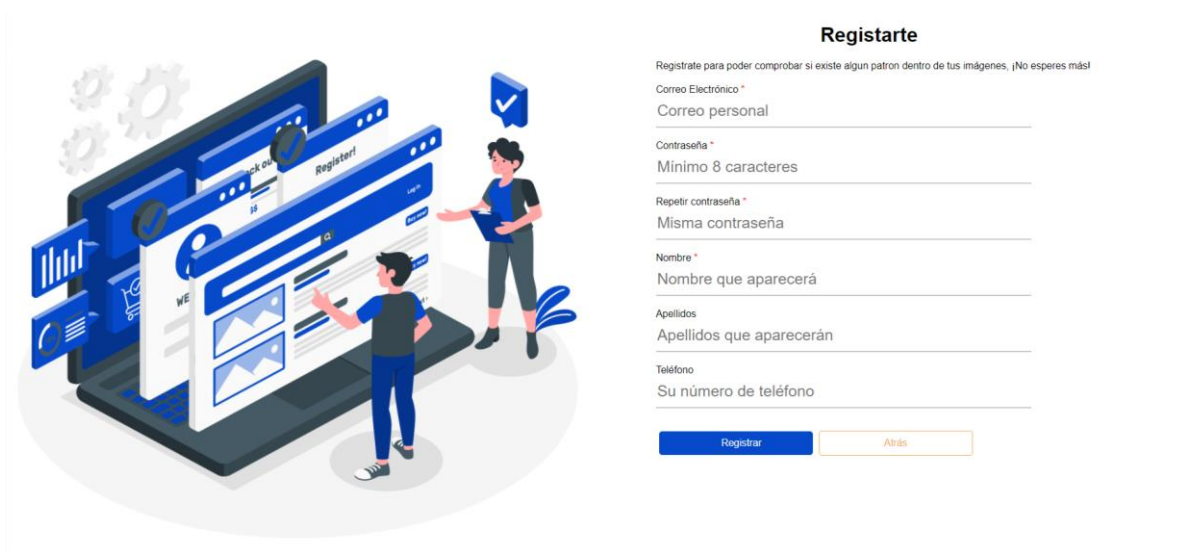


Figura 8.5 Pantalla de registro de un nuevo usuario

#### 8.1.4 Perfil

Esta es una página sobre la cual el usuario va a poder visualizar y modificar sus datos personales, pudiendo modificar la contraseña para el acceso a este servicio. Solo podrán acceder si han iniciado sesión.

Desde aquí no se permite al usuario eliminar su cuenta, solo se le permite darse de baja.

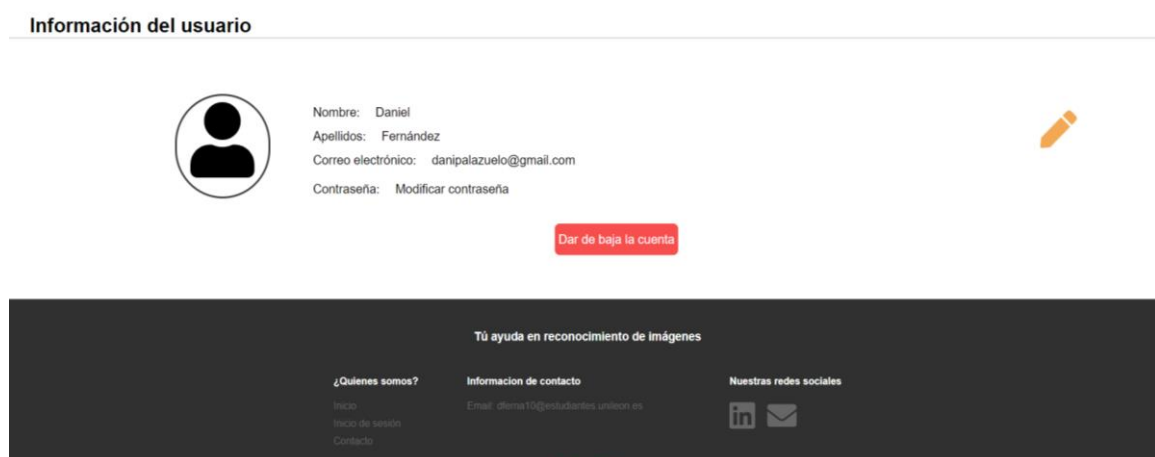


Figura 8.6 Pantalla del perfil del usuario.

#### 8.1.4 Cambio de contraseña

Esta es una de las interfaces que se van a poder acceder desde dos puntos de vista. Por un lado, se accederá desde el perfil de usuario para poder modificar nuestra contraseña. Para ello, se deberá tener una sesión iniciada. Por otro, cuando se olvida la contraseña, al realizar la petición de cambio, desde la interfaz de inicio de sesión.

The image shows a web form for changing a password. The form is titled "Cambia tu contraseña" and contains two input fields: "Contraseña:" and "Repite tu contraseña:". Below the fields are two buttons: "Actualizar" (Update) and "Cancelar" (Cancel). Below the form is a dark footer with the text "Tú ayuda en reconocimiento de imágenes" and three columns of links: "¿Quiénes somos?", "Información de contacto" (with email "Email: dforma10@estudiantes.unibcn.es"), and "Nuestras redes sociales" (with LinkedIn and email icons).

Figura 8.7 Interfaz de cambio de contraseña, para el usuario.

#### 8.1.5 Actualización del perfil

A esta vista solo se puede acceder si el usuario se encuentra con una sesión iniciada. Esta interfaz sirve para modificar alguno de sus datos personales, y solo se puede acceder desde la vista anteriormente comentada de perfil del usuario.

Desde aquí se podrá modificar: nombre, apellidos, correo electrónico y teléfono.

**Actualiza tus datos personales**

Nombre: Daniel

Apellidos: Fernández

Correo electrónico: correo@gmail.com

Teléfono: 666999333

Actualizar
Cancelar

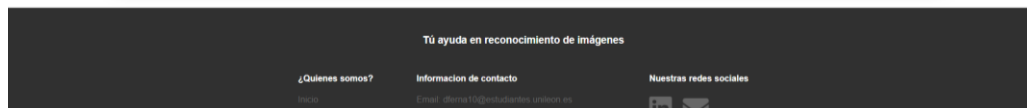


Figura 8.8 Interfaz para la actualización de los datos del usuario

### 8.1.6 Contacto

Es una interfaz estática y bastante simple para que cualquier usuario pueda ponerse en contacto con este servicio con la intención de resolver sus dudas, informar sobre nuevos patrones para que el servicio se mejore, posibles incidencias en el funcionamiento de la aplicación, etc.

**No dudes y ponte en contacto con nosotros**

Si tienes alguno de los siguientes puntos, o alguna duda sobre nosotros, o nuestra aplicación ponte en contactos con nosotros, intentaremos responderte en el menor tiempo posible

- Necesitas registrarte
- Tienes dudas acerca de la aplicación
- Has tenido algún problema a utilizar este servicio
- ¿Quieres más información?
- ¿Quieres hacer una sugerencia?

✉ Correo electrónico: [dferna10@estudiantes.unileon.es](mailto:dferna10@estudiantes.unileon.es)

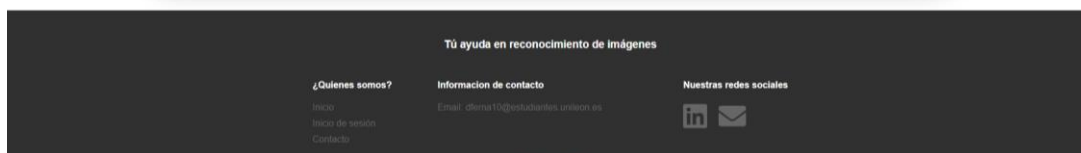


Figura 8.9 Pantalla de contacto de la web, para notificar errores, mejoras, etc.

### 8.1.7 Subida y comprobación de imágenes

Es la interfaz más importante de la aplicación donde solo se accede si el usuario ha iniciado la sesión. Desde aquí puede subir un número limitado de imágenes, cinco por cada subida. Además, tampoco se puede superar el número máximo de imágenes siendo su límite 50.

Por otra parte, al estar las imágenes subidas se puede comprobar a qué categoría pertenecen haciendo una petición al servidor pulsando el botón de comprobar.

Por ser parte a mayores de subir la imagen también se podrán eliminar de la interfaz y del propio servidor haciendo una petición.



Figura 8.10 Interfaz para la subida y comprobación de las imágenes.

### 8.1.8 Enrutamiento con VUE

Para la realización del enrutamiento de la aplicación se empleará la librería de vue-router.

Con esta librería se pueden controlar los distintos códigos de error de la aplicación como puede ser el **error 404**.

A la hora de tener la aplicación en fase de producción se encuentra que, al recargar una página, esta no se encuentra y nos saca el error 404. Con el fin de evitar este error, se tiene que añadir en el servidor apache donde se encuentre el HTML, un fichero .htaccess.

En este fichero .htaccess, se añade el código de la **Figura 8.11**, que redireccionará cualquier ruta a el fichero HTML.

```

<IfModule mod_rewrite.c>
  RewriteEngine On
  RewriteBase /
  RewriteRule ^index\.html$ - [L]
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteRule . /index.html [L]
</IfModule>

```

Figura 8.11 Fragmento del `.htaccess` para que sean visibles todas las rutas de la aplicación. (Fuente: <https://router.vuejs.org/guide/essentials/history-mode.html#example-server-configurations> )

Con este fragmento de código se soluciona el problema anterior y también, mejoramos la accesibilidad a las rutas de la web.

### 8.1.5 Menú de navegación

Para que la aplicación sea más cómoda de utilizar se empleará un menú de navegación con distintos estados. Un estado será para cuando el usuario este sin una sesión iniciada, y otro para cuando tenga la sesión abierta. Con esto se pretende que, si el usuario esta con su cuenta pueda modificar sus datos, o acceder a estas interfaces específicas.

En la **Figura 8.12** se puede ver el primer estado de esta barra de menú, que contiene tres opciones.

De estas tres opciones, la de inicio, contacto, y el logo son comunes a los dos estados. Dos de estas opciones como el inicio y el logo, enviarán al usuario a la página principal y la de contacto a la vista de contacto.



Figura 8.12 Menú sin iniciar sesión con la cuenta de usuario.

Por otro lado, la opción de Iniciar sesión se enviará a la ventana de login, donde si se inicia sesión con la cuenta de usuario aparecerá el siguiente menú, como el mostrado en la **Figura 8.13**.



Figura 8.13 Menú con la sesión iniciada.

En este menú se muestran dos nuevas opciones, una permite ver los datos de usuario, enviándola al perfil de usuario, y la otra, la de cerrar la sesión de la cuenta que volverá a aparecer el menú de la **Figura 8.12**.

Por otro lado, el menú estaría hecho de forma adaptable a cualquier dispositivo independientemente de su pantalla, como se ve en la **Figura 8.14**. Tratándose de un dispositivo móvil.



Figura 8.14 Menú visto en dispositivo móvil, diseño responsive.

### 8.1.6 Pie de página

El pie de página sirve solo para mostrar datos acerca del proyecto, logo, copyright, o redes sociales, a parte del mapa de la web.



Figura 8.15 Pie de página de la web.

### 8.1.7 Mis imágenes

La interfaz de “Mis imágenes”, es donde el usuario, una vez iniciada su sesión, podrá ver todas las imágenes que ha ido añadiendo a la web. Se le da la posibilidad del filtrar por los diferentes patrones y por fechas de más antigua a más reciente, o viceversa.

Aquí también cuenta con la opción de eliminar imágenes sueltas o todas las imágenes.

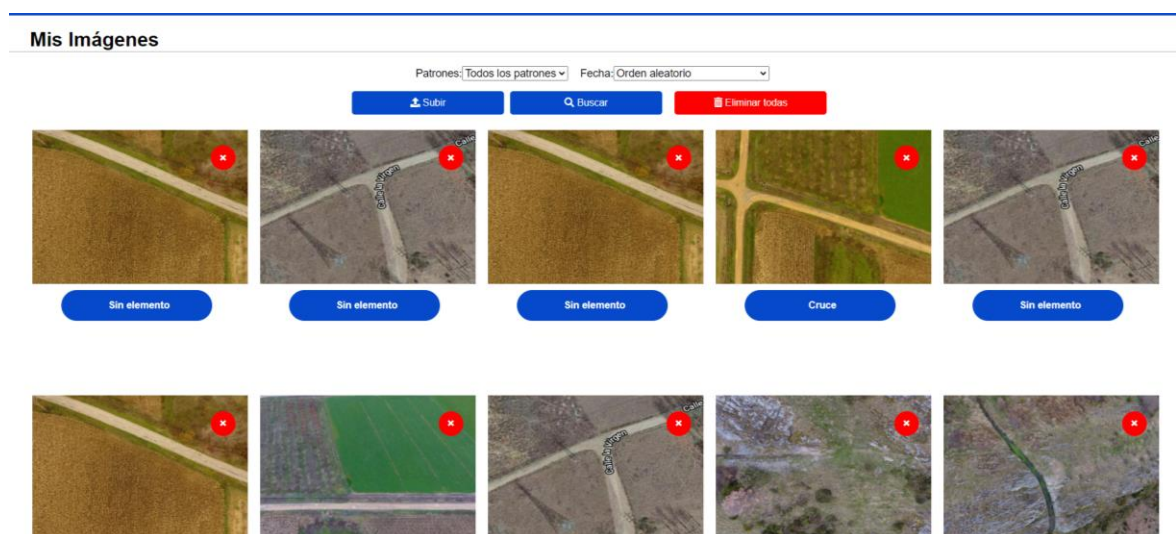


Figura 8.16 Interfaz de mis imágenes, propia de cada usuario.



## 8.2 Parte del servidor

### 8.2.1 Conexión a base de datos y modelos tablas.

La conexión a la base de datos, se realizará por medio del ORM de Sequelize, dentro del servidor de Node.

Para que todo esto funcione se tiene que crear la conexión, indicando el puerto y la IP donde se encuentra el servidor de base de datos, y el tipo de base de datos con el que se trabajará, en este caso será MySQL.

Tras esta conexión se crearán todos los modelos de las distintas tablas de base de datos, indicando los campos de cada tabla, a qué tipo de datos pertenece, y si están relacionadas. Lo bueno de trabajar con este Sequelize es que al hacer las peticiones devuelve los datos en un formato cómodo para la web como es el JSON.

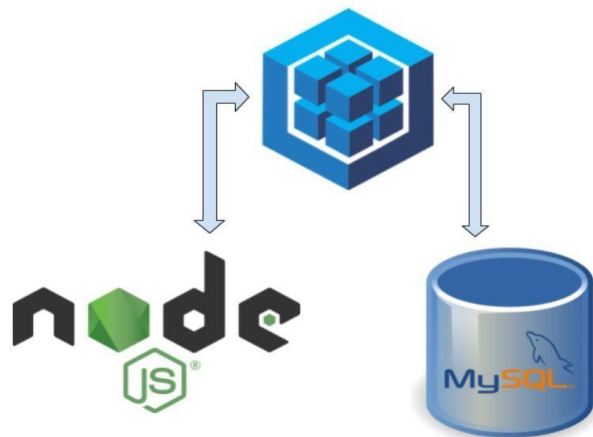


Figura 8.17 ORM de Sequelize con conexión a base de datos MYSQL mediante Node  
(Fuente: <https://medium.com/@ahmedcharef/sequelize-orm-with-node-js-koa-mysql-foreign-keys-a3097ab91a8> )

### 8.2.2 Subida de imágenes

Al subir imágenes desde la interfaz de subida de imágenes, estas se suben al servidor general para ser almacenadas y así después poder clasificar la imagen según su patrón.

Sobre esta petición a parte de subir las imágenes, se cambia el nombre para que no haya dos imágenes iguales y existan problemas de sobrescritura. También se añaden a bases de datos relacionando la imagen con el usuario, añadiendo una fecha de subida, y

devolviendo a la interfaz un conjunto completo de los datos, que se han añadido a la base de datos. Con esto se obtiene una mejor referencia a la hora de operar con ellas o eliminarlas.

En la eliminación de las imágenes, estas no se borran del servidor. De esta manera, quedan útiles para futuros entrenamientos de la red neuronal, pero sí que se borran de la base de datos, quitando toda relación con el usuario que la subió.

### 8.2.3 Envío de correos electrónicos

Será una de las funcionalidades con las que contará la aplicación web, y que se realizará desde la parte servidora. Para ello se utilizará la librería de javascript de **nodemailer**, en la cual se tendrá que poner un correo fijo que será el [dferna10@estudiantes.unileon.es](mailto:dferna10@estudiantes.unileon.es) y un correo dinámico, que se introducirá por la interfaz de usuario del frontend. A parte del asunto y el cuerpo del mensaje



*Figura 8.18 Icono de la librería utilizada para envío de correos. (Fuente: <https://nodemailer.com/about/>)*

Para poder enviar correos con esta librería se tiene que activar la opción de nuestro correo de Google, de permitir aplicaciones poco seguras, como podemos ver en la **Figura 8.19**.

## ← Acceso de aplicaciones poco seguras

Algunos dispositivos y aplicaciones utilizan una tecnología de inicio de sesión poco segura, lo que aumenta la vulnerabilidad de tu cuenta. Te recomendamos que desactives el acceso de estas aplicaciones, aunque también puedes activarlo si quieres usarlas a pesar de los riesgos que conllevan. Desactivaremos este ajuste de forma automática si no lo utilizas. [Más información](#)

Permitir el acceso de aplicaciones poco seguras: Sí



*Figura 8.19 Aplicaciones poco seguras para permitir el envío de correos*

Tras la activación de aplicaciones poco seguras del correo electrónico, ya se pueden enviar emails con esta librería.

### 8.2.4 Comprobación de patrones

Al hacer la petición desde la interfaz de subida y comprobación de las imágenes, se envía una matriz de los objetos JSON de las imágenes, que posteriormente se enviarán a nuestra red neuronal.

En este punto ejecutamos el script de Python de predicción, enviándole un mínimo de tres parámetros:

- **Archivo de predicción:** archivo que es el que queremos ejecutar.
- **Número de imágenes:** número total de imágenes que queremos procesar.
- **Imágenes:** Imágenes enviadas con el nombre y la extensión.

Tras la ejecución de este script, se devuelve un JSON, con la imagen y su etiqueta predicha por el sistema, a parte se envía a la interfaz. También, se actualiza este valor de los patrones en la base de datos.

## 9 Optimización de motores de búsqueda (SEO)



Figura 9.1 Infografía sobre el contenido del SEO (Fuente: <https://marketing4food.com/como-encontrar-palabras-clave-para-seo/>)

### 9.1 ¿Qué es SEO?

Search Engine Optimization (SEO) es la optimización de una web para aparecer en los primeros resultados de búsqueda orgánica en los diferentes buscadores. Esto sirve para que los motores de búsqueda indexen la información de las páginas web, en función de la importancia para los usuarios.

A parte depende de muchos factores de la web, como pueden ser los títulos, y las meta etiquetas de cada página, si es un diseño responsive, si los valores de carga son óptimos, etc.

### 9.2 SEO en la web

En esta aplicación web como es una SPA (Single Page Application), para que funcione el SEO, deberemos instalar una librería de JavaScript, que nos permite cambiar los títulos y las meta etiquetas de forma dinámica dependiendo de la interfaz en la que nos encontremos.

Los títulos tienen que ser únicos, y bien referenciados a la funcionalidad que realice cada página. Sin embargo, si a mayores de esto se le introduce palabras clave mejorará bastante este apartado.

Las meta-descripciones, son descriptivas y diferentes para cada una de las páginas, intentando ser atractivas para el usuario que las vea. Ejemplo de ellos son los títulos que deben contener algunas palabras clave para así tener mejores resultados.

En relación a las imágenes, es obligatorio que tengan una breve descripción de que es la imagen en su etiqueta alt. Además, en cada imagen se debe colocar la etiqueta **loading="lazy"**, que evitará cargar la imagen de forma innecesaria influyendo que tarde más tiempo en la carga inicial de la web. También, tienen que tener un tamaño adecuado para no bloquear mucho la carga inicial, por lo que lo mejor es reducir su tamaño desde un principio al similar al que se quiere dar en la web.

Si realizan estas acciones, se tendrán unos resultados óptimos de carga de la aplicación web, por lo que no se encontrará penalizada por los motores de búsqueda y mejor indexación se obtendrá.

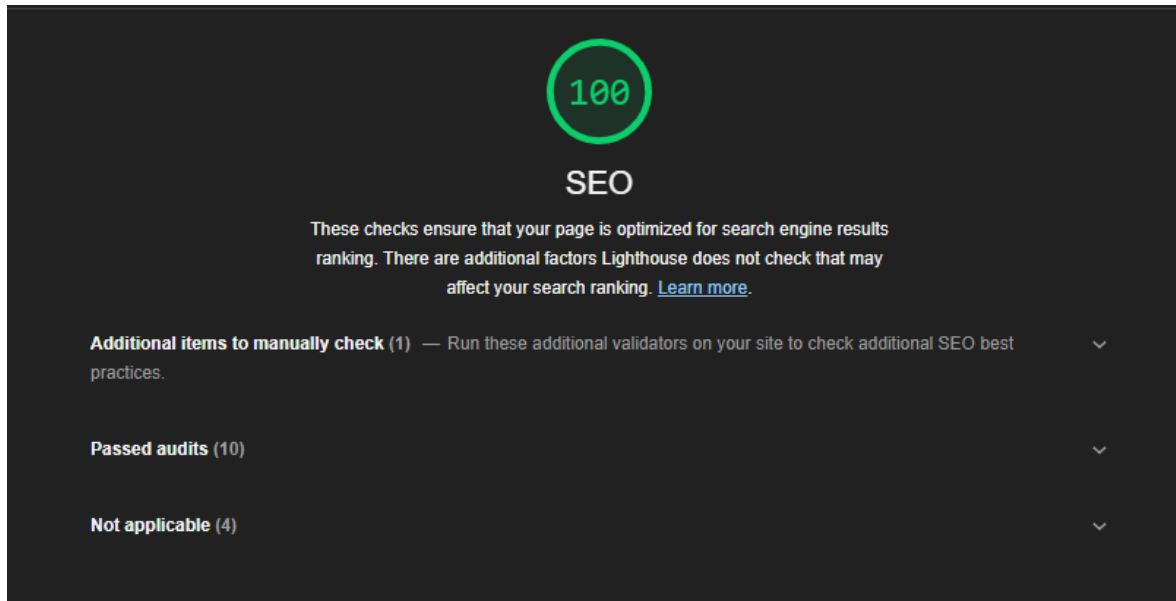
Una forma de tratar los errores 404, es la creación de páginas completamente personalizadas que redirijan al usuario a páginas completamente funcionales. Con esto los motores de búsqueda beneficiarán esta aplicación.

Otra de las opciones realizadas es la de crear rutas simples, que puedan llegar a transmitir información sobre la funcionalidad de esa página.

### 9.3 Informe SEO

Se puede analizar el SEO de la web en un entorno local con la herramienta de Lighthouse. Esta aplicación genera un informe sobre aspectos de la web, en el cual aparecen el SEO, accesibilidad, y el uso de las buenas prácticas empleadas. Dentro de estos informes además se pueden ver los elementos problemáticos de la web para tratar de corregirlos.

Como se puede ver en la **Figura 9.2** nos encontramos con un SEO al 100%. Por lo que la página web estaría bastante bien posicionados con esta medida.



*Figura 9.2 Reporte de SEO de Lighthouse.*

También se encuentra la medida de Accesibilidad, en la cual se encuentra en 95 de los 100 puntos. Esta medida indica pequeños fallos, los cuales pueden ser que los colores no se ven bien o no hay contrastes con el fondo. Por otro lado, otra de las penalizaciones es que no se sigue el orden de los elementos. Esta medida se puede observar en la **Figura 9.3**.

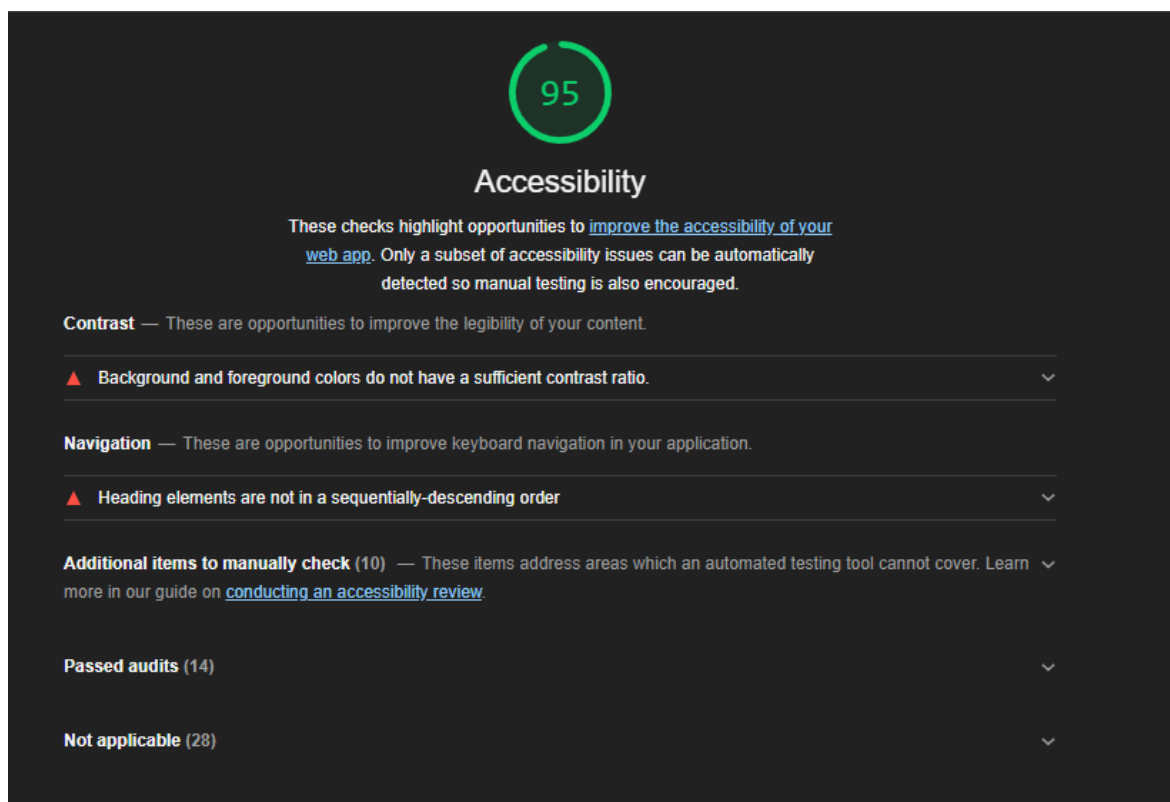


Figura 9.3 Reporte de la accesibilidad de la aplicación.

Por último, la medida de buenas prácticas, en la cual se puede ver que se tienen unos resultados en un punto el cual no es del todo malo, pero se podría mejorar.

Uno de los problemas con los que se nos encontramos es que no se usa https, al tratarse de un entorno local. Otra de las penalizaciones, se trata de un tamaño incorrecto de imagen.

Esta última medida se puede observar en la **Figura 9.4**.

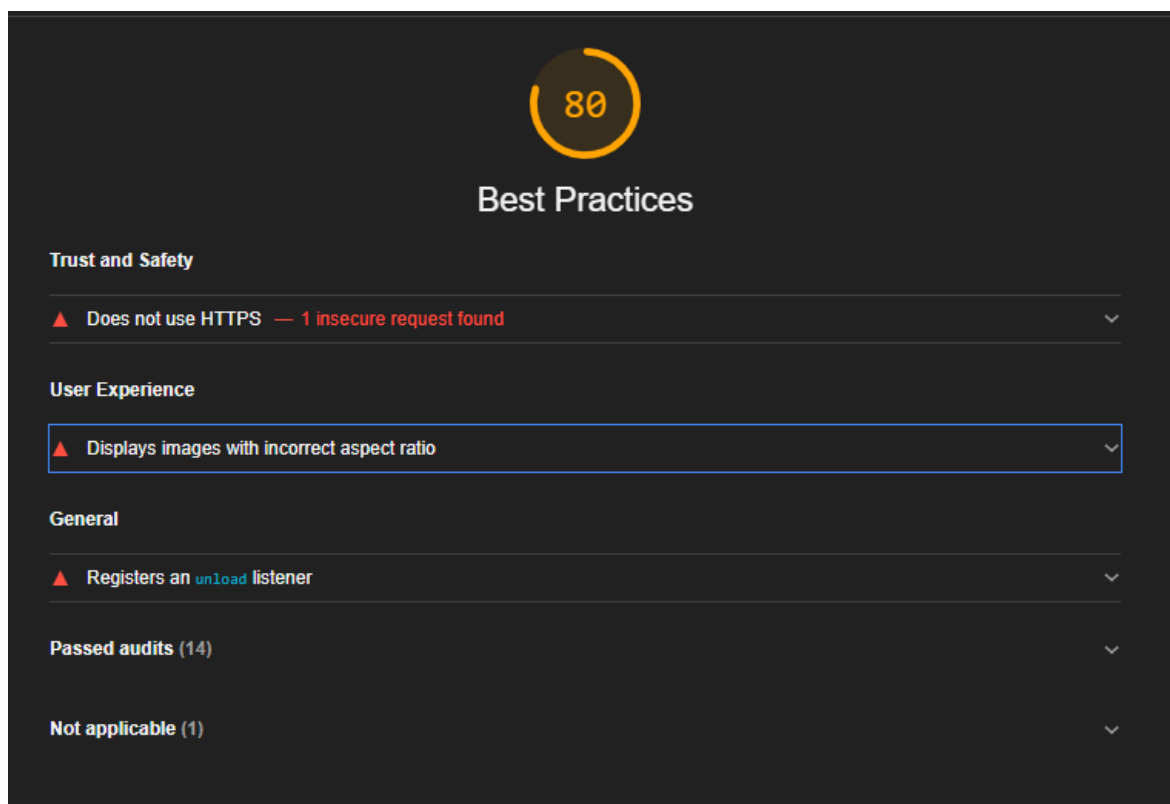


Figura 9.4 Reporte de buenas prácticas.



## 10 Control de versiones

Para evitar los errores o pérdida de versiones estables de todo el sistema, se han utilizado un controlador de versiones dentro del proyecto, con dos repositorios diferentes para cada parte del proyecto. Uno para la red neuronal y otro para la aplicación web.

El controlador de versiones que se ha empleado es GitHub, en el que se tendrán mínimo dos ramas, una de producción y otra de desarrollo. En la de producción se obtendrá la última versión estable de cualquiera de los 2 software para así no tener problemas de modificaciones y errores en otra zona del código. Mientras que en la de desarrollo, solo se tendrá la fase de pruebas, implementación, mejoras y corrección de errores.



*Figura 10.1 Control de versiones GitHub usado en el proyecto (Fuente: <https://medium.com/swlh/how-to-auto-sync-update-from-one-github-repository-to-other-repository-using-github-workflow-6492f4830d45> )*

### 10.1 Seguimiento de la red neuronal

Se puede ver el seguimiento del repositorio de la red neuronal convolucional dentro de GitHub, en la **Figura 10.2**.

Nov 15, 2020 – Sep 6, 2021

Contributions: Commits ▾

Contributions to main, excluding merge commits and bot accounts

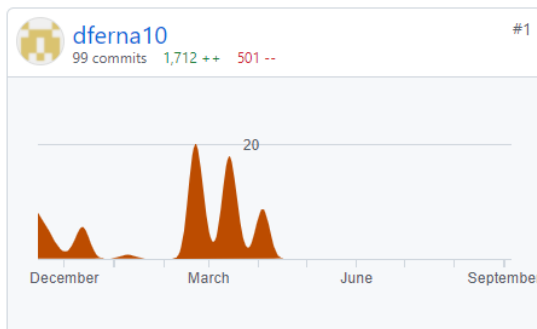
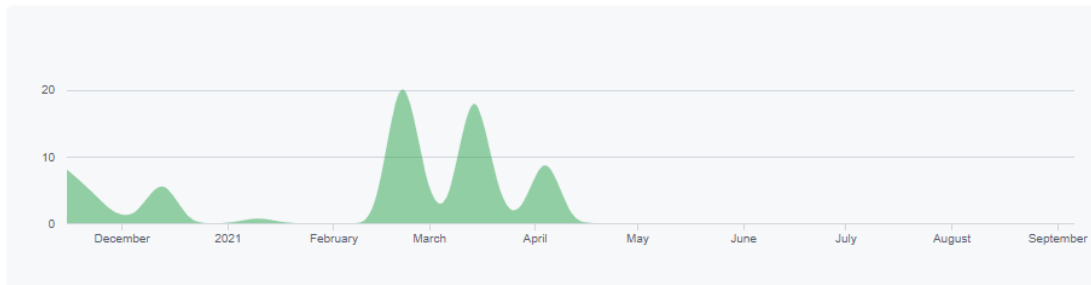


Figura 10.2 Seguimiento de commits de la red neuronal.

También se muestra el seguimiento de las diferentes ramas de este repositorio en la **Figura 10.3**. En esta imagen se encuentran las uniones y cambios entre ambas ramas comentadas anteriormente.

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.

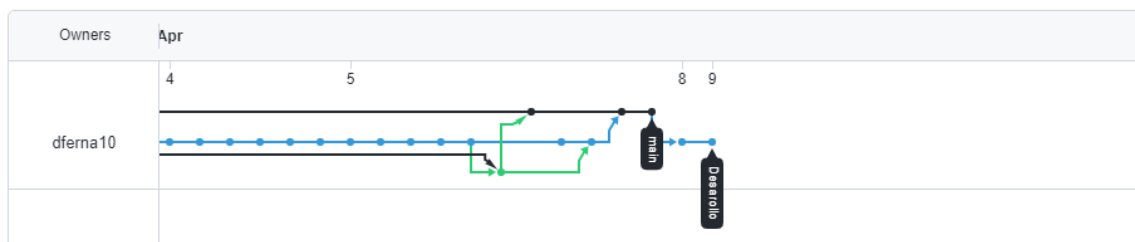


Figura 10.3 Seguimiento de las ramas de la red neuronal.

## 10.2 Seguimiento de la aplicación web

De igual forma, se muestra en la **Figura 10.4** el seguimiento del repositorio en el que se encuentra la aplicación web, mostrando el número de entradas por meses, líneas eliminadas y añadidas.

Oct 11, 2020 – Sep 6, 2021

Contributions: Commits ▾

Contributions to main, excluding merge commits and bot accounts

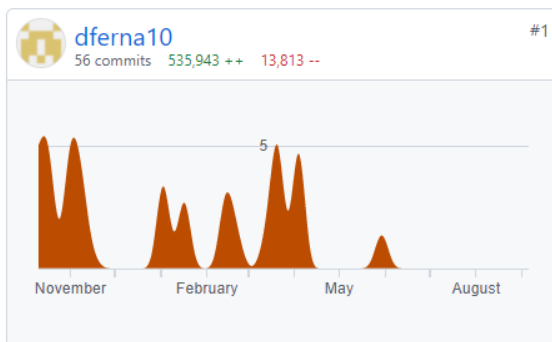
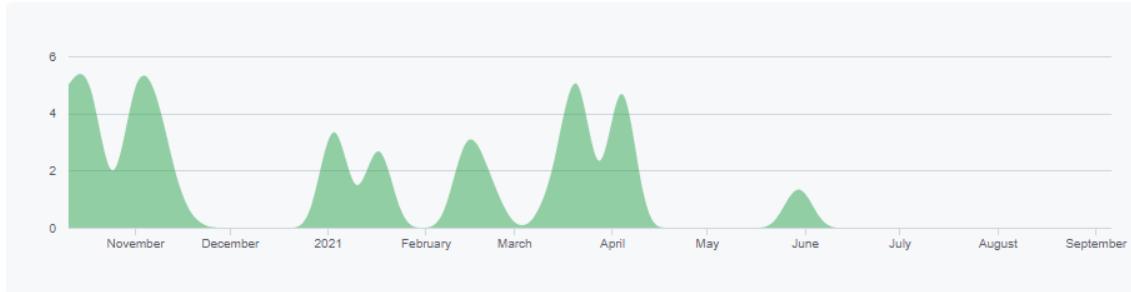


Figura 10.4 Seguimiento de commits de la aplicación web.

Ahora se puede observar el seguimiento de las ramas a lo largo del tiempo, con todas sus uniones y versiones estables. Esto lo podemos observar en la **Figura 10.5**.

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.

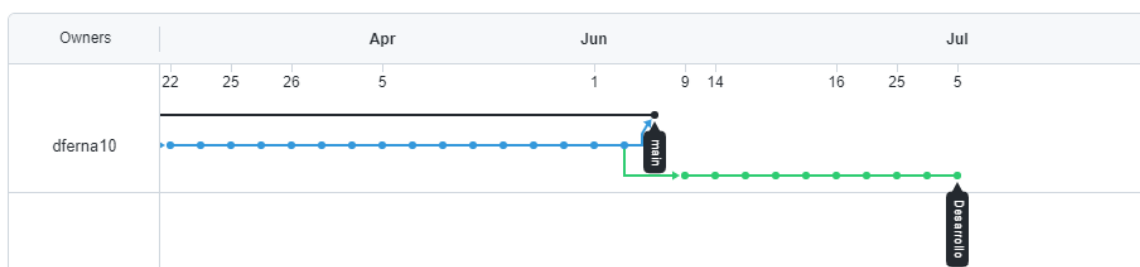


Figura 10.5 Seguimiento de las ramas de la aplicación web.

## 11 Resultados

Para la realización del proyecto, se ha propuesto la creación de un nuevo modelo de red neuronal convolucional, como el de la **Figura 11.1**.

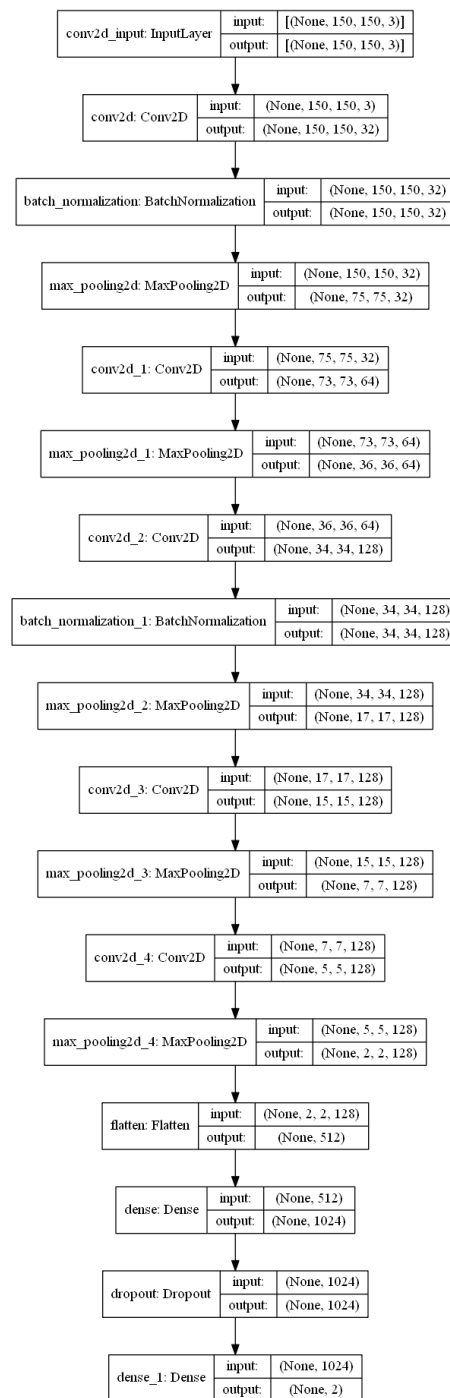


Figura 11.1 Red convolucional planteada al inicio del proyecto

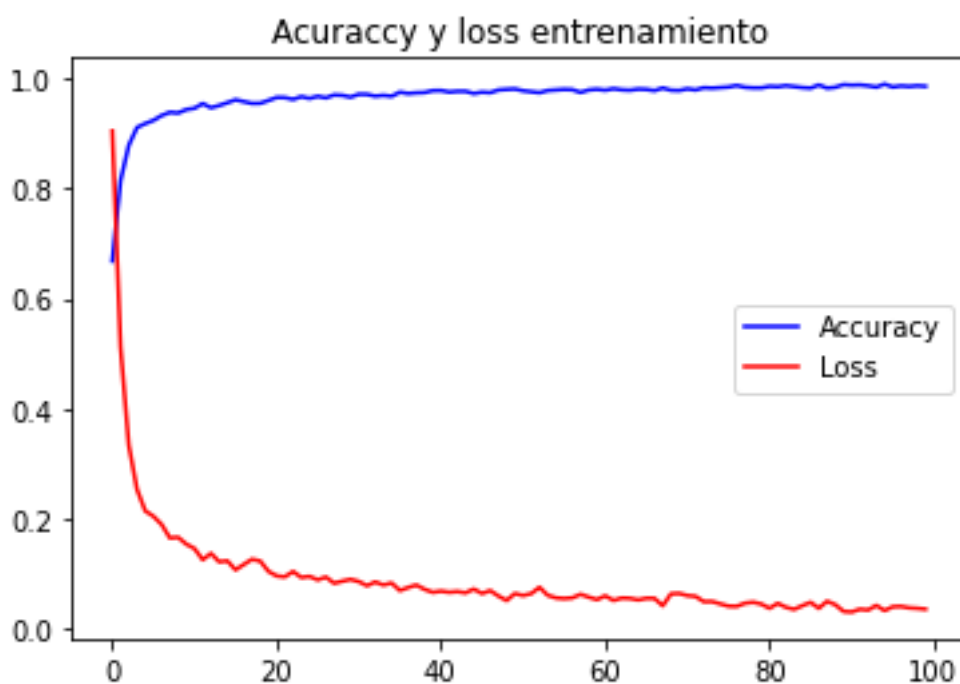
### 11.1 Entrenamiento inicial

Al durar tanto tiempo el entrenamiento de la red neuronal convolucional propuesta, se ha añadido un sistema de puntos de referencias tras cada iteración que finaliza, cogiendo el mayor valor de “accuracy”. Con esto, aunque no termine de entrenar la red, se pueden tener un punto de copia que sirve de referencia para después probar a predecir o volver a entrenar nuestra red desde ese punto.

Tras el primer entrenamiento con todas las imágenes, se han obtenido unos resultados que no son del todo malos, pero son bastante mejorables.

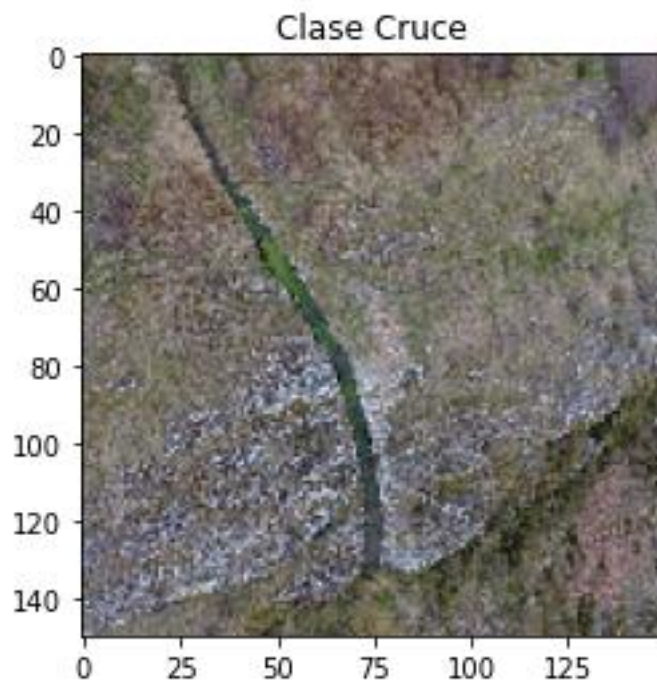
Se ha conseguido que el modelo alcance un 99% de accuracy en fase de entrenamiento, con 100 iteraciones. Posteriormente, este resultado se contrastará con la fase de test.

En la **Figura 11.2** se observa que se encuentra un modelo bastante constante, y que no cae en el sobreajuste de la red.



*Figura 11.2 Gráfica de entrenamiento de la red donde podemos ver la precisión y la pérdida de nuestro modelo*

En la fase de test se ha llegado a obtener un 88% de acierto ante el conjunto de imágenes utilizadas en este momento. Ver la **Figura 11.3**



*Figura 11.3 Ejemplo de error a la hora de predecir un nuevo elemento en una imagen.*

En la **Figura 11.3**, se observa el error que se comentaba anteriormente a la hora de predecir un elemento sobre una imagen nueva. En esta figura se puede ver que se trata de un canal de transporte de agua, y la red lo clasifica como un cruce de caminos. Esto mismo está pasando con otras de las imágenes de test para esta fase de pruebas de predicción.

## 11.2 Mejoras y correcciones del entrenamiento inicial.

Tras realizar este entrenamiento, que duró aproximadamente veinticinco horas, en el cual se cargaban las imágenes por lotes redimensionándolas en memoria pasando de 3000x4000 píxeles a 150x150 píxeles, nos dimos cuenta de que tardaba mucho. Por este motivo, se tomó la decisión de redimensionar las imágenes en otro directorio diferente en sus correspondientes órdenes para así posteriormente entrenar la red completa con estas imágenes con un tamaño de 150x150.

También se ha reducido la red neuronal para el procesamiento de las imágenes, que se puede ver en la **Figura 11.4**. Con esto hemos cambiado para que el paso de cada “kernel” convolucional y de “pooling” en vez de ir pasando por cada pixel, fuese cambiando de 2 en 2 estos filtros. Por lo tanto, se ha reducido la red.

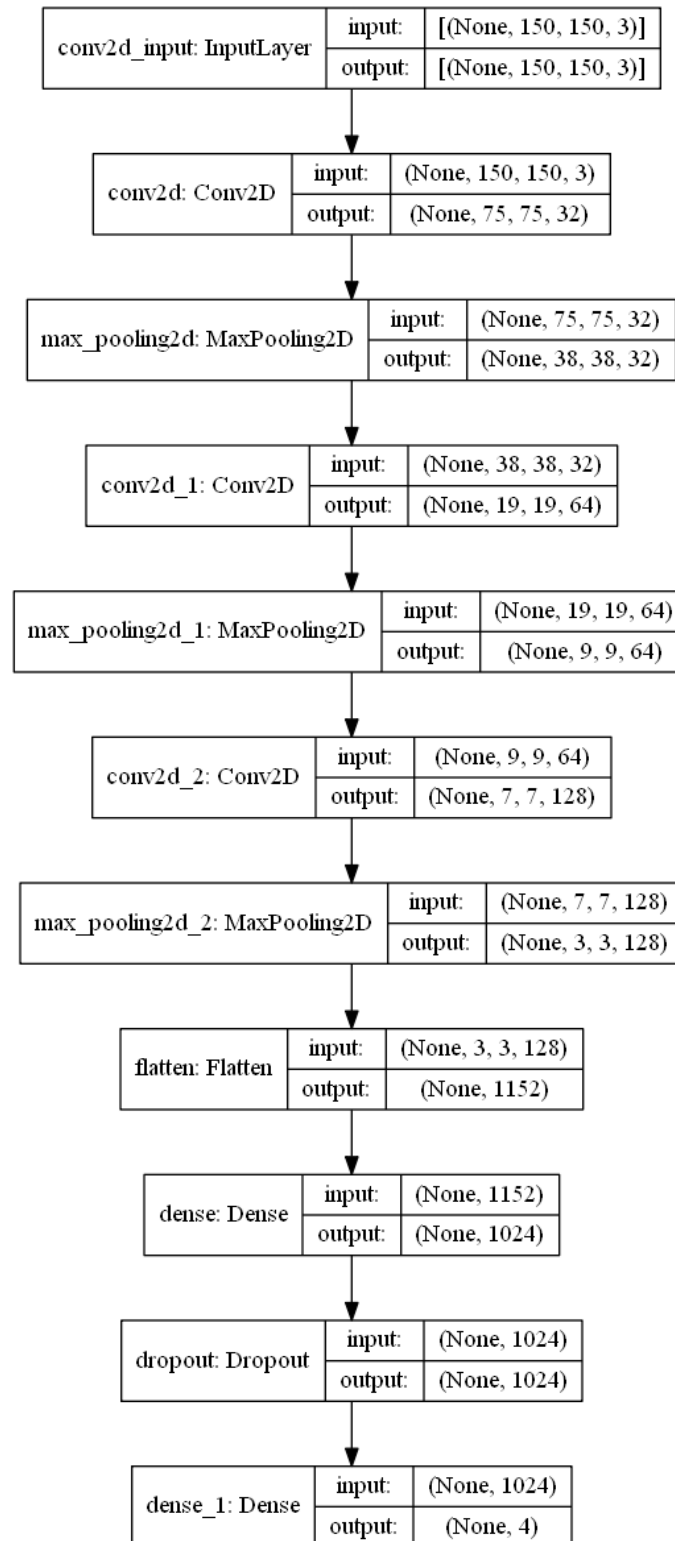
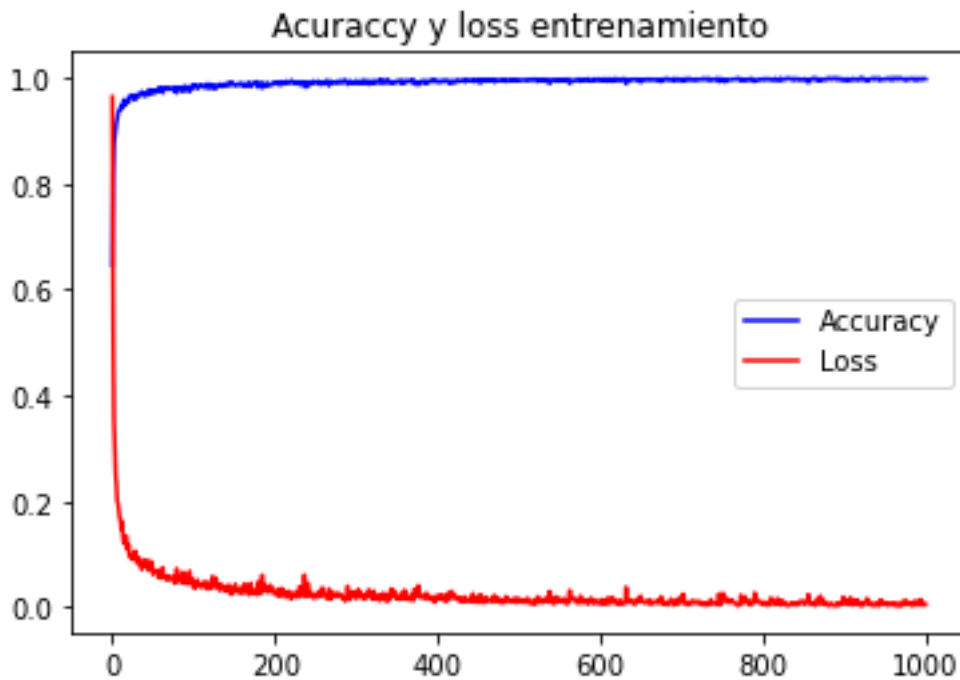


Figura 11.4 Nueva red neuronal convolucional, mejorada para la solución del problema

Con este cambio en el tamaño de las imágenes se consigue que el entrenamiento sea más eficiente y rápido con lo cual, en menos tiempo, se pueden llevar a cabo muchas más iteraciones, mejorando así el modelo.

A través de los cambios realizados, se entrenó en el mismo tiempo, que el modelo anterior la misma red neuronal, pero con 1000 períodos, obteniendo los resultados de la **Figura 11.5**, con un valor muy cercano al 1 de “accuracy”.



*Figura 11.5 Gráfica del entrenamiento de 1000 períodos con loss y accuracy*

Tras este entrenamiento, se comprobó las imágenes reservadas de test y llegó a sacar un 95% de acierto. Sin embargo, se vio que fallaba mucho al predecir las que no tienen cruce. Esta categoría es la más desbalanceada a la hora del entrenamiento la red, dado que solo cuenta con 500 imágenes y las otras tienen 1000 aproximadamente.

Se creó un nuevo script de Python con la carga por lotes en la que se han ido aumentando las imágenes de no cruce, con el data augmentation de Keras. Estas imágenes se han ido guardando para así tener las 1000 aproximadamente. Con esto ya se tienen igualadas las cuatro clases requeridas. Estos cambios aplicados, se pueden ver en la **Tabla 11.1**



Categoría	Entrenamiento	Test	Total
Canales	1035	611	1646
Cruces	899	518	1417
Minas	998	1084	2082
No cruces	1023	220	1243
	Total		6388

Tabla 11.1 Aumento y reestructuración del conjunto de datos.

### 11.3 Solución de errores de predicción

Uno de los problemas que se han obtenido ha sido que de un alto porcentaje de acierto en el test, llegando a un 95%, las predicciones no eran del todo válidas, dándose un alto porcentaje de error.

Todo esto se ha solucionado normalizando los píxeles de las imágenes que entran a la red, en la fase de predicción. Esta normalización consiste en cambiar los valores de píxeles de una imagen de 1 a 255, a quedar entre los valores de 0 y 1.

Al realizar la corrección de la normalización, antes de cargar en memoria el mapa de características de las imágenes a la hora de predecir los patrones. Con esto se ha mejorado el acierto que salía de esta parte. Como se puede observar en la **Figura 11.6**

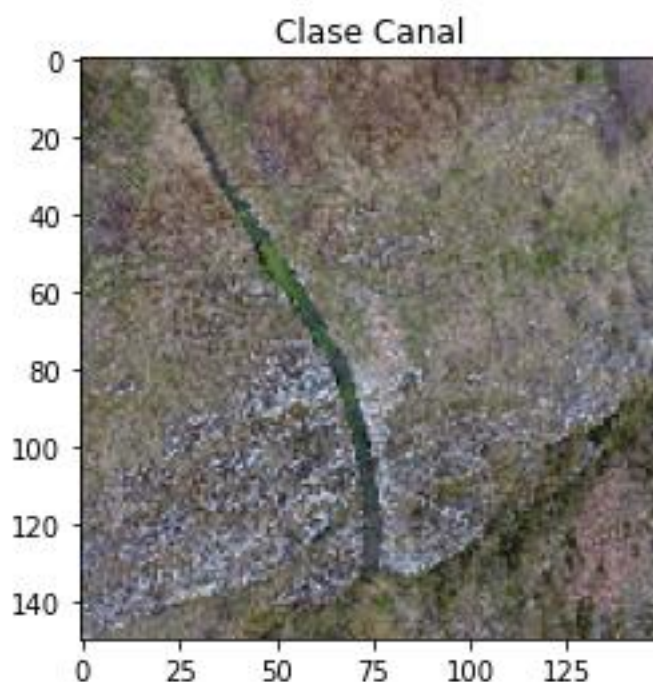


Figura 11.6 Imagen de demostración de corrección de la clase a una correcta

Si comparamos las **Figuras 11.3** y **Figuras 11.6**, podemos darnos cuenta de que se trata de la misma imagen solo que la primera la clasificaba como cruce, cuando en realidad es un canal de transporte de agua. Por ese motivo, la normalización tiene que existir para evitar que la red neuronal convolucional genere errores en la clasificación de las nuevas imágenes.

## 12 Conclusiones y recomendaciones

### 12.1 Trabajo realizado

En este Trabajo de Fin de Grado se han tratado los siguientes temas:

- Captura de imágenes aéreas con dron de los diferentes patrones que se desearon convenientes para este TFG.
- Construcción del dataset con las imágenes capturadas por el dron, y separación en los diferentes patrones, que son sus etiquetas.
- La creación de un modelo de red neuronal para el reconocimiento de los distintos patrones seleccionados.
- Implementación de una aplicación web para que sea más fácil y accesible para cualquier usuario utilizar este servicio.

### 12.2 Trabajos futuros

Como acciones futuras de este proyecto, se proponen la mejora y corrección de errores en torno a dos aspectos principales:

Por un lado, tenemos la aplicación web, para la cual se proponen varias mejoras visuales para así evitar ambigüedades, como que el usuario no llegue a comprender la interfaz para un realizar el correcto funcionamiento del sistema. Dentro de este mismo punto también se tratará la mejora de la subida de imágenes, para que pueda aumentar el número de imágenes subidas y sea más eficiente.

También se deberá mejorar la conexión del servidor web con el modelo de red neuronal para llegar a ser más eficiente y rápido.

Por otro lado, se propone la mejora del modelo de red neuronal para que aparte de clasificar las imágenes aéreas dentro de un patrón, también las pueda ubicar dentro de un mapa. Siendo mucho más fácil para el usuario que utilice el sistema ver donde se encuentran los elementos.

### 12.3 Problemas encontrados

A lo largo del tiempo de desarrollo han ido surgiendo distintos problemas con las diferentes partes del proyecto, que se van a ir comentando.

Dentro del modelo de sistema inteligente, resultó bastante complejo que reconociese bien todos estos patrones indicados. Para ello, es necesario llegar a tener un gran porcentaje de

acierto o fiabilidad. Con esto se tuvieron que ajustar un gran número de parámetros hasta alcanzar el resultado deseado.

Otro de los puntos que fueron un poco más complejos fue la creación del dataset, ya que se tenían algunas imágenes mezcladas. Para ello, en el conjunto inicial, aunque estaban ya separadas en tres vuelos completamente diferentes, se tuvo que eliminar aquellas que no contenían el patrón, y separar las que podían estar en otra clase.

Un claro ejemplo de este caso fue la separación de las imágenes de caminos que obligó a recorrer todas para separar las que contuviesen cruce de las que no.

La aplicación web resultó ser una herramienta versátil a pesar de la dificultad para establecer la conexión del servidor con la red neuronal. La creación del correo electrónico para enviar y recibir los datos de los usuarios también contó con un cierto grado de dificultad hasta enlazar con los cambios de la interfaz necesaria.

#### 12.4 Opinión personal

La implementación de este proyecto, me ha servido para aprender un poco más acerca de las redes neuronales. He tenido que investigar sobre las mismas y concretamente las convolucionales, además de crear este modelo completamente desde cero ajustando todos los parámetros hasta llegar al resultado obtenido.

En el ámbito de la aplicación web, lo que me resultó más complicado fue establecer las conexiones con la red neuronal y, el envío de correos electrónicos.

Desde mi punto de vista ha sido un proyecto muy ilusionante tanto desde la perspectiva profesional como personal. He aprendido mucho con este trabajo y me he dado cuenta de lo complicado que puede resultar entrenar un sistema inteligente para obtener unos resultados fiables.

## 13 Análisis de prototipo.

### 13.1 Presupuesto

En este apartado se expondrán los costes totales generados en la realización de este proyecto.

Se empezará por los costes de software y hardware. Todo el software empleado ha sido software libre, por lo cual no se han generado gastos extra.

Los costes de hardware los encontramos en la **Tabla 13.1**.

Concepto	Unidades	Tiempo de uso en semanas	Coste Total (€)
Ordenador	1	40	900
Gráfica NVidia GT 1650	1	35	150
Dron DJI Phantom 4	1	1	1.999

*Tabla 13.1 Costes hardware del proyecto.*

Por otro lado, tenemos los gastos de personal involucrado en el proyecto, se encuentran expuestos en la **Tabla 13.2**.

Concepto	Unidades	Tiempo de trabajo (Horas)	Coste por hora (€/h)	Coste total (€)
Ingeniero Informático	1	800	10	8.000
Piloto de dron	1	20	15	300

*Tabla 13.2 Costes de personal del proyecto.*

Por último, otros costes derivados de la implementación del proyecto son los de la **Tabla 13.3**.

Concepto	Tiempo de uso en semanas	Coste por semana (€)	Coste Total (€)
Conexión a internet	40	10	400
Luz	35	15	525
Lugar de trabajo	35	40	1400

*Tabla 13.3 Otros costes relacionados con el proyecto*

A continuación, se exponen los gastos que ha generado la implementación de este proyecto. Esto se podrá visualizar en la **Tabla 13.4**.

Concepto	Total
Hardware y software	3049
Personal	8300
Otros Gastos	2325
	13.674

*Tabla 13.4 Coste global del proyecto*

Como se puede ver el final del proyecto sería con un gasto total de 13.674€.

## 13.2 Análisis de mercado

Dentro de este apartado se estudiará las diferentes empresas existentes y compararlas con nuestro sistema creado.

### 13.2.1 Análisis de empresas

Para realizar este análisis se han encontrado diversas empresas por internet que emplean drones y técnicas de Deep Learning.

Una de estas empresas es el propio fabricante del dron DJI. Esta empresa utiliza estos sistemas para:

- **Construcción:** medir datos y analizarlos.
- **Infraestructuras:** inspecciones detalladas.
- **Energía:** Inspecciones de estas infraestructuras de forma segura.
- **Seguridad Pública:** Obtener la información en tiempos cortos.
- **Agricultura:** para conocer mejor los cultivos para mejorar cosechas.
- **Rodaje:** mejorar la planificación de todos los equipos.

También se han encontrado con Pix4D, que tienen algunas de las funcionalidades que se han comentado en la empresa DJI.

### 13.2.2 Comparativa con el prototipo

Las empresas analizadas no tratan elementos de patrimonio, sino que trabajan sobre elementos de la actualidad, como pueden ser agricultura eficiente, inspección de infraestructuras, etc. Por lo tanto, este prototipo podría encontrar un nicho de mercado como sería la investigación de patrimonio, arqueología, etc.

El empleo de este prototipo se podrían hallar nuevos elementos que en la actualidad no se hayan descubierto. Además, con esta aplicación se podrían encontrar elementos de patrimonio y que no fuesen de patrimonio. Sin embargo, su principal uso sería relacionado con el patrimonio histórico.

### 13.2.3 Análisis de costes

Como se puede ver en el apartado de presupuesto, se encuentra el coste total de la construcción del prototipo, pero faltaría añadirle ciertos gastos extra para que fuese completamente funcional.

Estos gastos van a ser:

- **Servidor:** este servidor será donde alojemos todo nuestro sistema (aplicación web, servidor de base de datos, red neuronal, y todas las imágenes subidas por el usuario). Tiene que ser un servidor fuerte para aguantar la subida de grandes cantidades de imágenes, y poder clasificarlas según su categoría.
- **Certificado SSL:** certificado de seguridad necesario para que la aplicación sea segura de cara al usuario y los motores de búsquedas.
- **Dominio:** url por la que el usuario va a acceder a nuestro servicio web. Por ejemplo: *dronrecognition.com*

Todos los gastos comentados se pueden ver en la **Tabla 13.5**.

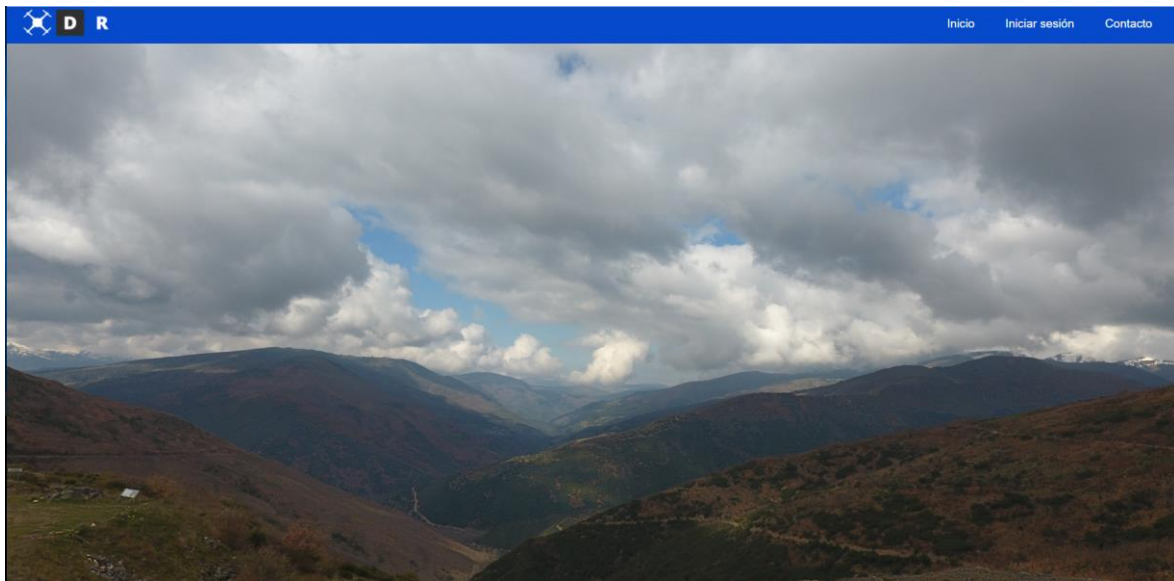
Concepto	Unidades	Tipo de Pago	Coste (€)
Servidor	1	Mensual	8
Certificado SSL	1	Anual	55.99
Dominio	1	Anual	20

*Tabla 13.5 Gastos de producción del prototipo.*

## 14 Manual del usuario

Para comenzar a utilizar esta aplicación web, se va a mostrar la forma de usar todas las opciones y menús.

Al inicio se encontrará con la página principal, que puede ver cualquier usuario que acceda a este servicio. En ella, como en todas las demás vistas, se puede ver en la parte superior una barra de menú que vamos a explicar.



*Figura 14.1 Página principal de la aplicación web.*

Ahora mismo como no hay una sesión iniciada, solo tiene las acciones de iniciar tu cuenta de usuario en el menú de *"Iniciar sesión"* o ponerse en contacto con los creadores de este servicio en el botón de *"Contacto"*.

Si continuamos bajando en esta página nos encontramos con una vista que nos mostrará los elementos que se reconocen, y un botón para enviarnos a la subida de imágenes para probar la aplicación. Este último elemento, si no se encuentra la sesión iniciada, nos enviará al inicio de sesión o registro en el caso de que no se tenga una cuenta.





Figura 14.2 Segunda parte de la página principal.

Por último, en esta página, podemos ver otra forma de ponerse en contacto con la empresa, y un breve pie de página donde podemos ver tres campos, el primero es un mapa del sitio web, el segundo un correo de contacto, y el último son las redes de la empresa.

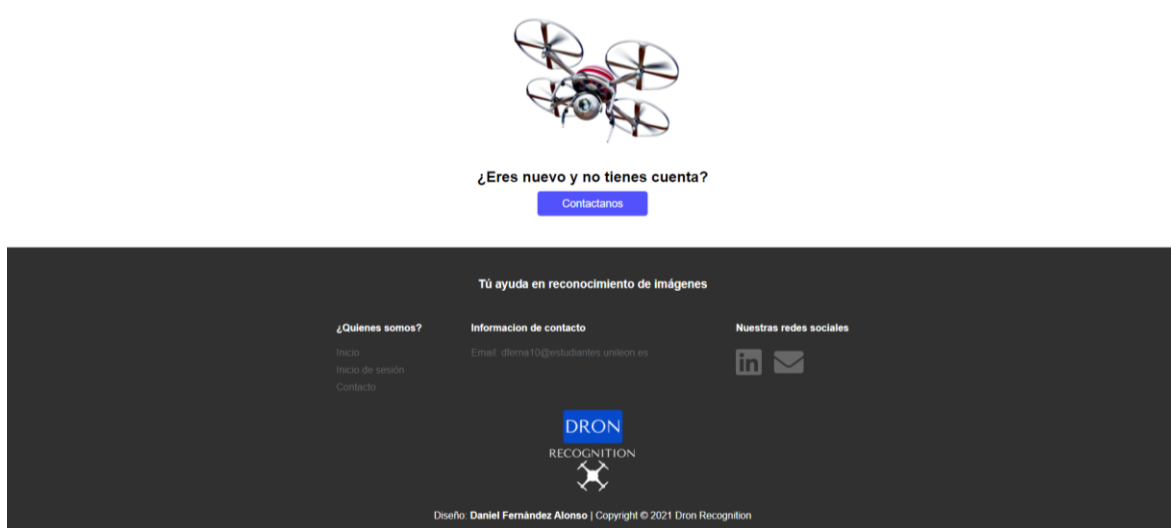


Figura 14.3 Fin de la página principal.

## 14.1 Página de login

En esta vista, el usuario puede registrarse si es nuevo, o iniciar la sesión si ya tiene una cuenta dentro de esta aplicación.

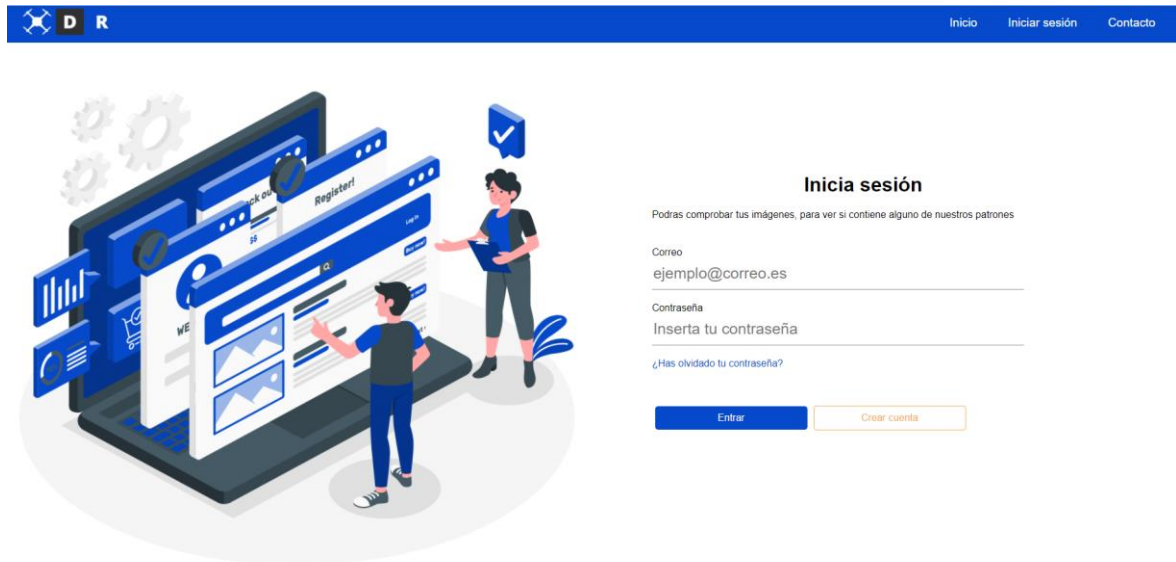


Figura 14.4 Interfaz de inicio de sesión

Sobre esta vista tenemos el menú ya comentado, que aparecerá en todas las vistas de la aplicación web.

El login cuenta con tres elementos de acción:

- 1) **Botón de “Entrar”**: este botón sirve para comprobar que el usuario se encuentra registrado en base de datos y que, además, haya introducido bien todos los datos para su inicio de sesión en el que se requiere como vemos correo electrónico y contraseña.
- 2) **Botón de “Crear cuenta”**: esta nos dirigirá al registro del usuario.
- 3) **¿Has olvidado tu contraseña?**: este botón hará saltar un cuadro de dialogo donde tendrá que introducir su correo electrónico con el que se encuentra registrado. Si no se encuentra registrado. A través de esta opción posteriormente se enviará un correo para realizar un cambio de contraseña. Podemos verlo en la **Figura 14.5**.

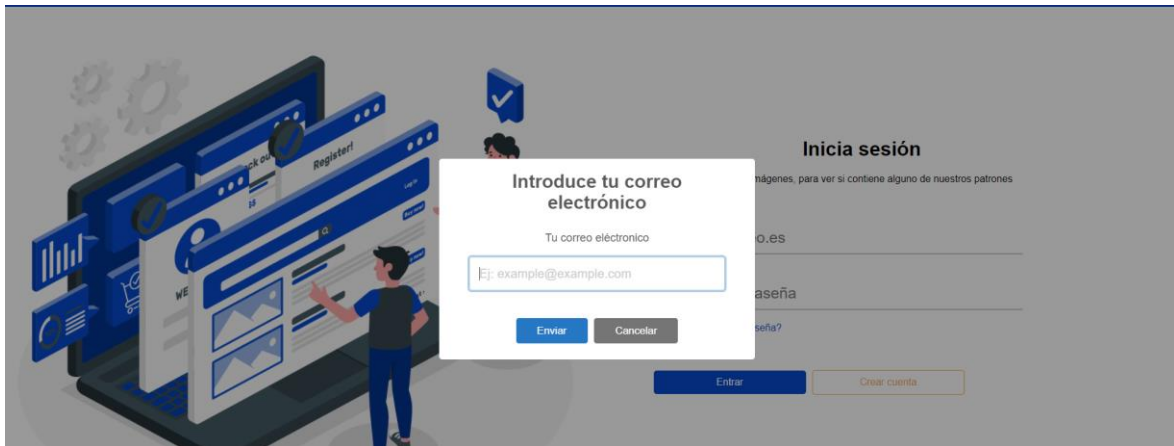


Figura 14.5 Alerta de ¿Has olvidado tu contraseña?

Para el registro de una nueva cuenta lo vamos a ver sobre la **Figura 14.6**. En la cual tendremos que completar todos los campos requeridos.

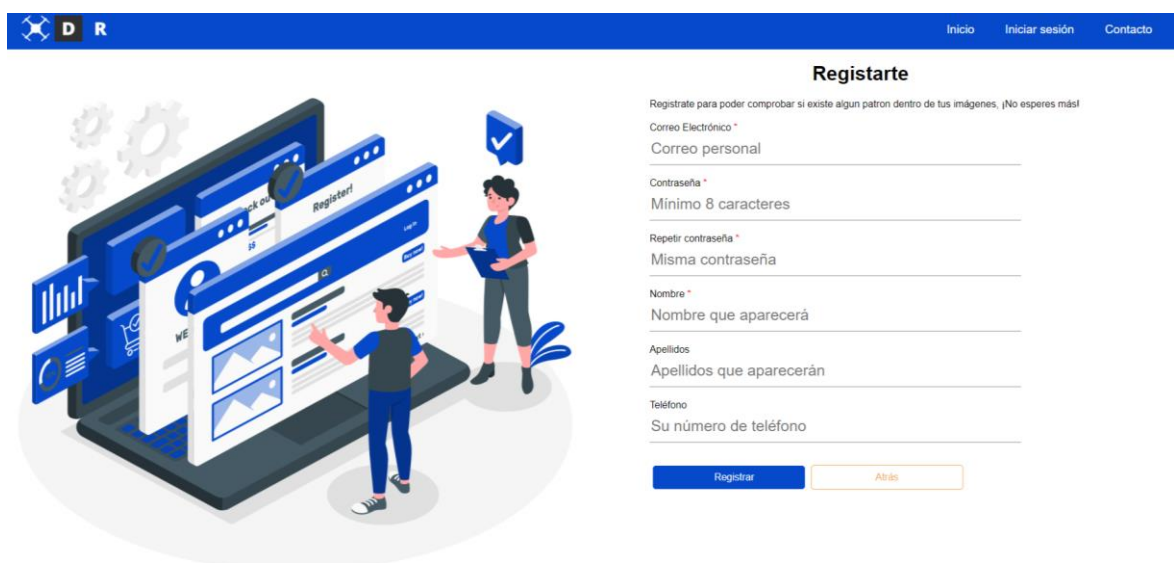


Figura 14.6 Interfaz del registro del usuario.

Aquí tenemos tres campos de texto que realizan una funcionalidad especial:

- **Correo electrónico:** este campo según terminemos de introducir nuestro correo verificará si el mismo es válido. Para ello, se comprobará que no se encuentre registrado en base de datos.
- **Contraseña y repetir contraseña:** estos campos de texto comprueban que las contraseñas sean iguales.

Los botones que componen esta interfaz, como podemos ver el de **“Registrar”**. Si todos los campos que encuentran rellenos y son válidos, con las comprobaciones comentadas anteriormente, nos registra en el sistema. También, tenemos el botón de **“Atrás”**, el cual envía al inicio de sesión.

## 14.2 Página de perfil

Esta es la vista donde tendremos varias acciones de cara al perfil del usuario, las cuales son:

- **Modificar datos personales:** en el lápiz naranja de la derecha, pulsando en él nos enviará a la interfaz de actualización de datos personales del usuario.
- **Cambio de contraseña:** en el botón que aparece al lado de la contraseña que nos enviara a la interfaz de cambio de contraseña.
- **Botón “dar de baja”:** este botón es válido cuando el usuario quiere dejar de utilizar la aplicación, pulsando este botón, tendrá 90 días para deshacer esta acción o su cuenta será eliminada por completo.

Todo esto se puede observar en la **Figura 14.7**.

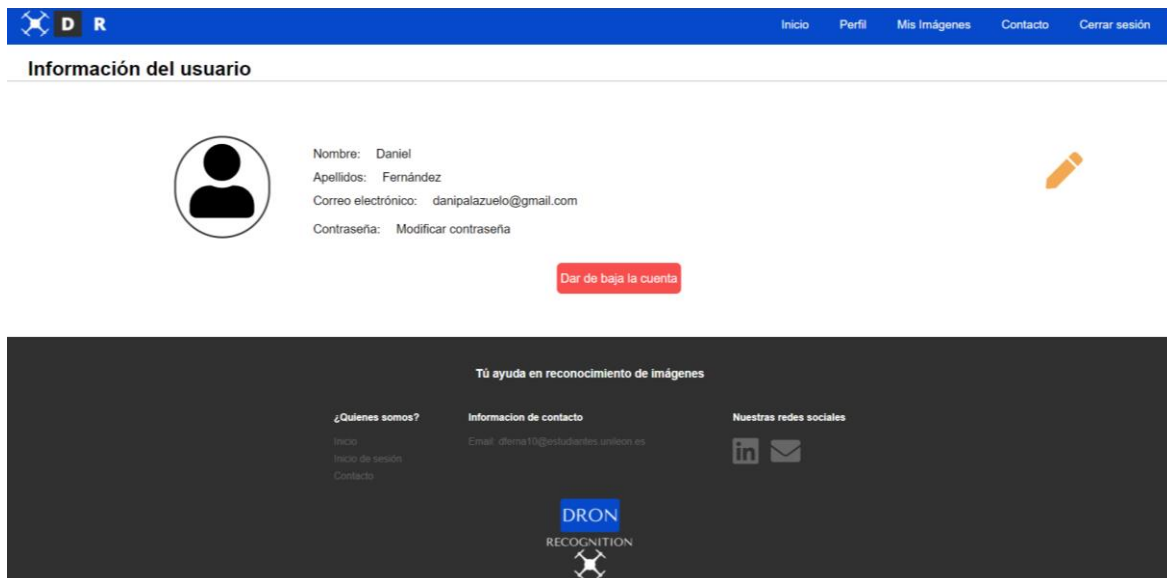


Figura 14.7 Interfaz de perfil del usuario.

### 14.3 Página de Mis Imágenes

Sobre esta interfaz el usuario que utilice la aplicación podrá ver todas aquellas imágenes que haya subido y comprobado en la aplicación web. Desde aquí tenemos varias acciones en la interfaz principal.

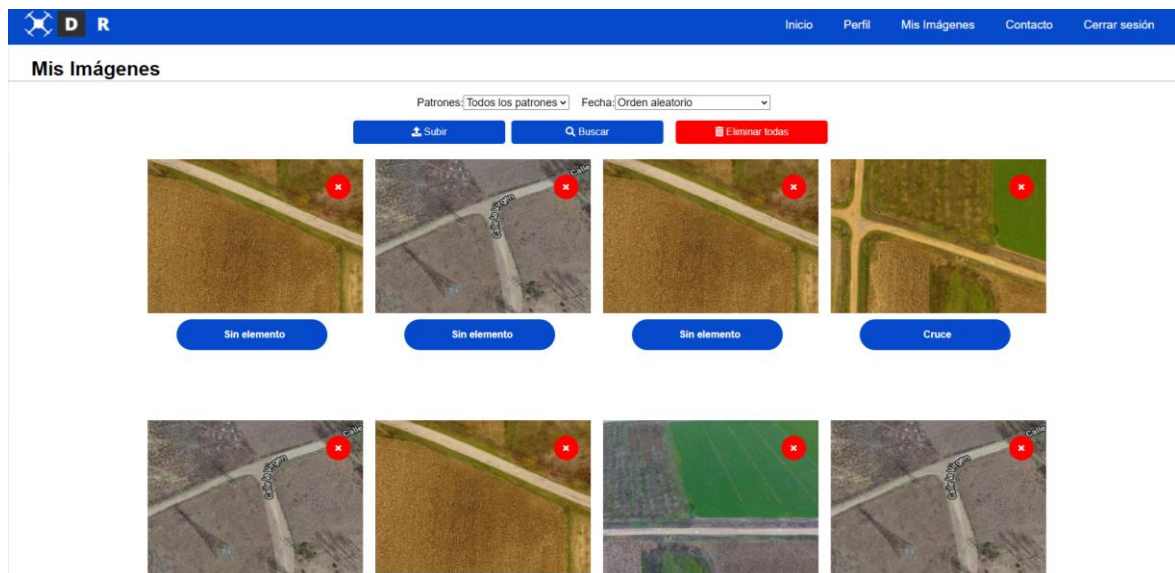


Figura 14.8 Muestra de interfaz de Mis Imágenes.

En la parte superior tenemos dos menús selectores de patrón o de orden por fecha, con esto el usuario puede modificar los datos que quiere que se muestren filtrados en la pantalla.

Por otro lado, nos encontramos tres botones:

- **Subir:** este botón nos enviará a la interfaz de subida de imágenes para así poder añadir nuevas.
- **Buscar:** se usa después de realizar el filtrado para poder aplicar los cambios necesarios en la interfaz.
- **Eliminar todas:** este botón nos sirve para eliminar todas las imágenes del usuario de la base de datos.

Estos menús y botones explicados podemos verlos en la **Figura 14.9**.

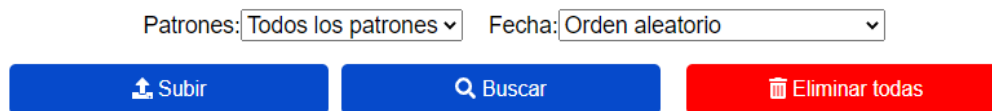


Figura 14.9 Menús y botones de la interfaz de Mis Imágenes

Si al realizar el filtrado, por las características oportunas no se encuentran datos en el sistema registrado, aparecerá en blanco la pantalla.

#### 14.4 Página de Cambio de contraseña

A esta interfaz se puede acceder desde el correo de recuperación de contraseña y desde el perfil del usuario. Desde aquí el usuario puede cambiar su contraseña si se le ha olvidado o quiere introducir una con más seguridad.

Para ello, se pedirá al usuario que introduzca dos veces la misma contraseña para evitar fallos de que se haya confundido en algún elemento.

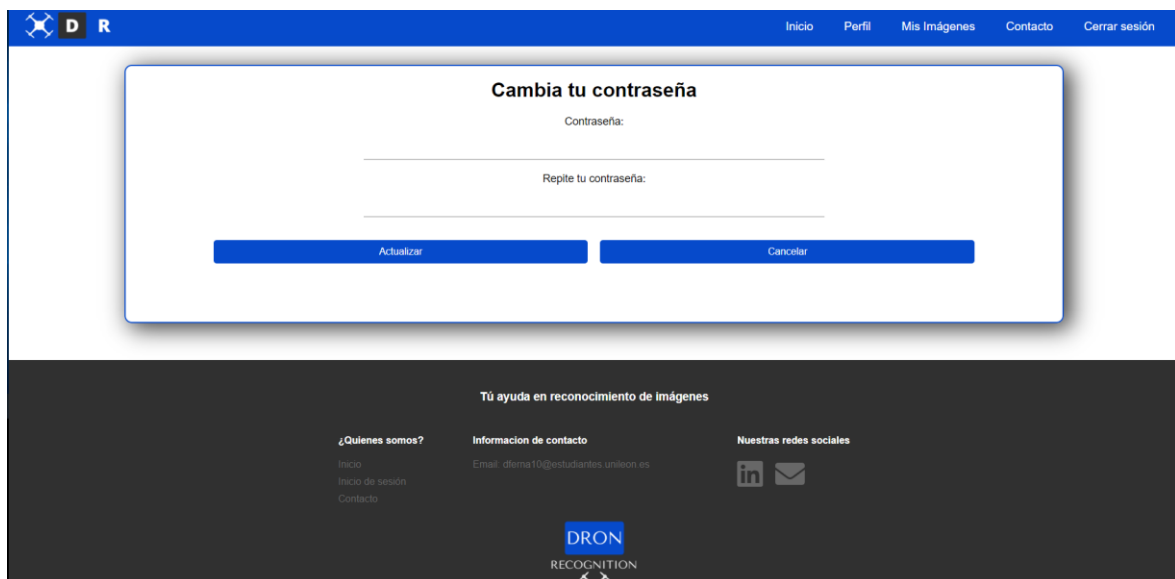
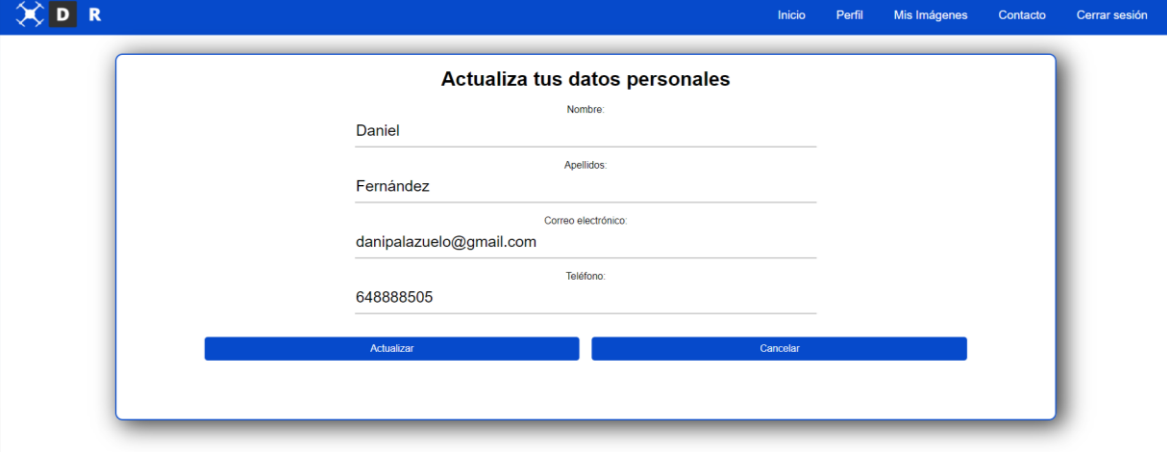


Figura 14.10 Interfaz de cambio de contraseña.

En esta interfaz, tenemos dos posibles acciones que se tratan de los botones de: **“Actualizar”** en el que el usuario cambiara la contraseña o, el de **“Cancelar”** en el que se le redirigirá a la interfaz principal, si se trata de un olvido de contraseña, o a la interfaz de perfil si ha querido cambiar su contraseña.

## 14.5 Página de actualización de datos personales

En esta interfaz el usuario podrá modificar sus datos personales, o empresariales. Para ello se le mostrarán los datos que tiene actualmente registrados en el sistema, y se le dará la opción de cambiarlos.



The screenshot shows a web interface for updating personal data. At the top, there is a blue navigation bar with a logo on the left and links for 'Inicio', 'Perfil', 'Mis imágenes', 'Contacto', and 'Cerrar sesión' on the right. Below the navigation bar is a white form titled 'Actualizar tus datos personales'. The form contains five input fields with the following labels and values: 'Nombre:' with 'Daniel', 'Apellidos:' with 'Fernández', 'Correo electrónico:' with 'danipalazuelo@gmail.com', and 'Teléfono:' with '648888505'. At the bottom of the form are two blue buttons: 'Actualizar' and 'Cancelar'.

Figura 14.11 Interfaz de actualización de los datos personales.

Si el usuario ha accedido a esta interfaz y no quiere realizar algún cambio, puede pulsar el botón de **“Cancelar”** y se le enviará a su perfil de usuario.

Si por el contrario ha modificado alguno de sus datos, al pulsar el botón **“Actualizar”** sus datos se guardarán en el sistema.

## 14.6 Página de contacto

A esta página puede acceder cualquier usuario para ponerse en contacto con Dron Recognition, en la cual se podrán hacer distintas acciones siempre a través de correo electrónico.

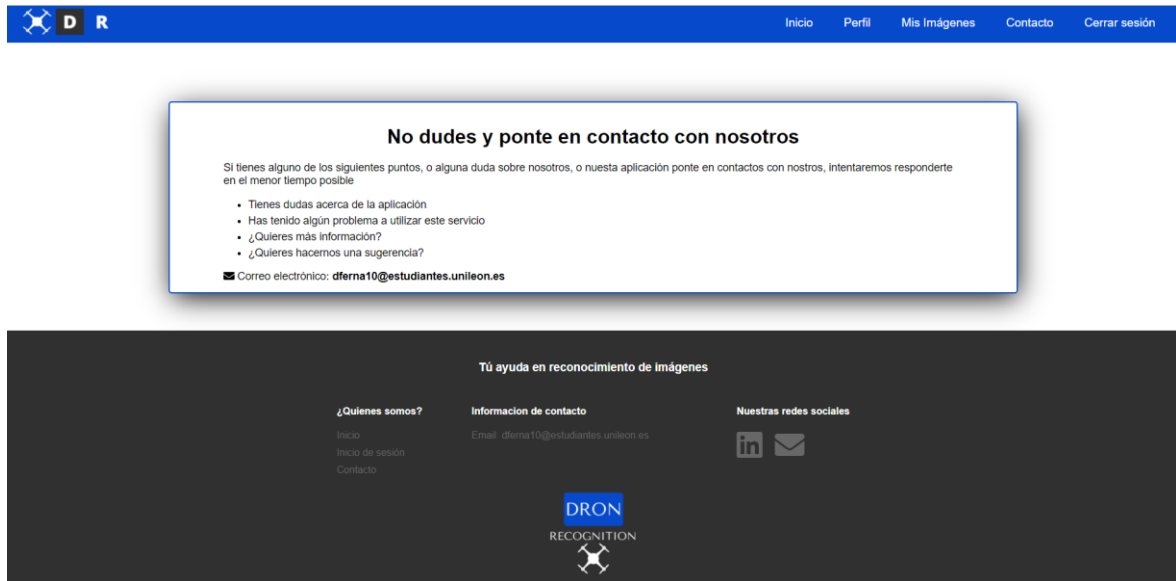


Figura 14.12 Página de contacto con la empresa.

Mediante esta interfaz se aporta al usuario algunas ideas de cómo ponerse en contacto con esta aplicación, aunque pueda ser cualquier opción que se le pueda ocurrir.

Algunas de las que esta interfaz nos marca pueden ser:

- Que tengan dudas acerca de la aplicación.
- Si está teniendo problemas al trabajar con este servicio.
- Si necesita más información para poder trabajar con ello.
- Si quiere hacernos alguna sugerencia de patrones a añadir.



## 15 Agradecimientos

Me gustaría realizar una serie de agradecimientos a todas esas personas que han estado ahí durante toda la carrera, amigos, mi familia y en especial a mis padres, por apoyarme en todo momento.

También agradecer a mis tutores Maite y Javier, por proponerme este proyecto. Estar pendientes y ayudándome con todas aquellas dudas que fueron surgiendo, y todas las facilidades para ir a volar el dron para la captura de datos. Un proyecto que ha sido una maravilla para aprender más sobre Deep Learning y conocer un poco más el pasado León.

A todas y cada una de estas personas, GRACIAS

## 16 Bibliografía

- [1] Jumper, J., Evans, R., Pritzel, A. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021). Available: <https://doi.org/10.1038/s41586-021-03819-2>
- [2] Iván de Paz Centeno, Enrique Alegre Gutiérrez, Víctor González Castro, Óscar García-Olalla Olivera, María Teresa García Ordás, Eduardo Fidalgo Fernández, “Detección de rostro y estimación de edad y género utilizando sistemas de visión portables de bajo coste” Available: <https://buleria.unileon.es/handle/10612/8055>
- [3] M. Saqib, S. Daud Khan, N. Sharma and M. Blumenstein, "A study on detecting drones using deep convolutional neural networks," 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2017, pp. 1-5 Available: <https://ieeexplore.ieee.org/abstract/document/8463284>
- [4] M. Saqib, S. Daud Khan, N. Sharma and M. Blumenstein, "A study on detecting drones using deep convolutional neural networks," 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2017, pp. 1-5 Available: [https://ieeexplore.ieee.org/abstract/document/8078541?casa\\_token=M-Btttd4UekAAAAA:gf\\_n9QEWxb2U87yOtdZVky3j4kuJ2unr\\_Dof5cFD73w8WZcK4haJO5-qYR63dyoXbw3Uc4j1OOrU](https://ieeexplore.ieee.org/abstract/document/8078541?casa_token=M-Btttd4UekAAAAA:gf_n9QEWxb2U87yOtdZVky3j4kuJ2unr_Dof5cFD73w8WZcK4haJO5-qYR63dyoXbw3Uc4j1OOrU)
- [5] S. Yong and Y. Yeong, "Human Object Detection in Forest with Deep Learning based on Drone's Vision," 2018 4th International Conference on Computer and Information Sciences (ICCOINS), 2018, pp. 1-5 Available: [https://ieeexplore.ieee.org/abstract/document/8510564?casa\\_token=FNE1PmiQCkAAAAA:ePCBggJaDC\\_lpeFrFHO3VoBZH7Y-dik2pg1sCBiKC-IGT7B96PeMr-OP6RZUCMfjeIWJx1UI0hnC](https://ieeexplore.ieee.org/abstract/document/8510564?casa_token=FNE1PmiQCkAAAAA:ePCBggJaDC_lpeFrFHO3VoBZH7Y-dik2pg1sCBiKC-IGT7B96PeMr-OP6RZUCMfjeIWJx1UI0hnC)
- [6] Fernández-Lozano, J., Gutiérrez-Alonso, G., & Fernández-Morán, M. Á. (2015). Using airborne LiDAR sensing technology and aerial orthoimages to unravel roman water supply systems and gold works in NW Spain (Eria valley, León). *Journal of Archaeological Science*, **53**, 356-373. Available: [https://www.researchgate.net/publication/268388814\\_Using\\_airborne\\_LiDAR\\_sensing\\_technology\\_and\\_aerial\\_orthoimages\\_to\\_unravel\\_roman\\_water\\_supply\\_systems\\_and\\_gold\\_works\\_in\\_NW\\_Spain\\_Eria\\_valley\\_Leon](https://www.researchgate.net/publication/268388814_Using_airborne_LiDAR_sensing_technology_and_aerial_orthoimages_to_unravel_roman_water_supply_systems_and_gold_works_in_NW_Spain_Eria_valley_Leon)