

Trabajo de Final de Grado

# Aplicación para vigilancia de entornos agrícolas con UAVs y Procesado de Imagen(Machine Learning)

**Autor:** Jonathan Sanchez Murillo

**Tutor:** José Antonio Soria Pérez

Convocatoria Otoño 2021



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH

---

Escola Superior d'Enginyeries Industrial,  
Aeroespacial i Audiovisual de Terrassa

Grado en Ingeniería Electrónica Industrial y Automática



*“De vez en cuando, una nueva tecnología, un antiguo problema y una gran idea se convierten en una innovación»”*

Dean Kamen



## Resumen

La finalidad de este trabajo de fin de grado era montar un dron y adaptarlo para el uso de ciberseguridad en los campos de cultivo. Esto se realiza mediante la adquisición de imágenes aéreas y el posterior envío automatizado mediante una unidad terrestre remota, en mi caso una Raspberry pi 4. El dron tiene la capacidad de volar autónomamente programando la ruta que queremos que realice.

En este documento se explica desde como ensamblar el dron, hasta la programación y desarrollo de una red neuronal convolucional para la detección de personas en las fotos que envía el dron mediante la unidad terrestre.

Se puede dividir el proyecto en cuatro grandes partes, la primera nos introduce en el mundo de los drones, aplicaciones que tienen en la actualidad y mas en concreto en la agricultura.

La segunda parte consta de una descripción de los componentes que conforman el dron y una descripción del proceso de ensamblaje, configuración y calibración del dron.

La tercera parte consta del desarrollo de una aplicación web, en mi caso la modificación y adaptación de una desarrollada por un compañero, y la implementación de la misma para poder usar la red neuronal desarrollada para el procesado de las imágenes recibidas a través del servidor.

Para finalizar, se muestran los resultados y las conclusiones obtenidas gracias a las pruebas que se han realizado.

## **Abstract**

The purpose of this final degree project was to assemble a drone and adapt it for cybersecurity in crop fields. This is done through the acquisition of aerial images and the subsequent automated sending through a remote ground unit, in my case a Raspberry pi 4. The drone has the ability to fly autonomously by programming the route we want it to take.

This document explains from how to assemble the drone, to the programming and development of a convolutional neural network for the detection of people in the photos sent by the drone through the ground unit.

The project can be divided into four main parts, the first of which introduces us to the world of drones and their current applications, particularly in agriculture.

The second part consists of a description of the components that make up the drone and a description of the assembly process, configuration and calibration of the drone.

The third part consists of the development of a web application, in my case the modification and adaptation of one developed by a colleague, and the implementation of the same in order to use the neural network developed for the processing of the images received through the server.

Finally, the results and conclusions obtained thanks to the tests that have been carried out are shown.

# Índice general

<b>1</b>	<b>Introducción</b>	<b>9</b>
1.1	Antecedentes . . . . .	9
1.2	Objeto . . . . .	9
1.3	Alcance . . . . .	10
1.4	Requerimientos . . . . .	10
1.5	Justificación . . . . .	12
1.6	Contenidos de la memoria . . . . .	13
<b>2</b>	<b>Estado del arte</b>	<b>15</b>
2.1	Que es un dron? . . . . .	15
2.2	Clasificación de los UAV . . . . .	15
2.3	Aplicaciones de los drones . . . . .	17
2.3.1	Drones en la agricultura . . . . .	18
2.4	Control de vuelo . . . . .	20
2.5	Componentes . . . . .	22
2.5.1	Chasis . . . . .	22
2.5.2	Tren de aterrizaje . . . . .	23
2.5.3	Hélices . . . . .	23
2.5.4	Motores . . . . .	24
2.5.5	Controlador de vuelo . . . . .	25
2.5.6	Variadores (ESC) . . . . .	26
2.5.7	Receptor . . . . .	26
2.5.8	Telemetría . . . . .	27
2.5.9	Emisora . . . . .	27
2.5.10	Batería . . . . .	27
2.6	Modos de vuelo . . . . .	28
2.7	Normativa drone . . . . .	29
2.8	Inteligencia artificial . . . . .	32
2.8.1	Machine Learning . . . . .	32
2.8.2	Deep Learning . . . . .	32
2.8.3	Segmentación por instancias de imágenes . . . . .	33
2.8.4	Fast R-CNN, Faster R-CNN y Mask R-CNN . . . . .	34
2.9	Historia de la IA hasta las redes neuronales artificiales . . . . .	36
2.10	Antecedentes de la visión artificial . . . . .	37

<b>3</b>	<b>Montaje y configuración del dron</b>	<b>39</b>
3.1	Componentes . . . . .	39
3.1.1	Frame . . . . .	39
3.1.2	Motor . . . . .	40
3.1.3	ESC . . . . .	41
3.1.4	Hélices . . . . .	42
3.1.5	Batería . . . . .	42
3.1.6	Controladora de vuelo . . . . .	43
3.1.7	Transmisor . . . . .	45
3.1.8	Receptor . . . . .	47
3.1.9	Módulo de telemetría . . . . .	48
3.1.10	GPS . . . . .	49
3.1.11	Módulo de potencia . . . . .	49
3.1.12	Gimball . . . . .	50
3.2	<i>Firmware</i> de vuelo . . . . .	50
3.2.1	Estación de control terrestre: <i>Mission Planner</i> . . . . .	50
3.3	Ensamblaje . . . . .	51
3.3.1	Ensamblaje del frame y tren de aterrizaje . . . . .	51
3.3.2	Montaje de los motores y variadores . . . . .	53
3.3.3	Soldadura y conexión e instalación de la electrónica a la controladora de vuelo . . . . .	54
3.4	Configuración del dron . . . . .	56
3.4.1	Comprobación de la configuración de rotación de las hélices . . . . .	56
3.4.2	Conexión e instalación del firmware . . . . .	56
3.4.3	Calibración de los ESC . . . . .	57
3.4.4	Calibración mediante Mission Planner . . . . .	58
3.5	Comunicación GoPro-Servidor . . . . .	62
3.5.1	Programación del script de la Raspberry Pi 4 . . . . .	63
3.5.2	Programación del script del Servidor ejecutado en el Ordenador . . . . .	64
<b>4</b>	<b>Machine Learning</b>	<b>69</b>
4.1	¿Que es el aprendizaje automático? . . . . .	69
4.1.1	Aprendizaje profundo . . . . .	70
4.2	Redes Neuronales Convolucionales . . . . .	71
4.2.1	Convoluciones . . . . .	71
4.2.2	Función de activación . . . . .	73
4.2.3	Submuestreo o subsampling . . . . .	73
4.2.4	Muestreo con Max-Pooling . . . . .	73
4.2.5	Clasificación . . . . .	73
4.3	Implementación . . . . .	74
4.3.1	Tecnologías usadas para la implementación . . . . .	74
4.3.2	Datasets utilizados . . . . .	75
4.3.3	Selección de clases a detectar . . . . .	75
4.3.4	Entrenamiento de la R-CNN . . . . .	75

4.3.5	Métricas de evaluación . . . . .	76
4.4	Desarrollo del código . . . . .	78
4.4.1	Definición del modelo . . . . .	78
4.4.2	Definición del dataset . . . . .	78
4.4.3	Parámetros de la CNN . . . . .	79
<b>5</b>	<b>Resultados</b>	<b>81</b>
5.1	Primera prueba de vuelo . . . . .	81
5.2	Segunda prueba de vuelo . . . . .	82
5.3	Tercera prueba de vuelo . . . . .	82
5.4	Cuarta prueba de vuelo . . . . .	85
5.5	Procedimientos para realizar vuelos . . . . .	87
5.6	Análisis de los registros de datos . . . . .	87
5.6.1	Análisis automático de datos de telemetría . . . . .	88
5.6.2	Calidad del GPS . . . . .	89
5.6.3	Altura . . . . .	90
5.7	Comunicación GoPro-Servidor . . . . .	90
5.8	Procesado de imagen . . . . .	91
5.8.1	Matriz de confusión . . . . .	91
5.8.2	Detección correcta de personas . . . . .	94
5.8.3	Detección parcial en una imagen . . . . .	94
5.8.4	Detección incorrecta a causa de condiciones desfavorables . . . . .	95
<b>6</b>	<b>Conclusiones y futuros trabajos</b>	<b>97</b>
6.1	Conclusiones . . . . .	97
6.2	Futuros trabajos . . . . .	97
	<b>Bibliografía</b>	<b>99</b>



# Índice de figuras

1.1	Aeronave JMT . . . . .	10
1.2	Máscara para la detección de personas . . . . .	11
2.1	Clasificación dron . . . . .	16
2.2	Dron pulverizador . . . . .	20
2.3	Sentido de las hélices . . . . .	21
2.4	PID dron . . . . .	21
2.5	Ejemplo de chasis UAV diseñado con Generative design . . . . .	23
2.6	Diferentes tipos de hélices . . . . .	24
2.7	Motor BLDC . . . . .	24
2.8	Ejemplo Ardupilot . . . . .	26
2.9	Ejemplos de ESC . . . . .	26
2.10	Ejemplo de telemetría . . . . .	27
2.11	Control de dron . . . . .	29
2.12	Relación de pertenencia entre la inteligencia artificial, el aprendizaje automático y el aprendizaje profundo . . . . .	32
2.13	Clasificación de objetos . . . . .	33
2.14	Detección de objetos . . . . .	33
2.15	Segmentación semántica . . . . .	34
2.16	Segmentación de instancias . . . . .	34
2.17	Comparativa Arquitectura Fast R-CNN vs Faster R-CNN . . . . .	35
2.18	Arquitectura Mask R-CNN . . . . .	35
2.19	Arquitectura Mask R-CNN . . . . .	36
3.1	Dimensiones del frame . . . . .	40
3.2	Conexión de las ESC . . . . .	41
3.3	ESC . . . . .	42
3.4	Materia LiPo . . . . .	43
3.5	APM 2.8 y módulo GPS . . . . .	44
3.6	Parte interna del APM 2.8 . . . . .	44
3.7	Transmisor . . . . .	46
3.8	Modulación QPSK . . . . .	46
3.9	Receptor R9DS . . . . .	47
3.10	Módulos de telemetría y su cableado . . . . .	48
3.11	GPS . . . . .	49
3.12	Módulo de potencia . . . . .	49
3.13	Gimball . . . . .	50

## Índice de figuras

3.14	Interfaz de usuario principal . . . . .	51
3.15	Frame con el tren de aterrizaje ensamblado . . . . .	52
3.16	Placa de potencia . . . . .	53
3.17	Soldadura y protección con termoretráctil de los conectores . . . . .	54
3.18	Soldadura de los variadores a la placa de potencia . . . . .	55
3.19	Diagrama de conexionado al APM . . . . .	55
3.20	Conexionado variadores al APM . . . . .	56
3.21	Conexión de la APM con Mission Planner . . . . .	56
3.22	Instalación del <i>firmware</i> . . . . .	57
3.23	Selección del tipo de frame para la calibración . . . . .	58
3.24	Posiciones de calibración de el acelerómetro . . . . .	58
3.25	Calibración del GPS . . . . .	59
3.26	Calibración del transmisor . . . . .	60
3.27	Modo Auto . . . . .	62
3.28	Comunicación GoPro-servidor . . . . .	63
3.29	Flujo de control MVC . . . . .	64
3.30	Página 'Iniciar sesión' . . . . .	66
3.31	Página 'Sube una imagen' . . . . .	67
3.32	Página 'Inicio' . . . . .	68
4.1	Programación clásica vs Machine learning . . . . .	70
4.2	Proceso Machine Learning vs Deep learning . . . . .	71
4.3	Convolución . . . . .	72
4.4	Ejemplo de aplicación de la función de activación ReLu . . . . .	73
4.5	Ejemplo de Max-Pooling . . . . .	74
4.6	Proceso que sigue una imagen en una CNN hasta ser procesada . . . . .	74
4.7	Ejemplo de imagen etiquetada . . . . .	76
4.8	Cálculo de la métrica IoU . . . . .	78
5.1	Como armar drone . . . . .	82
5.2	Zona de pruebas de vuelo . . . . .	83
5.3	Plan de vuelo . . . . .	84
5.4	Error Mission Planner . . . . .	85
5.5	Plan de vuelo . . . . .	85
5.6	Imagen aérea . . . . .	86
5.7	Imagen aérea del terreno . . . . .	87
5.8	Ruta realizada en 3D . . . . .	88
5.9	Log Analyzer . . . . .	89
5.10	GPS Hdop y número de satelites . . . . .	90
5.11	Altura de vuelo . . . . .	91
5.12	Página principal del servidor . . . . .	92
5.13	Detección de una única persona . . . . .	94
5.14	Detección parcial de personas . . . . .	95

5.15 Detección incorrecta a causa de condiciones desfavorables . . . . . 95



# Índice de cuadros

2.2	Clasificación del dron según AESA . . . . .	17
2.3	Normativa de subcategorías . . . . .	31
3.1	Características del frame . . . . .	39
3.2	Especificaciones del motor . . . . .	40
3.3	Especificaciones de los variadores . . . . .	41
3.4	Especificaciones de la batería . . . . .	43
3.5	Especificaciones del transmisor . . . . .	47
3.6	Características del receptor . . . . .	47
3.7	Especificaciones del módulo de telemetría . . . . .	48
3.8	Estructura de la base de datos . . . . .	65
4.1	Matriz de confusión . . . . .	77
5.1	Datos del vuelo . . . . .	86
5.2	Matriz de confusión . . . . .	92



# Lista de abreviaturas

BEC	Batery Eliminator Circuit
BLDC	Brushless DC Motor
CA	Corrent Altern
CC	Corrent Continu
CNN	Convolutional Neural Networks
ESC	Electronic Speed Control
GPU	Graphics Processing Unit
HTTP	Hypertext Transfer Protocol
I/O	Input/Output
I2C	Inter-Integrated Circuit
IA	Inteligencia Artificial
IoU	Intersection over Union
MVC	Modelo-Vista-Controlador
PWM	Pulse Width Modulation
RC	Radio Control
RCNN	Region Based Convolutional Neural Networks
RF	Radio Frecuencia
rpm	Revoluciones por minuto
SPI	Serial Peripheral Interface



# 1 Introducción

## 1.1. Antecedentes

Actualmente, la agricultura se encuentra en el punto de búsqueda de nuevas tecnologías modernas que puedan aplicarse para mejorar la producción y rendimiento, manteniendo los ecosistemas sin generar gastos en la motorización de los campos de cultivo y residuos innecesarios[1]. Hasta la actualidad, el control y la supervisión de los campos de cultivo de superficies extensas ha sido una tarea muy compleja, ya que no hay ninguna tecnología aplicada que ayude al agricultor a decidir cuando tiene que sulfatar o aplicar tratamientos a los cultivos que no sea recorrer todos los terrenos para tenerlos bajo control, cosa que para recorrer grandes superficies se utilizan tractores por ejemplo, y por lo tanto contaminamos el ecosistema porque no tenemos otra manera de realizar este proceso si no es de esta manera[2]. Por lo que, es obvio que los agricultores están en una situación de necesidad de nuevas tecnologías para poder satisfacer la gran demanda mundial de diferentes tipos de productos para poder facilitar la producción de estos en procesos como puede ser el riego o estado de maduración[3].

En la actualidad, según estudios realizados se prever un crecimiento demográfico de entre 7.000 y 9.000 millones de habitantes entre 2010 y 2050, por lo que la producción agrícola tendría que crecer para poder satisfacer este crecimiento. Esto genera una problemática entre los agricultores, ya que no pueden conocer la demanda y por lo tanto, tomen decisiones que a corto y medio plazo pueden afectar en el desabastecimiento de según que productos, pérdidas económicas y un posible impacto ambiental negativo en sus cultivos.

El desarrollo de herramientas basadas en Tecnologías de la Información y Comunicaciones (TIC) están bastante presentes en este ámbito y destinadas a mejorar estos problemas con información precisa y fiable, aportando un papel muy importante en el futuro de la agricultura.

## 1.2. Objeto

Este trabajo tiene como horizonte el desarrollo de un dron cuadrilátero para tareas de video vigilancia de terrenos agrícolas. En este, se pretende obtener imágenes aéreas para enviarlas a un servidor web que implementa técnicas de Inteligencia Artificial (IS) para marcar personas en la imagen.

### 1.3. Alcance

Para este fin, se ha planteado un proyecto multidisciplinario de varios estudiantes. Concretamente, el trabajo de este proyecto se centra exclusivamente en el procesado de las imágenes obtenidas desde la aeronave mediante técnicas de IS. en la unidad harreare computacional terrestre. En el trabajo de A. de Blas *Aplicación de Vigilancia de Entornos agrícolas con Uvas y Procesado de Imagen (Desarrollo Servidor Web)* [4].

Partiendo de una aeronave construida a partir de un kit comercial estándar, y a la que se puede programar planes de vuelo para supervisar un entorno agrícola, el trabajo se centra en conseguir imágenes con personas, etiquetarlas para que, de manera bolinee se entrene una red neuronal para detectar las personas en la imagen, así como la posición.

El trabajo que expone este documento se ciñe en incorporar la cámara de video a la aeronave para poder realizar instantáneas, etiquetar las imágenes que se obtienen para poder realizar el aprendizaje de la red neuronal involuciona, y realizar la inferencia correspondiente para calcular la precisión de la aplicación.

### 1.4. Requerimientos

En cuanto a la estructura mecánica de la aeronave, se ha adquirido un quiróptero de la marca MTRO con piezas de fibra de carbono que lo hacen ligero y al mismo tiempo resistente a golpes e impactos moderados (Figura 1.1). El paquete incluye un controlador de vuelo AP 2.8.0 de Pilotar con figurable mediante la aplicación Mission Planear. Este controlador, además de gestionar el control del vuelo automático, con Pida para la gestión de la altura así como la geolocalización GPS (Global Positioning System), dispone de puertos digitales de comunicación I2C/SPI, I/O para conectar «gimball» de la cámara de video o GPS - entre otros sensores .



**Figura 1.1:** Aeronave JMT

La aeronave funciona con una batería de 11.1V y 6000 mAh de ion-litio tipo 3S y motores JMT HYD 3508 sin escobillas que garantizan una duración de vuelo

de unos 20 minutos aproximadamente. El control de vuelo manual se realiza con un transmisor de la marca RadioLink, -modelo AT9S -que cuenta con 9 canales localizados cerca de la banda de de 2.4GHz, y un receptor que hay que poner en la aeronave. En cuanto a la telemetría se utiliza transceptores RF de 433MHz de 0.1W que permite enviar los datos de vuelo de la aeronave en un rango de 4 Km según especificaciones.

A la aeronave se le incorpora una cámara un gimball con GoPro 4 que puede obtener imágenes con una resolución 4000 x 3000 a 12 Megapíxeles. Se han realizado un total de 20 instantáneas con la aeronave a diferentes alturas, distancias y posiciones, y en entornos diferentes con el objetivo de fotografiar personas.

Con las imágenes, se desarrolla una aplicación IA típica de aprendizaje supervisado. Esta aproximación consiste en configurar el modelo a partir de ejemplos, generando manualmente las máscaras para cada persona en todas las fotografías (Figura 1.2). De este modo se le indica a la red neuronal la tarea que debe aprender a realizar. Para ello, un grupo importante de instantáneas se reserva para el aprendizaje del modelo, mientras que el resto sirve para realizar una estimación de la precisión que tendría el modelo en producción, una vez finalizado el aprendizaje. Las fases que conlleva este proceso son las siguientes:



**Figura 1.2:** Máscara para la detección de personas

1. **Ingeniería de datos.-** En esta fase, se preparan las instantáneas, etiquetándolas adecuadamente para la fase de entrenamiento. Se ha utilizado la herramienta LabelMe para este fin , etiquetando cada una de las personas que aparecen en cada instantánea. En este sentido, pueden haber desde fotografías que no contienen ninguna persona hasta otras en las que aparecen varias. Así se puede realizar el aprendizaje y evaluar la precisión del detector en diferentes situaciones. Las instantáneas se han obtenido con el permiso correspondiente de las personas implicadas.

2. **Modelo de clasificación.-** Se escoge un modelo de red neuronal con filtros convolucionales. Se sabe que el filtro por convolución es muy empleado en aplicaciones de tratamiento de imagen para resaltar rasgos y patrones, facilitando así la detección de estructuras. Para este trabajo, se ha escogido Mask-RCNN del repositorio de Matterport disponible en Github[5] . Este es un modelo planteado en 2017 y que, recientemente, ha mostrado buenos resultados en otros ámbitos de procesamiento de imágenes.
3. **Conjuntos de entrenamiento, validación y test.** - Se han dividido las imágenes en tres grupos: 1) entrenamiento (*training*); 2) validación (*validation*); y 3) test. El primero se usa para la configuración de los parámetros del modelo y contiene la mayoría de las imágenes (75 % de las imágenes), actualizándolos en base al cálculo de un error que se obtiene a partir de las etiquetas. Para que el algoritmo sepa si está aprendiendo adecuadamente, se necesita otro conjunto (validación) que realice pequeñas inferencias durante el aprendizaje. Este contiene el 15 % del total de las imágenes. Una vez entrenado el modelo, se requiere de otro conjunto (test) para estimar la precisión y predecir el comportamiento del detector después del entrenamiento, una vez se ponga en marcha durante la fase de producción.
4. **Producción.-** Con la red entrenada, el último paso consiste en introducir las imágenes del conjunto de test para que la red marque las personas calculando las máscaras por sí sola. Estas se comparan con las previamente etiquetadas, a partir de la matriz de confusión, para obtener estimar la precisión de detección.

Para estos pasos, se configura Mask-RCNN para entrenar durante 100 épocas (*epochs*) con un conjunto de entrenamiento de 129 imágenes. Éste se ha realizado con el entorno de la plataforma Google Collab que incluye GPU (Graphic Processor Unit) para acelerar el cálculo, habiendo requerido un total de 12 horas , 934s/epochs, 2s/step teniendo un total de 500 steps.

Dado que se trata de un conjunto de imágenes pequeño, para mitigar la varianza traslacional del clasificador se han generado imágenes sintéticas a partir de las obtenidas, rotando el contenido aleatoriamente cierto ángulo aleatorio, emulando el hecho de que el dron puede tomar instantáneas desde cualquier posición. Consiguiendo una precisión de detección del 94.44 %.

## 1.5. Justificación

La motivación sobre este proyecto se basa por el querer conocer un poco más sobre el uso de los drones en la industria, y que se está expandiendo en el ámbito de la agricultura. Entre las que destacan, el uso de vehículos aéreos no tripulados para la gestión del agua, tratamientos localizados con herbicidas y fertilizantes, detección temprana de enfermedades y plagas en cultivos supervisión y vigilancia, y que ac-

tualmente cobran importancia como indicadores de calidad en cultivos, la generación de inventarios de cultivos, o el control de subvenciones agrarias[?]

En la actualidad, el término Smart-Agriculture la tecnología ha cambiado el concepto de la agricultura, y trata de hacerla más rentable, eficiente, segura y simple. Se obtienen importantes beneficios gracias a la tecnología de la agricultura digital y su constante evolución. De esta manera se convierte en un negocio rentable, inteligente y sostenible. Los drones son una tecnología fácil de usar y son capaces de recoger grandes cantidades de datos en intervalos cortos de tiempo. Los hurtos en zonas de cultivo cada vez son más habituales y el problema surge cuando el agricultor ve mermados sus ingresos por culpa de quienes acuden a sus cultivos para sustraer una pequeña cantidad de productos para su auto consumo o para venderla en diferentes comercios.

Surge la idea de crear una aplicación de vigilancia aérea de parcelas con sus cosechas. A pesar de que es una industria todavía muy madura y en desarrollo que todavía le falta por crecer, es una buena oportunidad para obtener conocimientos y entrar en la temática.

Finalmente, una motivación muy importante para nosotros, estudiantes del Grado de ingeniería electrónica industrial y automática, es la de poder aplicar varias disciplinas unificadas en un solo proyecto. El ensamblaje del dron junto con sus componentes electrónicos, la soldadura del cableado, y testeo de los diferentes Componentes. Y finalmente la parte de la programación, que durante la carrera hemos obtenido nociones de lenguajes como C++, Java, y con este proyecto hemos conseguido adentrarnos un poco más aprendiendo Python para la programación de los Scripts y del servidor web.

## 1.6. Contenidos de la memoria

Este capítulo ha repasado las líneas generales del proyecto. El presente documento se centra en el ámbito de Machine Learning requerido para esta aplicación y se divide en cinco capítulos. El siguiente capítulo está reservado al estado del arte de los UAVs , el sector agrícola y Machine learning. Se dan a conocer las herramientas mas importantes disponibles en cuanto al uso de UAVs, tanto en lo que se refiere a hardware como al software, así como el avance producido en los métodos de entrenamiento de las redes neuronales que ha dado lugar al Deep Learning .EL tercer capítulo describe el montaje y la configuración del dron, así como los scripts utilizados para entrenar la red neuronal en la detección de personas en imágenes , haciendo una breve mención a la aplicación web aplicación web que complementa a este trabajo.

El cuarto capítulo expone los resultados, haciendo breve mención a las pruebas de vuelo y enfocándose en la precisión del detector, discutiendo los mismos en cuanto al ámbito de aplicaciones de video vigilancia en la agricultura.

## *Introducción*

El último capítulo es una reflexión sobre el proyecto, se habla de lo que se ha aprendido, y se discuten cuáles deberían de ser las prioridades a futuro para desarrollar otras aplicaciones o mejorar la actual.

También se incluye en el proyecto documentos como el presupuesto, los scripts del servidor web, el script de entrenamiento de la red neuronal e inferencia de Mask-RCNN.

## 2 Estado del arte

### 2.1. Que es un dron?

Un dron se puede denominar de diferentes maneras[?]:

- UAV / VANT (Unmanned Aerial Vehicle o Vehículo aéreo no tripulado) que hace referencia a cualquier dispositivo que pueda volar sin tripulación.
- Drone es el sinónimo parcial de UAS, siendo un sistema aéreo no tripulado que es usado para una función concreta.
- UAS (Unmanned Aerial System) es el conjunto de componentes.
- RPA (Remotely Piloted Aircraft) hace referencia al avión es controlado de manera remota por una persona física.
- Multicópteros son aeronaves con varios rotores que hace posible el despegue vertical.

El dron hace referencia a una aeronave que vuela sin tripulación, reutilizable, capaz de mantener de manera autónoma un nivel de vuelo controlado y sostenido, y propulsado por un motor de explosión, eléctrico o de reacción[?].

El primer dron del que se tiene constancia es del año 1922 y fue construido por George de Bothezat, a pesar de que no fue posible levantarlo a más de cinco metros del suelo. Existen UAV de uso civil como comercial, pero desde un principio fue utilizado en aplicaciones militares.

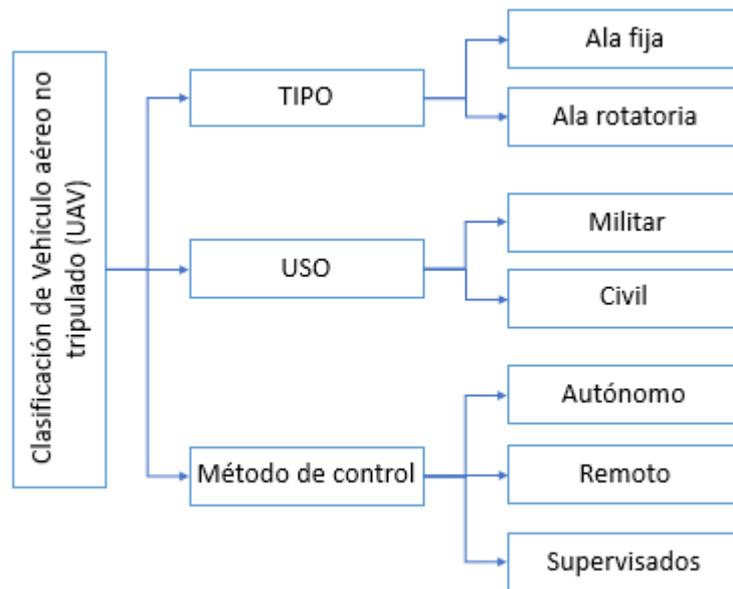
Las ventajas más frecuentes del drone son la posibilidad de usar en áreas de alto riesgo o de difícil acceso, no requiere ninguna actuación de pilotos en la zona de combate, favorece al sector industrial al poder ser utilizado en muchos procesos productivos, reducen el riesgo humano, conllevan a una contaminación menor, se necesita un tipo de formación reducido, son bastante precisos y tienen mucha versatilidad.

### 2.2. Clasificación de los UAV

Los drones se pueden clasificar de distintas maneras (véase en la figura 2.1), dependiendo de varios factores. La primera clasificación que se puede realizar es dependiendo del uso, drones de uso civil (desde juguetes hasta utilizados para uso comercial) o drones de uso militar, estos son conocidos como UCAV (Unmanned

Combat Air Vehicle). También posible clasificarlos por el método de control, según si es autónomo, remoto o supervisado. Otra forma distinta de clasificarlos es por el tipo de ala:

- Ala rotatoria, son el tipo de dron más extendido y utilizado porque se controlan mejor y son capaces de soportar una carga mayor. En este tipo se encuentran los tricópteros, cuadricópteros, hexacópteros y octacópteros.
- Ala fija, son capaces de aprovechar el aire y generar fuerzas que les permite mantenerse en el aire aprovechando la aerodinámica del diseño. Se caracterizan por presentar una arquitectura sencilla y una durada de vuelo mucho más larga.



**Figura 2.1:** Clasificación dron

Según AESA (Agencia Estatal de Seguridad Aérea) existe una nueva clasificación de aeronaves no tripuladas, mediante una etiqueta de identificación de clase. Existen 7 clases de drones C0, C1, C2, C3, C4, C5 y C6 (véase en el cuadro 2.2) . Sus diferencias van en función de las restricciones operacionales que tienen y los sistemas que integran. Estas distinciones de clases tienen que ver principalmente con la masa máxima al despegue (MTOW) del dron, su velocidad máxima o su altura máxima alcanzable por encima del puente de despegue[?].

Clase	MTOW	Velocidad máxima	Altura máxima alcanzable	Característica especial
C0	> 250 g	19 m/s	120 m	-
C1	> 900 g	19 m/s	120 m	-
C2	> 4 kg	3 m/s	120 m	-
C3	> 25 kg y una envergadura inferior a 3 m	-	120 m	-
C4	> 25 kg, incluida la carga útil	-	-	Destinados para la práctica del aeromodelismo
C5	> 25 kg	5 m/s	Disponer un sistema que proporcione distancia información clara y concisa	No ser una aeronave de ala fija, salvo si es una UA cautiva.
C6	> 25 kg	50 m/s	Tener un sistema que proporcione al piloto a distancia información clara y concisa sobre la altura del dron.	-

Cuadro 2.2: Clasificación del dron según AESA

Equipar a los drones con accesorios extra puede suponer un cambio de clase.

## 2.3. Aplicaciones de los drones

En la actualidad tienen diferentes funciones que son esenciales dentro de la sociedad, desde usos comerciales hasta el rescate aéreo. Su diseño compuesto por cámaras, dispositivo GPS y sensores de cualquier tipo, fue desarrollado principalmente para

uso militar, tanto misiones de espionaje hasta portadores de misiles de guerra con la finalidad de disparar contra objetivos teledirigidos[?].

Gracias al descenso de los precios de fabricación, fue ampliado el uso y aplicaciones de los UAV y podemos disfrutar del desarrollo de nuevas tecnologías para la investigación científica o entretenimiento. Los drone son controlados remotamente por mandos o aplicaciones para smartphone/tablets.

Existe una gran variedad de aplicaciones para dron que hasta el momento han sido aplicados como:

1. Filmar eventos mostrando los mejores ángulos gracias a que pueden volar a una altura inferior y a una gran maniobrabilidad respecto a un brazo de grúa.
2. Drones de reparto haciendo envíos por la zona
3. Drone de emergencias que destacan por su efectividad en situaciones límites, especialmente en zonas que han sido aisladas o de difícil acceso;
4. Búsqueda de personas que permite volar a poca altura con un dispositivo fotográfico que transmita a tiempo real
5. Drone vigilancia muy utilizados a través de la Guardia Civil
6. Control de incendios forestales con la finalidad de anticipación en la prevención y expansión de incendios
7. Búsqueda y analítica de restos arqueológicos a través de grandes recorridos mediante la toma de fotografías de la zona
8. Drones capaces de acceder a zonas peligrosas para predecir erupciones, eventos geológicos o incluso para registrar vuelos con GPS para las investigaciones biológicas
9. Drones manipuladores de materiales nocivos, que son tareas demasiado peligrosas para el ser humano
10. Drones capaces de crear redes de internet en áreas donde no hay acceso a ellas.

El uso de drones permite a diferentes sectores crear nuevos modelos de negocio. Gracias al constante desarrollo de nuevas tecnologías es posible encontrar todo tipo de soluciones para aplicaciones de los UAV. Se estima que para el año de 2050 los vehículos no tripulados podrán generar un total de 150.000 empleos[6].

### **2.3.1. Drones en la agricultura**

Los drone son un gran avance en la tecnología, y conlleva a unas grandes ventajas en el sector de la agricultura[?]. Son relativamente económicos, con tiempos de vuelos suficientes para recorrer grandes zonas agrícolas en poco tiempo, con cámaras de visión de alta definición capaces de observar y analizar bastante con precisión. Estos dispositivos pueden incorporar cámaras con visión nocturna capaces de poder tomar

imágenes nocturnas y procesarlas. Los más utilizados son drones de ala rotatoria por la forma que consiguen mantenerse en el aire, generando la sustentación a través de las fuerzas generadas por las hélices de sus rotores. Tienen mucha versatilidad y son capaces de despegar y aterrizar de forma vertical, la única desventaja es que tienen un tiempo de vuelo reducido.

Dentro del sector agrícola, el dron puede realizar un gran número de tareas[?]:

1. **Supervisión de áreas fumigadas.** La gran distancia de visión del dron permite el análisis de seguimiento de las fincas.
2. **Indicadores de calidad en cultivos.** Las imágenes multispectrales recibidas desde un UAV combinados con parámetros medidos en campo, permiten obtener distintos indicadores de calidad o producción de cultivos.
3. **Generación de inventarios de cultivo.** La visión aérea es una herramienta excepcional para la generación de inventario de cultivos. Es una buena herramienta para lugares de difícil acceso o zonas con mucha cobertura nubosa.
4. **Monitoreo de cultivos.** Gracias a la facilidad del dron puede recibir imágenes de campos de cultivos extensos combinado con cámaras de visión, para mostrar el desarrollo y analizar las eficiencias del cultivo. Los drones pueden proporcionar un mapeo de campo preciso que incluye información de elevación que permite a los productores encontrar cualquier irregularidad en el campo. Tener información sobre la elevación del campo es bastante útil para determinar patrones de drenaje.
5. **Riego:** Dado que el cambio climático afecta cada vez más a las condiciones de sequía, es fundamental crear soluciones de riego eficientes. Los drones pueden capturar información precisa sobre la salud del cultivo, incluidos los niveles de humedad. Esto implica a que el agua se puede distribuir de manera eficiente.
6. **Pulverización.** Los rociadores de drones pueden dirigirse a áreas específicas para maximizar la eficiencia y ahorrar en costos químicos, como por ejemplo el dron que se muestra en la figura 2.2.
7. **Plantación y siembra:** Las sembradoras de drones automatizadas se utilizan principalmente en las industrias forestales. Plantar con dron significa que se pueden replantar áreas de muy difícil acceso sin poner en peligro al ser humano. también son capaces de plantar de manera mucho más eficiente.
8. **Salud de las plantas.** Los drones equipados con un equipo de imágenes especial llamado índice de vegetación de diferencia normalizada que utiliza información de color detallada para indicar la salud de la planta.
9. **Polinización con drones.** Drones polinizadores autónomos que funcionan y monitorean la salud de los cultivos sin instrucciones constantes de los operadores.

Por último la seguridad es una industria de rápido crecimiento, y por eso nos hemos centrado el tema principal del proyecto. Es bastante útil para la gestión agrícola,

son capaces de monitorear los confines de una granja. Las cámaras de drones pueden proporcionar una descripción general de la zona, y mediante una red neuronal incorporada es capaz de procesar las imágenes y detectar cualquier incidencia.

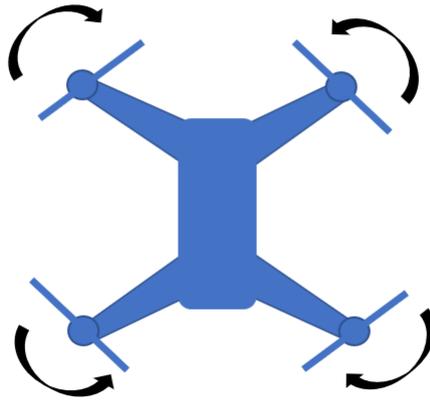


**Figura 2.2:** Drone pulverizador

## **2.4. Control de vuelo**

Un punto importante para el control de vuelo son los elementos de un control PID que tiene como función mejorar la estabilización de vuelo y control de maniobras, así como las distintas variables que puede afectar a la efectividad y desempeños de drone tipo cuadcoptero.

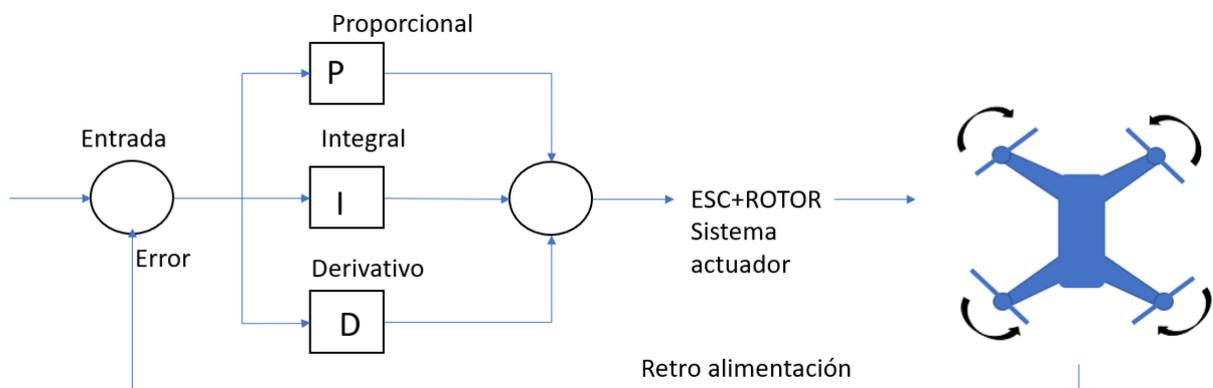
Los cuadcoptero son sistemas de vuelo que llevan cuatro hélices ubicadas alrededor de una estructura principal, las fuerzas aplicadas requieren de mecanismos para estabilizar y controlar la correlación en un constante equilibrio. Cada hélice tiene un sentido, dos giran en sentido horario mientras las otras dos giran en sentido anti horario (véase en la figura 2.3). El movimiento de las cuatro hélices es controlado por sistemas electrónicos.



**Figura 2.3:** Sentido de las hélices

La implementación del control PID se puede dividir en 3 secciones, tal y como se muestra en la figura 2.4 [7]:

1. Control PID Roll, control de giro horizontal que evita que el dron gire
2. Control PID Yaw, control de giro vertical que evita que las fuerzas de movimiento hagan que el sistema gire sobre su propio eje.
3. Control PID pitch, control de elevación que mantiene el avance o retroceso del sistema.



**Figura 2.4:** PID dron

La figura muestra la entrada del sensor de aceleración y giroscopio del UAV que puede estabilizar 3 ejes enviando correcciones al actuador. Gracias a la tecnología actual, estas señales pueden ser analizadas por un microcontrolador, que analiza la entrada mediante algoritmos, procesa los datos y evalúa su corrección en un tiempo.

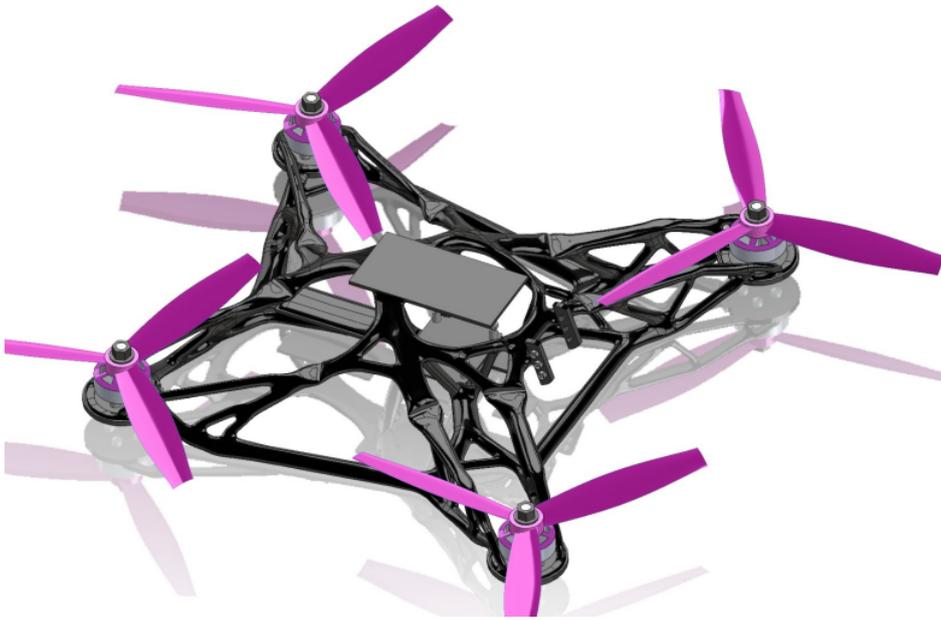
## **2.5. Componentes**

En este apartado encontraremos las partes principales que componen al dron. La base del dron es la placa controladora de vuelo, para el dron es lo mismo que un cerebro para los seres humanos, capaz de enviar señales a todo el cuerpo, a todos el sistema compuesto, encargado de procesar todo el conjunto de información de los sentidos[?].

### **2.5.1. Chasis**

Este es el cuerpo del dron, la base que sujeta todo el conjunto de piezas[8]. Esta base puede fabricarse de distintas formas, variando sus características debido a la forma del chasis (cuadricoptero, hexacoptero...) y las dimensiones que puede llegar a alcanzar. El tipo de chasis se debe elegir bien porque es distinto según el tipo de dron que se necesite. Por ejemplo en los drones de carreras, están contruidos de la manera más ligera y aerodinámica posible. La durabilidad es un desafío en este tipo, porque equivale a generar un peso adicional. El chasis suele componerse por el cuerpo principal, que es el que sujeta el conjunto de componentes, y los brazos, que son los que llevan incorporados los motores y las hélices (como por ejemplo la figura 2.5) . Existen diferentes tipos de chasis:

- Chasis fabricados con fibra de carbono, son materiales muy resistentes, poco pesados y costosos.
- Chasis de fibra de vidrio, no tienen la misma resistencia que el chasis de fibra de carbono. Son más blandos pero obviamente son más económicos.
- Chasis de plástico, es el material más económico. La principal ventaja es que los propios fabricantes pueden generar su propio chasis con impresión en tres dimensiones.



**Figura 2.5:** Ejemplo de chasis UAV diseñado con Generative design

### 2.5.2. Tren de aterrizaje

Es una parte esencial del dron, para poder evitar daños al dron. Los hay de distintos tipos, como tren de aterrizaje fijo o tren de aterrizaje retráctil. El problema es que puede interferir en una cámara con un campo de visión de 360°. El tren de aterrizaje suele ser resistente porque debe evitar los daños con aterrizajes bruscos y ásperos.

### 2.5.3. Hélices

Las hélices son perfiles aerodinámicos giratorios, planas con algo de inclinación de manera que el aire es capaz de aprovechar las fuerzas originadas por el cambio de velocidad o presión. Aprovechan la fuerza que les transmiten los motores para generar una fuerza contraria a la gravedad y superior, denominada como fuerza de sustentación. Tienen dos tipos de posiciones que sirven para contrarrestar el par motor. Básicamente son el principal elemento que permite mantenerse en el aire.

Las hélices pueden ser de dos tipos[9], como se puede ver en la figura 2.6. Los materiales de estas pueden ser tanto de plástico como de fibra de carbono. La selección del material dependerá de los requerimientos que se le estipulen al dron.



**Figura 2.6:** Diferentes tipos de hélices

### 2.5.4. Motores

El motor es el encargado de transformar la energía eléctrica que recibe por parte de la batería en energía cinética[10]. Esta energía pasa a transmitirse a las hélices de los drones y causará un empuje que permitirá elevar el vuelo. Casi todos los cuadcopters utilizan motores sin escobillas (véase la figura 2.7), porque son más eficientes, más fiables y más silenciosos. Dependiendo del diseño del motor puede afectar al tiempo de vuelo.

Los cuadcopters llevan 4 motores instalados y deben estar bien colocados para lograr un cuadricóptero equilibrado. Por ello se necesita la siguiente configuración.

- Frontal izquierdo, motor en sentido horario.
- Frontal derecho, motor en sentido anti horario.
- Trasero izquierdo, motor en sentido anti horario.
- Trasero derecho, motor en sentido horario.



**Figura 2.7:** Motor BLDC

Un motor BLDC destaca por su alta eficiencia, larga vida útil buena relación respecto al peso y un rango alto de velocidad. El motor “*Brushless*” o BLDC contiene una cantidad de electroimanes conectados entre sí en pares específicos. Van conectados al controlador del motor, que es que gestiona el motor activando y desactivando secciones de electroimanes para ocasiones específicas con la finalidad de hacer que el rotor gire debido a la fuerza magnética. Están formados por 2 grandes partes, el rotor que es la parte que gira y lleva los imanes, y el estátor, que al contrario del rotor no gira y lleva los electroimanes.

### 2.5.5. Controlador de vuelo

El controlador de vuelo es el cerebro del dron. Es una placa de circuitos integrados incluyendo sensores que detectan los cambios de orientación. Incluyen sensores básicos de giroscopios y acelerómetros. Tiene la función de dirigir las RPM de cada motor en respuesta a la entrada[?].

Dispone de una serie de entradas y salidas, y sensores incorporados que determinan en tiempo real la posición de la aeronave. Estos van desde simples giroscopios para orientación hasta barómetros para mantener automáticamente las altitudes. El GPS también se puede utilizar para el piloto automático o de seguridad.

Con una configuración adecuada del controlador vuelo, las entradas de control de un piloto deben corresponder exactamente al comportamiento del dron. Los controladores de vuelo son configurables y programables, lo que permite realizar ajustes basados en distintas configuraciones de múltiples rotores. El PID se utiliza para sintonizar el controlador, lo que produce una respuesta rápida y bloqueada. Las características de los controladores de vuelo son:

- Retención de altitud, es una característica avanzada que permite moverse y mantenerse a cierta distancia del nivel del suelo.
- GPS, permite al dispositivo mantener su posición y recoger un registro de datos de vuelo.
- Estabilización giroscópico, mantiene estable el dron durante el tiempo de vuelo.
- Modo de orientación, sirve para poder controlar la orientación del dron.
- Retención de posición, permite permanecer en una posición fija durante tiempos de vuelo.
- Autonivelación, permite mantener nivelado el dron en el aire.

Se pueden dividir en dos categorías principales basado en si el hardware es de código abierto o cerrado. Los más populares de código abierto son APM (Ardupilot) como se muestra en la figura 2.8, Open Pilot CC3D o MultiWii. Y de código cerrado los más populares son Wookong y Naza[11].



Figura 2.8: Ejemplo Ardupilot

### 2.5.6. Variadores (ESC)

Un variador o controlador electrónico de velocidad (Electronic Speed Controller -ESC) ,componente mostrado en la figura 2.9, cuya finalidad es la de variar la velocidad de un motor eléctrico, su dirección y a veces también actuar como freno dinámico. Convierte la energía de una batería de corriente continua en corriente trifásica para alimentar un motor eléctrico, que no es más que un componente electrónico compacto en un dron[12].

No importa qué tipo de motor eléctrico esté conectado al inversor, el ESC interpreta la información de control. Antiguamente los accionamientos eran mecánicos e impulsados por máquinas. El ESC cambia la tasa de cambio del conjunto transistor de efecto variable. Hacen posible un control más suave y preciso de la velocidad de giro de los motores y se consigue una mejora en la eficiencia respecto a los mencionados variadores mecánicos.



Figura 2.9: Ejemplos de ESC

### 2.5.7. Receptor

El receptor es la pieza del dron que permite recibir las señales de radio de la emisora, joystick o radiocontrol.

### 2.5.8. Telemetría

La telemetría (Figura 2.10) es un dispositivo del dron que facilita la información del nivel de batería, la corriente de los motores, la cobertura, velocidad, altitud y coordenadas GPS. Permite monitorear el estado del dron en tiempo real.

Uno de los aspectos más importantes a tener en cuenta es saber que ancho de banda se está utilizando.



Figura 2.10: Ejemplo de telemetría

### 2.5.9. Emisora

La emisora es un control remoto que permite poder controlar el dron. Realiza las funciones de transmisión por radiofrecuencia de las ordenes generadas por el piloto, mientras que un receptor que permiten controlar al dron.

El sistema de 2,4 GHz casi no recibe interferencia. Dicha frecuencia funciona totalmente a las antiguas utilizadas con una frecuencia fija. Actualmente emiten en una banda más ancha llamada distribución dinámica de espectro también utilizada por el estándar Bluetooth o Wi-Fi.

La comunicación entre el receptor y la emisora consiste en un enlace ascendente que transmiten ordenes a los actuadores del dron y un enlace de bajada que devuelve los datos a la telemetría.

### 2.5.10. Batería

La batería es otro componente esencial del dron. Se encuentra en la categoría LiPo (Polímero de litio) porque ofrece la mejor combinación del mercado en términos de densidad de energía, capacidad y vida útil. Un dron equipado con varias baterías no aumenta el tiempo de vuelo en proporción a la cantidad de baterías añadidas.

Factores como la potencia del motor, el ventilador y el peso de la batería hacen que varias baterías sean ineficientes.

Estas baterías de polímero de litio deben ser con la mayor tensión posible, para reducir la corriente necesaria y obtener una gran eficiencia del dron.

## **2.6. Modos de vuelo**

Son muchos los modelos de dron que incluyen modos de vuelo inteligentes, controlados mediante los controles de la emisora como se explica en la Figura 2.11. Actualmente encontramos 14 modelos de vuelo, tales como:

- Estabilizador, permite volar el UAV manualmente autonivelando los niveles de roll y pitch.
- Mantenimiento de altitud, el dron mantiene una altitud constante pudiendo maniobrar el roll, pitch y yaw.
- Merodear, consiste en mantener la misma posición, grados y altitud.
- RTL, el UAV navega desde la posición actual para volar alrededor de su zona lanzamiento. Permite volver y aterrizar automáticamente,
- Automático, seguirá un plan de vuelo programado, guardados en la controladora de vuelo mediante comandos.
- Acro, usa los joysticks de la emisora para controlar la velocidad angular del dron. Este modo es útil para acrobacias.
- Sport, puedes mantener el vehículo a un determinado ángulo y bloquear la altitud.
- Drift, permite volar el dron como si fue un avión, con tiempos automáticos coordinados.
- Modo guiado, el dron es guiado a una cierta ubicación mediante un módulo radio de telemetría.
- Modo posición, es parecido al modo merodear pero con el control manual del throttle.
- Modo seguimiento, es posible que el dron pueda seguir tus movimientos mediante una radio de telemetría.

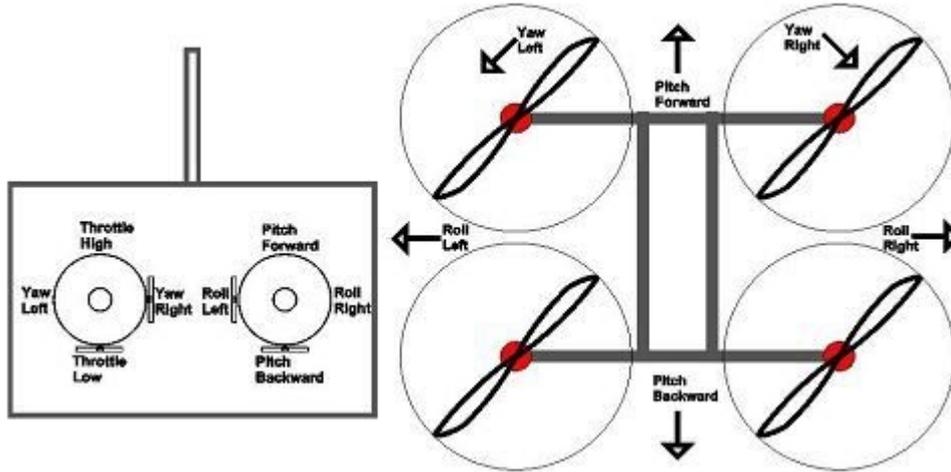


Figura 2.11: Control de drone

## 2.7. Normativa drone

Actualmente ya no se hace ninguna distinción entre vuelos profesionales y vuelos recreativos, por lo tanto cualquier persona debe atender a las mismas normativas. Independientemente de las categorías (mencionadas en capítulos), hay una serie de normas en España que se debe de tener en cuenta:

- El dron, siempre debe estar al alcance visual del piloto.
- Nunca debe sobrepasar los 120m de altura en pleno vuelo.
- No volar en un mínimo de 8km de cualquier zona de aeropuerto, aeródromo o espacio aéreo controlado.
- No es obligatorio el seguro de responsabilidad civil, aunque es muy recomendable tener uno. El piloto será siempre responsable de los daños causados por su vehículo.
- El dron debe llevar una placa identificativa ignífuga con datos del dron y piloto. Indicando el fabricante, tipo de dron, modelo, número de serie, nombre del piloto y datos de contacto.
- Se debe proteger siempre el derecho a la intimidad para no vulnerar la Ley de Protección de Datos.

Además se debe volar con buenas condiciones meteorológicas para garantizar la seguridad del espacio aéreo y de las personas. No es necesario registrar el dron a menos que esté certificado, pero si es obligatorio realizar un registro en AESA. Dicho registro será válido durante 5 años y se reconocerá en todos los países miembros de EASA, según la directiva 2009/48/CE.

## *Estado del arte*

Dependiendo de los riesgos de operaciones de vuelo, existen tres categorías operacionales englobadas a la normativa:

1. Categoría abierta, vuelos de bajo riesgo sin autorizaciones.
2. Categoría específica, vuelos riesgo medio con autorizaciones en ciertos escenarios.
3. Categoría certificada, operaciones de riesgo alto.

LIMITACIÓN SUBCATEGORÍA	REQUISITOS DE AERONAVES	REQUISITOS DE PILOTOS
<p>A1</p> <p>Se permite el sobrevuelo de personas ajenas a la operación</p>	Construcción privada o previa a la norma de <250 g y <19 m/s	Familiarizarse con el manual de usuario del fabricante
	Clase 0 (<250 g)	Familiarizarse con el manual de usuario del fabricante
	Clase C1 (<900 g y <80J con e-ID y Geo-awareness)	Familiarizarse con el manual de usuario del fabricante. Complementar un curso online y superar un examen teórico online
<p>A2</p> <p>Se permite el vuelo cerca de personas ajenas a la operación Manteniendo, una distancia de seguridad (30-5 metros)</p>	Clase C2	Familiarizarse con el manual de usuario del fabricante
	(<4 kg con low-speed, e-ID y Geo-awareness)	Poseer un certificado de competencia de piloto remoto, obtenido mediante formación y examen online, autopráctica y examen presencial
<p>A3</p> <p>Operaciones en áreas donde no se espera poner en peligro a personas ajenas a la operación</p> <p>Manteniéndose a &lt;150 metros de áreas residenciales, comerciales, industriales o recreacionales</p>	Construcción privada o previa a la norma de <25 kg	Lo mismo que la clase C1 en A1
	Clase C2 (<4 kg con e-ID y Geo-awareness)	
	Clase C3 (<25 kg con e-ID y Geo-awareness)	
	Clase C4 (<25 kg)	

Cuadro 2.3: Normativa de subcategorias

## 2.8. Inteligencia artificial

Hoy en día los términos de Inteligencia Artificial, Machine Learning y Deep Learning (Figura 2.12) son muy utilizados. Aunque suelen emplearse los términos en inglés, en español sería Aprendizaje Automático y Aprendizaje Profundo. La IA es la inteligencia llevada a cabo por máquinas, y se define como el conjunto de técnicas que permiten a las computadoras imitar a los humanos.



**Figura 2.12:** Relación de pertenencia entre la inteligencia artificial, el aprendizaje automático y el aprendizaje profundo

### 2.8.1. Machine Learning

Un subconjunto de la IA es el aprendizaje automático, que tiene la capacidad de recibir conjuntos de datos y aprender por sí mismas, cambiando los algoritmos a medida que aprenden más sobre la información que están procesando.

### 2.8.2. Deep Learning

El otro subconjunto de la IA es el aprendizaje profundo, dentro de este se encuentran las redes neuronales artificiales, objeto de este proyecto. Estas redes imitan la conectividad del cerebro humano, clasifican conjuntos de datos y encuentran correlaciones entre ellos. La máquina adquiere nuevos conocimientos sin la intervención humana y aplica sus conocimientos a otros conjuntos de datos, cuantos más datos tenga la máquina a su disposición, más precisas serán sus predicciones.

### 2.8.3. Segmentación por instancias de imágenes

La segmentación de imágenes es un ámbito que se está desarrollando de una manera exponencial a causa de que es una aplicación con mucha viabilidad en el ámbito industrial, ya que la visión por computadora está al nivel de intentar automatizar tareas que la visión humana puede realizar.[13]

Hay varios tipos de clasificación de imágenes, empezando por una muy básica y llegando a una clasificación exhausta.

#### Clasificación de Imágenes

- **Clasificación de objetos.** Donde se identifica el contenido de la imagen, tal y como se muestra en la Figura 2.13: persona, árbol, césped y cielo.



Figura 2.13: Clasificación de objetos

- **Detección de objetos.** Donde se identifica el contenido y su posición en la imagen, tal y como se muestra en la Figura 2.14 las personas.



Figura 2.14: Detección de objetos

- **Segmentación semántica.** Donde cada píxel está etiquetado en la imagen en categorías. Como se muestra en la siguiente figura, la imagen se divide en

persona (rojo), árbol (verde oscuro), hierba (verde claro), cielo (azul). (Figura 2.15).

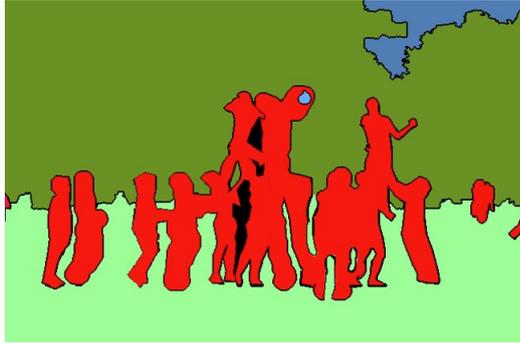


Figura 2.15: Segmentación semántica

- **Segmentación de instancias.** Es la anotación combinada de detección de objetos y segmentación semántica. La detección del objeto es lo primero y luego cada píxel se etiqueta (segmentación semántica). En comparación con la Figura 2.15, la segmentación de instancias distingue diferentes instancias de la misma categoría (diferentes personas se distinguen por diferentes colores) como se puede observar en la Figura 2.16



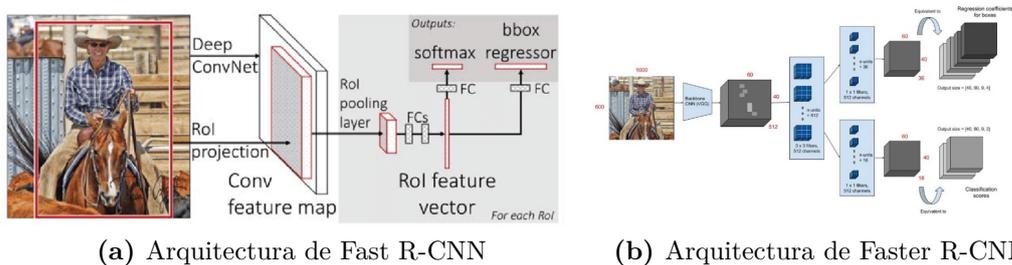
Figura 2.16: Segmentación de instancias

#### 2.8.4. Fast R-CNN, Faster R-CNN y Mask R-CNN

Un problema importante con RCNN es que la búsqueda selectiva de 2000 RoIs requiere un costo computacional extremadamente alto, incluido el cálculo de la CNN además del clasificador SVM y la regresión para cada RoIs. Una forma de mejorar el rendimiento computacional de R-CNN es utilizar una única red convolucional para todas las RoIs y generar un único extractor de características para cada imagen. Así es como funciona Fast R-CNN (Figura 2.17a). El resultado es que se gana tiempo de

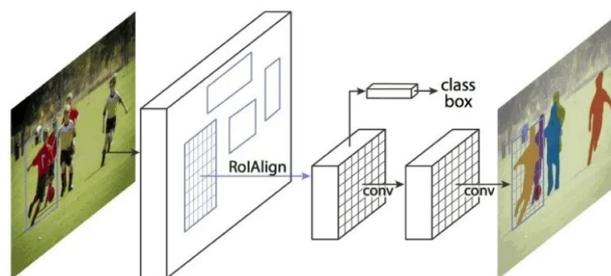
procesamiento al eliminar la redundancia de procesamiento. Por otro lado, Faster R-CNN (Figura 2.17b) comparte la CNN con el algoritmo de detección de RoIs, que reemplaza la búsqueda selectiva con una red neuronal RPN (Region Proposal Network). Esta modificación hace que Faster RCNN sea algo más rápida que Fast R-CNN, de ahí el adjetivo "Faster"[14].

En ambos casos, solo una CNN aplica la convolución a la imagen, mientras que el RoI se obtiene en paralelo y separado de ella. Cabe señalar que, a diferencia de la CNN tradicional, Fast R-CNN utiliza la estructura ConvNet o ResNet-101 sin red neuronal artificial en la última capa. Esto se debe a que, en ambos, se intenta correlacionar el interior del RoI con el mapa de características, haciendo coincidir su tamaño a través de una operación de agrupación máxima. Esta operación también requiere la deformación de la imagen dentro del RoI.



**Figura 2.17:** Comparativa Arquitectura Fast R-CNN vs Faster R-CNN

*Mask R-CNN* es igual que Faster R-CNN pero incorpora una rama de salida adicional con la imagen marcando los píxeles del objeto dentro de la RoI. Esta tercera ramificación, coge las características deformadas de la *RoI pooling* y las pasa por dos CNNs más (Fig. 2.18): una que es la que genera las máscaras binarias de cada RoI, y otra clasificadora que permite generar máscaras para cada clase sin que las clases compitan entre ellas. Esto es necesario, para poder marcar varios objetos de diferentes clases en una misma imagen[15].



**Figura 2.18:** Arquitectura Mask R-CNN

En resumen, en la siguiente Figura 2.19 se pueden ver las diferencias entre las 3 estructuras.

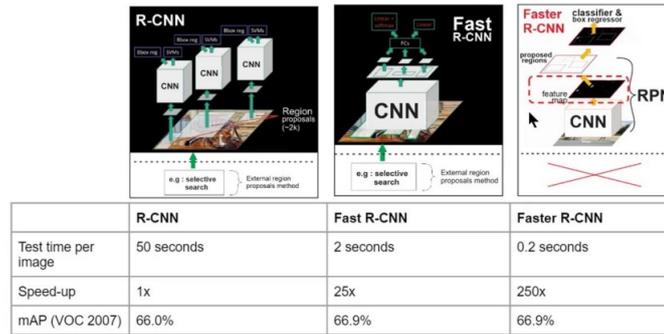


Figura 2.19: Arquitectura Mask R-CNN

## 2.9. Historia de la IA hasta las redes neuronales artificiales

A pesar de que los primeros referentes históricos se hallan aproximadamente en el año 1936 con Alan Turing, que es considerado padre de la inteligencia artificial, se considera que el punto de partida es a partir del año 1950. En el año 1950, Alan Turing publica un artículo llamado “**Computing machinery and intelligence**” en la revista *Mind*. En este artículo, Turing lanzaba la pregunta *¿Pueden las máquinas pensar?* y proponía un método para determinar si estas podían hacerlo. Estos métodos pasaron a denominarse **Test de Turing**, y mediante cuya superación por una máquina se podía considerar que sería capaz de un humano en una charla ciega. Este test sigue vigente en la actualidad.

Sin embargo, numerosos investigadores e historiadores consideran que el punto de partida fue el año 1956[16], donde **John McCarty**, **Marvin Misky** y **Claude Shannon** declararon en la conferencia de Darmouth el término como “*La ciencia e ingenio de hacer máquinas inteligentes, especialmente programas de cálculo inteligente*”.

En 1958, se creó la primera neurona, llamada Perceptron. Fue creada por F. Rosenblatt, y esta era un clasificador binario o discriminador lineal, el cual genera una predicción basándose en un algoritmo combinado con el peso de las entradas. Años más tarde, se demostró que no era capaz de resolver problemas relativamente fáciles y esto pausó el avance de las redes neuronales durante algunos años.

En 1989 se crea la primera red neuronal convolucional capaz de extraer características y luego clasificar. Aunque no fue hasta 2012 cuando este tipo de redes se empezaron a usar para la clasificación y localización visual.

El gran desarrollo en el ámbito del Deep Learning y las redes neuronales se produce a partir de 2006 gracias al aprovechamiento del poder de las GPU.

El interés hoy en día es diseñar e implementar sistemas que permitan analizar grandes cantidades de datos de forma rápida y eficiente.

## 2.10. Antecedentes de la visión artificial

Situándonos en la situación económica actual, es importante tener factores diferenciales frente a la competencia. Uno de estos factores puede ser aplicar la visión artificial en la industria o cualquier otro ámbito. Respecto la industria, ya hace varios años que se llevan aplicando técnicas de IA para el control de calidad de varios procesos, mejorando así la calidad, y revisando casi el 100 % del producto fabricado. También podemos ver que se han aplicado técnicas en el ámbito agrícola, controlando la humedad de los terrenos mediante cámaras de IA para saber cuando hay que regar exactamente el cultivo.



# 3 Montaje y configuración del dron

En este capítulo se encuentra la descripción de todos los componentes que se han usado para el montaje del dron, el proceso de ensamblaje y la puesta a punto del dron para que pueda volar. También se explica que aplicación se utiliza para la transmisión de imágenes realizadas por el dron mediante el protocolo HTTP y una comunicación entre el dron y un servidor.

## 3.1. Componentes

A continuación se introducen todos los componentes usados, la gran mayoría de ellos se han comprado conjuntamente en un pack de ensamblaje, para así abaratar costes.

### 3.1.1. Frame

Es la estructura en la que se montan todos los componentes *on board* del dron. Esta estructura de la marca JMT se conforma por 4 ejes cuya distancia entre ellos es de 560mm y una altura de 190mm.

El frame esta fabricado con fibra de carbono 3K de alta calidad. A pesar de sus dimensiones(véase la Figura 3.1), el diseño de este dron esta pensado para que sea fácil su transporte, por lo que el marco donde van sujetos los motores es plegable, reduciendo así el espacio que ocupa. Respecto a la distribución de pesos, este kit viene totalmente preparado para realizar una correcta distribución de pesos y así el centro de gravedad se mantenga lo mas centrado posible en el dron. en el Cuadro 3.1 podemos observar varias características del frame, que está formado por 4 ejes de fibra de carbono de 16mm que forman el soporte de los motores, dos placas centrales, para la instalación de los diversos componentes, y el tren de aterrizaje.

<b>Material</b>	Fibra de carbono 3K
<b>Distancia entre ejes</b>	560 mm
<b>Peso(Aprox.)</b>	650 g
<b>Carga máxima(Aprox.)</b>	2 Kg

**Cuadro 3.1:** Características del frame

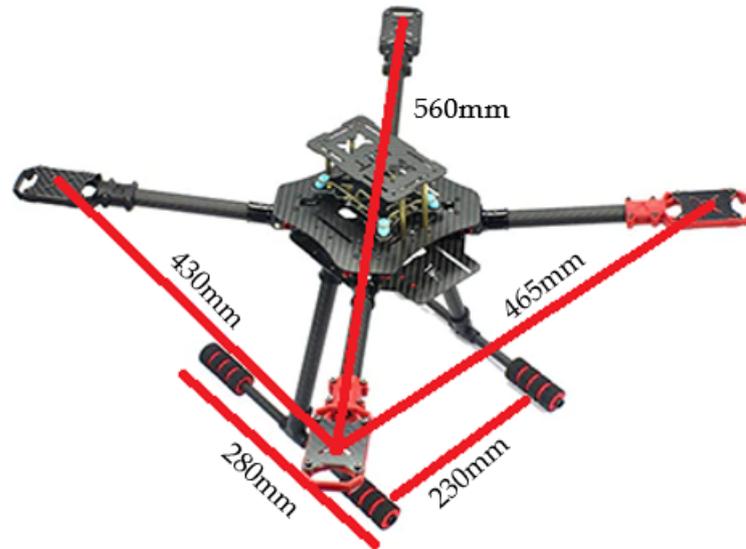


Figura 3.1: Dimensiones del frame

### 3.1.2. Motor

Los motores son *brushless* tal y como se ha comentado en la Sección 2.5.4 son los más usados para quadcopteros. Estos son de la marca JMT, y están diseñados tanto como para quadcopteros como hexacopteros.

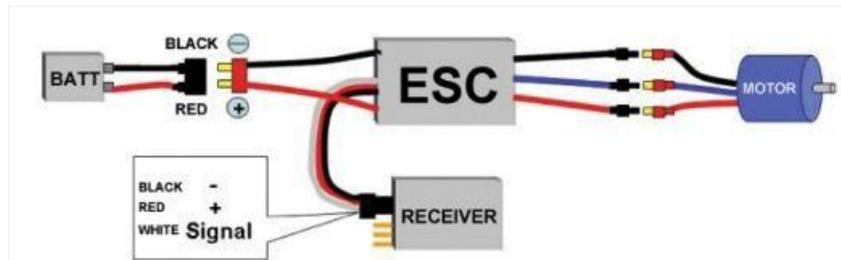
En la Tabla 3.2 podemos encontrar diferentes parámetros del motor, entre ellos uno de los más importantes es el KV, este parámetro define el número de rpm que el motor es capaz de ofrecer cuando se le aplica 1 Volt de tensión. A pesar de ir relacionado con las rpm, nos indica velocidad pero no mayor torque, ya que hemos de buscar un equilibrio para que no sufran sobrecalentamientos.

<b>Modelo</b>	JMT HYD 3508 (F15276)
<b>KV</b>	700 KV
<b>Intensidad sin carga</b>	0.4 A
<b>Intensidad máxima</b>	18.0 A
<b>Potencia</b>	198 W
<b>Peso(Aprox.)</b>	90 g
<b>Voltaje de entrada</b>	2-4S LiPo
<b>Hélices compatibles</b>	12 a 15 pulgadas

Cuadro 3.2: Especificaciones del motor

### 3.1.3. ESC

Los variadores 3.3 o controladores electrónicos de velocidad también son de la marca JMT. Tienen una corriente máxima de 40 A, y estos tienen 3 cables por un lado (positivo, negativo y tierra) que van conectados a la controladora, y 3 puertos de conexión para conectar los motores. En la figura 3.2 se puede observar como sería la conexión de los ESC[17].



**Figura 3.2:** Conexión de las ESC

Está diseñado con la finalidad de hacer girar el eje del motor a la velocidad establecida por la controladora cuando genera la señal PWM. En este caso, el ESC lleva integrado un BEC (Battery Eliminator Circuit) que permite alimentar el ESC con la misma batería que se alimenta al motor consiguiendo un ligero decremento de peso en el conjunto, ya que nos ahorramos un componente.

El parámetro a tener en cuenta, es la corriente máxima, ya que si el motor requiere mas corriente que la que necesita el ESC, se sobrecalentara y producirá fallos, llegando incluso a quemarlo. Los tres factores que tienden a incrementar el consumo de corriente son:

- Motores más grandes (diámetro y/o altura del estátor)
- Motores con KV más elevados
- Hélices más grandes (longitud y/o paso)

<b>Intensidad máxima</b>	40 A
<b>Entrada de alimentación</b>	2-4S LiPo
<b>Salida BEC</b>	5V/3A
<b>Peso</b>	32 g (con conector banana)
<b>Tamaño(Aprox.)</b>	68 x 25 x 8mm

**Cuadro 3.3:** Especificaciones de los variadores



Figura 3.3: ESC

### 3.1.4. Hélices

Las hélices son de fibra de carbono 3K, de diámetro 14 y 5.5 pulgadas de paso, siendo con esta información hélices de numeración 1455.

Como se menciona en la Sección 2.5.4 cada motor tiene un sentido de giro, funcionando dos a dos, por lo que tenemos dos tipos de hélices con perfiles opuestos, uno para el sentido horario y otro para el antihorario. La hélice que gira en sentido horario se llama *pusher*, mientras que la que lo hace en el otro sentido se llama *tractor*. La finalidad de las hélices es sustentar el dron de una forma equilibrada y plana.

### 3.1.5. Batería

Se ha usado una batería LiPo (Figura 3.4) gracias a que permiten realizar descargas muy rápidas, permitiendo a nuestro dron usar cantidades de energía elevadas en poco tiempo con una batería ligera y de unas dimensiones reducidas[18]. Se trata de una batería de 11.1V, con una configuración 3S1P, que esto quiere decir que esta conformada por 3 celdas en serie, con una carga nominal por celda de 3.7V. Respecto a la capacidad de la batería se trata de una batería de 6000mAh y una velocidad de descarga continua de 25C. Este parámetro nos indica la rapidez con la que podemos descargar la batería de una forma segura, en nuestro caso, aplicando una intensidad de carga de 6.0 A tardaría una hora en descargarse por completo, y teniendo en cuenta nuestra velocidad de descarga y aplicando 150A de corriente de forma continua, podríamos descargar por completo la batería en 2 minutos 24 segundos. En el cuadro 3.4 podemos observar las especificaciones de la batería usada.

Los fabricantes recomiendan que la batería nunca se descargue por debajo de los 3.2V por celda, por lo que en nuestro caso es importante no descargarla por debajo de los 9.6V en ninguno de los casos.

Respecto la carga, se tiene que hacer mediante un cargador específico que nos permita indicar la velocidad de carga, y se produzca un balanceo de carga en las celdas

para mantener en buen estado siempre la batería y no reducir el ciclo de vida de la batería, ya que si la cargamos o descargamos por encima de sus valores, produciríamos una reducción considerable de sus ciclos de vida.

Marca	TCB WORTH
Tipo y capacidad de la batería	LiPo 6000 mAh
Velocidad de descarga continua	25 C
Voltaje por celda y configuración	3.7V 3S1P (11.1V)
Voltaje máximo por celda	4.2V (12.6V)
Tasa de Carga	1C
Certificados	CE, RoHS, MSDS, UN38.3
Peso	419 g
Tamaño	158 x 48 x 27mm
Conector de la batería	XT60

Cuadro 3.4: Especificaciones de la batería



Figura 3.4: Materia LiPo

### 3.1.6. Controladora de vuelo

La controladora de vuelo que hemos usado es el Ardupilot APM 2.8 incluido en el kit de ensamblaje junto a un módulo GPS (Véase la Figura 3.5). Se trata de una controladora de piloto automático de código abierto compatible con Arduino. Con piloto automático nos referimos a que es capaz de realizar misiones GPS programadas con waypoints, estabilizarse de manera autónoma y telemetría bidireccional. Incluye un giroscopio de 3 ejes, acelerómetro y un barómetro de alto rendimiento[19]. Dispone de 4 MB de memoria para el registro de datos a bordo, y así poder analizarlos después del vuelo.



Figura 3.5: APM 2.8 y módulo GPS

Respecto a la parte interna de la controladora, podemos ver en la Figura3.6 los siguientes puertos I/O numerados[20].

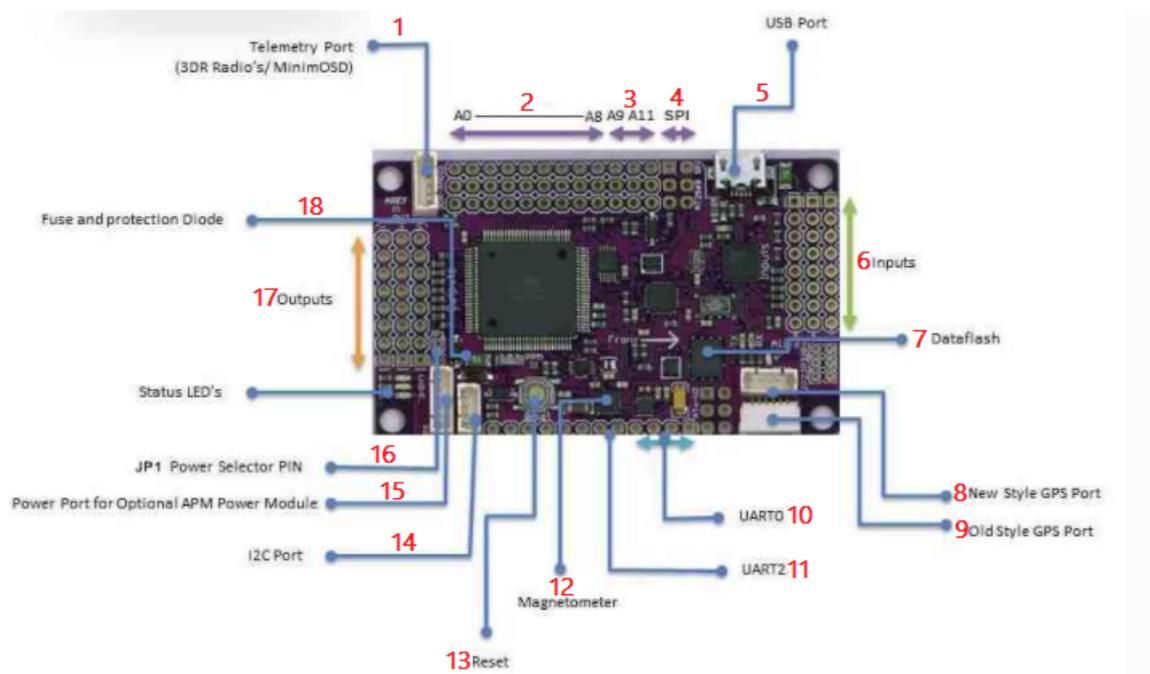


Figura 3.6: Parte interna del APM 2.8

A continuación una breve descripción de los puertos:

1. Telemetría: Puerto para conectar el módulo de telemetría.
2. Pines A0-A8: Pines de entrada analógicos . Funcionan a 5V y se suele usar para conectar sensores.
3. Pines A09-A11: Autostability Gimball output. Estabiliza la posición de grabación y captura de la cámara .
4. Pines A12-A13: Puertos SPI para programación con el microcontrolador AT-MEGA2560. Permite la comunicación con sensores y actuadores.
5. Puerto USB. Puerto destinado para la configuración del APM y controles de vuelo mediante la aplicación *Mission Planer*.
6. Inputs 1-8: Puertos destinados al control remoto.
7. Dataflash: Memoria de 4MB de almacenamiento.
8. Puerto para el sensor GPS.
9. Puerto I2C para brújula externa.
10. UART0: Puertos SPI para programación con el microcontrolador ATMEGA32U2.
11. UART2: Puerto para conectar el módulo de telemetría.
12. Giroscopio de 3 ejes.
13. Pulsador de RESET de la controladora de vuelo.
14. Puerto I2C para brújula externa.
15. Puerto para el módulo de potencia.
16. JP1: Jumper JP1 para seleccionar la fuente de alimentación. Si se activa el jumper, no es necesario módulo de potencia ya que la APM se alimenta a través de los variadores, aunque no es una opción recomendable.
17. Salidas encargadas de el envío de señales a los motores.

#### 3.1.7. Transmisor

El tranmisor es el AT9S (Figura 3.7) de 10 canales 2.4GHz, que opera en el espectro DSSS y FHSS de forma sincronizada. Opera libre de interferencia en dispositivos terrestres con un alcance superior a los 900m y en dispositivos aéreos de mas de 1500m.



Figura 3.7: Transmisor

El modo utilizado en este caso es el SBUS donde su frecuencia de funcionamiento es de 2.4GHz. También tiene 16 pseudocanales aleatorios y modulación QPSK (Modulación por desplazamiento de fase cuadrifásica, véase en la Figura 3.8). Este tipo de desplazamiento depende de la entrada binaria que reciba.

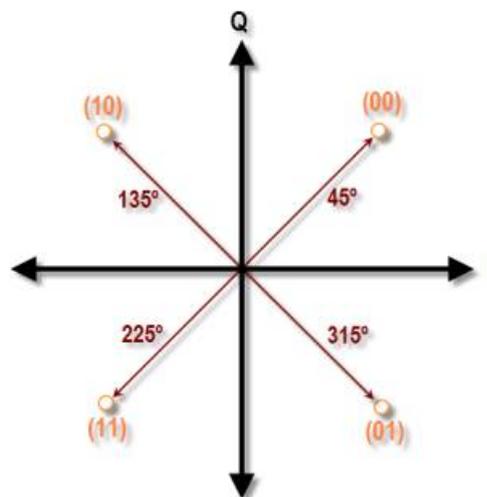


Figura 3.8: Modulación QPSK

En el Cuadro 3.5 encontramos las características principales del transmisor. En este proyecto se han utilizado 4 canales para los movimientos básicos (adelante/atrás, iz-

quierda/derecha, giro horario/antihorario y arriba/abajo) y 2 canales para el control de los cambios de modo de vuelo.

<b>Modelo</b>	AT9S
<b>Peso</b>	0.88 kg
<b>Dimensiones</b>	183 x 100 x 193mm
<b>Tensión de funcionamiento</b>	7.4 ~ 18.0V
<b>Alcance</b>	Tierra >900m; Aire 2 km
<b>Nº de Canales</b>	10 canales
<b>Frecuencia</b>	2.4 GHz

Cuadro 3.5: Especificaciones del transmisor

### 3.1.8. Receptor

El receptor utilizado es el Radiolink R9DS (Figura 3.9) de 2.4 GHz de 9/10 canales que venia incluido con el transmisor AT9S. Las especificaciones se pueden consultar en el Cuadro 3.6. En el desarrollo de este proyecto se han usado 8 canales PWM.

<b>Modelo</b>	R9DS
<b>Peso</b>	10.7 g
<b>Dimensiones</b>	41 x 23 x 14 mm
<b>Tensión de funcionamiento</b>	5V
<b>Nº de Canales</b>	9 canales PWM con LED rojo, 10 canales SBUS con LED azul
<b>Tipo de señal</b>	SBUS / PWM / PPM / CRSF
<b>Frecuencia</b>	2.4 GHz
<b>Alcance</b>	2000m

Cuadro 3.6: Características del receptor

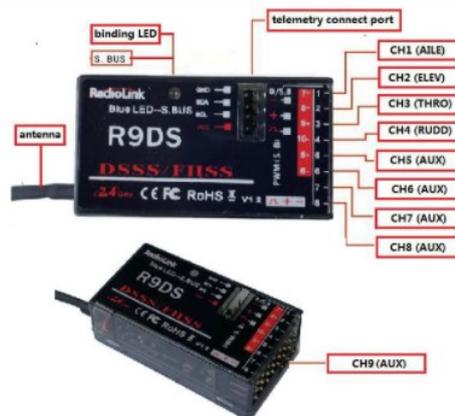


Figura 3.9: Receptor R9DS

### 3.1.9. Módulo de telemetría

El módulo de telemetría (Figura 3.10) se constituye por dos dispositivos radio, uno conectado al dron y el otro a un dispositivo terrestre, en nuestro caso un ordenador portátil. Podemos encontrar las especificaciones en el Cuadro 3.7.



**Figura 3.10:** Módulos de telemetría y su cableado

<b>Peso</b>	63 g
<b>Dimensiones del módulo aéreo</b>	51 x 30 x 10mm
<b>Dimensiones del módulo terrestre</b>	70 x 25 x 12mm
<b>Frecuencia</b>	433 MHz
<b>Sensibilidad del receptor</b>	-121 dBm
<b>Potencia</b>	500 mW
<b>Velocidad de transmisión</b>	fins a 250 kbps
<b>Protocolo de transmisión</b>	MavLink
<b>Técnica de modulación</b>	FHSS
<b>Alcance</b>	1500m
<b>Tipo de comunicación</b>	dúplex completa en 2 direcciones mediante TDM adaptable

**Cuadro 3.7:** Especificaciones del módulo de telemetría

### 3.1.10. GPS

El módulo GPS (Figura 3.11) se encarga de la transmisión de la posición del dron en tiempo real, en nuestro caso, este módulo lleva integrados una brújula, más precisa que la que lleva integrada la controladora de vuelo. Este GPS está integrado con un módulo *nublo LEA-6H*.



Figura 3.11: GPS

### 3.1.11. Módulo de potencia

La función del módulo de potencia usado (Figura 3.12) es alimentar todos los componentes que lleva instalados el dron, proporciona la tensión necesaria a cada uno de los componentes. Este módulo se alimenta directamente con la batería, y a partir de aquí se gestiona la alimentación tal y como se ha mencionado anteriormente. Lleva incorporado alimentación tanto a 5V como a 12V, siendo la primera para las ESC y la segunda para la alimentación del Gimball. Este módulo viene perfectamente adaptado para la instalación en el frame, incluyendo unos pasadores de goma para reducir al máximo las vibraciones.

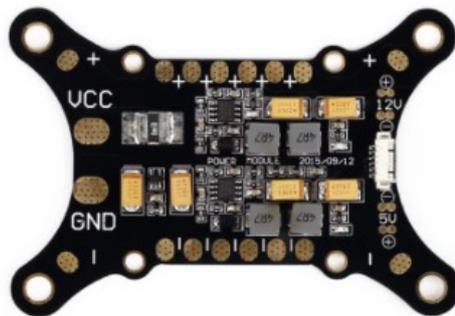


Figura 3.12: Módulo de potencia

### 3.1.12. Gimball

El gimball que se ha usado es uno de 2 ejes controlados por 2 motores BLDC. Tal y como se puede observar en la Figura 3.13, lleva incorporados pasadores de goma como el módulo de potencia para reducir vibraciones.

Va conectado directamente a la controladora de vuelo, y se calibra solo al conectarlo, es un dispositivo *Plug and Play*.



Figura 3.13: Gimball

## 3.2. Firmware de vuelo

En esta sección se explicará que programa se ha utilizado para el control del vuelo y algunas de sus funcionalidades.

### 3.2.1. Estación de control terrestre: *Mission Planner*

Mission planner es un programa solo disponible en entorno Windows. Para su ejecución necesitamos un dispositivo terrestre con entorno Windows, y tener una comunicación con el dron vía telemetría usando protocolo MavLink. Algunas de las funcionalidades son[21]:

- Cargar el firmware en la controladora de vuelo (APM, PixHawk. . .) .
- Establecer, configurar y poner a punto el dron para un rendimiento óptimo.
- Planificar, guardar y cargar planes de vuelo autónomos.
- Descargar y analizar los registros del vuelo creados por la controladora de vuelo.

- Supervisar el estado del dron mientras está en funcionamiento.
- Realizar un vuelo de forma FPV (vista en primera persona).

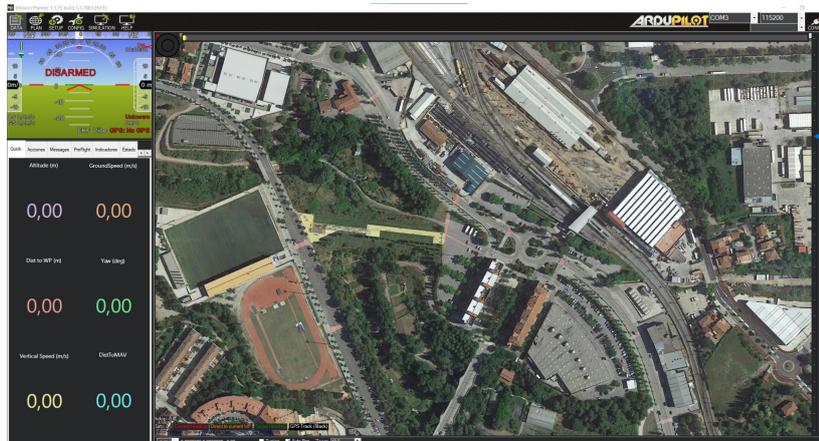


Figura 3.14: Interfaz de usuario principal

La Figura 3.14 muestra la interfaz de usuario principal de Mission Planner.

### 3.3. Ensamblaje

En esta sección se explicará detalladamente el ensamblaje del dron. Lo primero que se ha hecho, es revisar que estaban todos los materiales y una vez revisado, organizarlos según el orden de ensamblaje que se iba a seguir. Es importante aplicar una buena distribución de pesos para mantener el centro de gravedad lo mas centrado posible ya que si no quedará desequilibrado, y como consecuencia unos motores trabajaran más que otros para estabilizar el dron en todo momento del vuelo.

#### 3.3.1. Ensamblaje del frame y tren de aterrizaje

El chasis del dron esta conformado por un tren de aterrizaje unido por una placa donde se instalaran los variadores y los brazos plegables donde se instalaran los motores, y sobre esta, otra placa donde se colocara la controladora de vuelo.

El primer paso ha sido el montaje del tren de aterrizaje, montando primero el apoyo de la base, donde se introduce un pasador con forma de “T” para realizar la unión del apoyo con la placa que unirá los 2 apoyos, estos dos tubos de fibra de carbono se fijaran mediante unos soportes de plástico que bloquean el tubo al soporte mediante un pasador prisionero. Respecto el apoyo del tren de aterrizaje, se han instalado 4 puntos de apoyo, uno en cada extremo, con espuma de goma eva para mejorar su adherencia.

## Montaje y configuración del dron

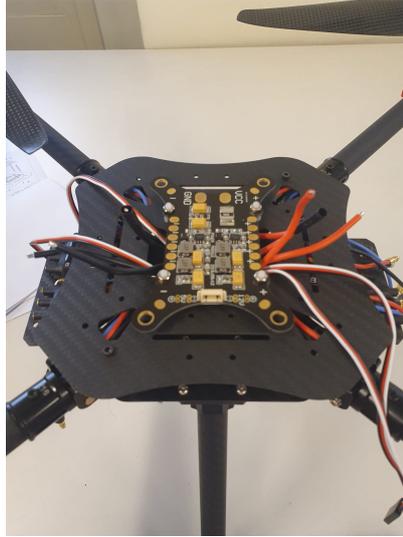
El segundo paso ha sido la fijación de los brazos plegables a la placa, estos van unidos mediante unas sujeciones ancladas a la placa, y que llevan unos pasadores con muelle para poder realizar el plegado y facilitar el transporte del dron. En los extremos lleva unos acoples de plástico donde se fijaran los motores.

En la Figura 3.15 se puede observar el resultado del montaje frame con el tren de aterrizaje.



**Figura 3.15:** Frame con el tren de aterrizaje ensamblado

Respecto a la placa superior, va colocada sobre la otra placa anclada a los brazos plegables, generando así una robustez en el chasis significativa. Sobre esta placa se ha instalado la placa de potencia que va fijada mediante tornillos a la placa superior (Véase la Figura 3.16), donde se ha soldado la alimentación de las mismas a la placa. La placa de potencia, como se ha mencionado anteriormente, también consta de dos placas unidas por unos pasadores de goma para evitar vibraciones, la placa superior únicamente sirve para poder colocar sobre ella la controladora de vuelo. Finalmente se fija el soporte de la batería, que va fijado con tornillos a unos pasadores más elevados, para dejar espacio entre la controladora y la batería.

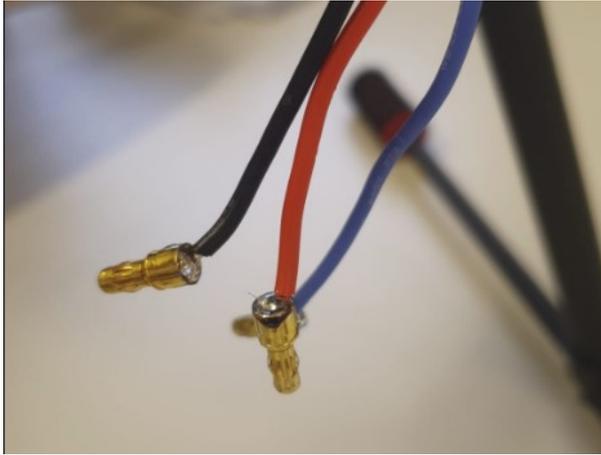


**Figura 3.16:** Placa de potencia

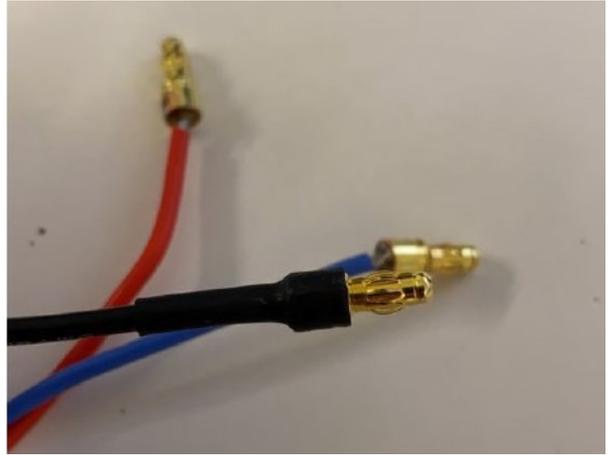
### 3.3.2. Montaje de los motores y variadores

Una vez montado todo el chasis, se procede a realizar el montaje de los motores en los soportes de los brazos plegables. Como se puede observar en la Figura 3.15, hay dos soportes de color rojo, y dos de color negro, esta distinción sirve para indicar cual es la parte posterior y anterior del drone. Los dos brazos con soportes rojos indican la parte posterior, y por consiguiente, los brazos con soporte negro la anterior.

Para la fijación de los motores, lo primero que se ha hecho ha sido pasar todos los cables por el interior del tubo, quedando así ocultos y mejorando estéticamente el resultado, y funcionalmente también, ya que evitamos que estos cables puedan interferir en el giro de las aspas, provocando un accidente y rotura de algunos componentes. Una vez hecho esto, se ha introducido estaño dentro del conector macho de tipo *bullet*, para así facilitar la soldadura, ya que aplicando luego un poco de calor solo tenemos que introducir un poco de cable previamente pelado y ya quedará totalmente unido. Para proteger la unión del conector con el cableado, se ha integrado un cable termoretráctil en la unión (Véase la Figura 3.17). Este proceso se ha realizado para los cuatro motores.



(a) Conectores soldados



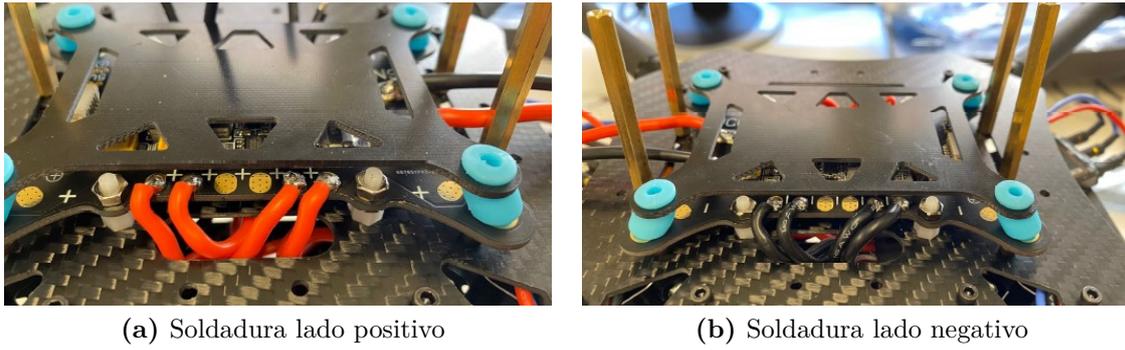
(b) Conectores retractilados

**Figura 3.17:** Soldadura y protección con termoretráctil de los conectores

Para finalizar, se han conectado los motores a los variadores, que a pesar de tener cables de 3 colores, los variadores no tienen una etiqueta para indicar donde va conectado cada cable, lo único que afectará en la conexión de estos cables es el sentido de giro de los motores. Así que a base de prueba y error se han configurado estos motores usando un variador de frecuencia para que giren 2 a 2 como se menciona anteriormente.

### 3.3.3. Soldadura y conexión e instalación de la electrónica a la controladora de vuelo

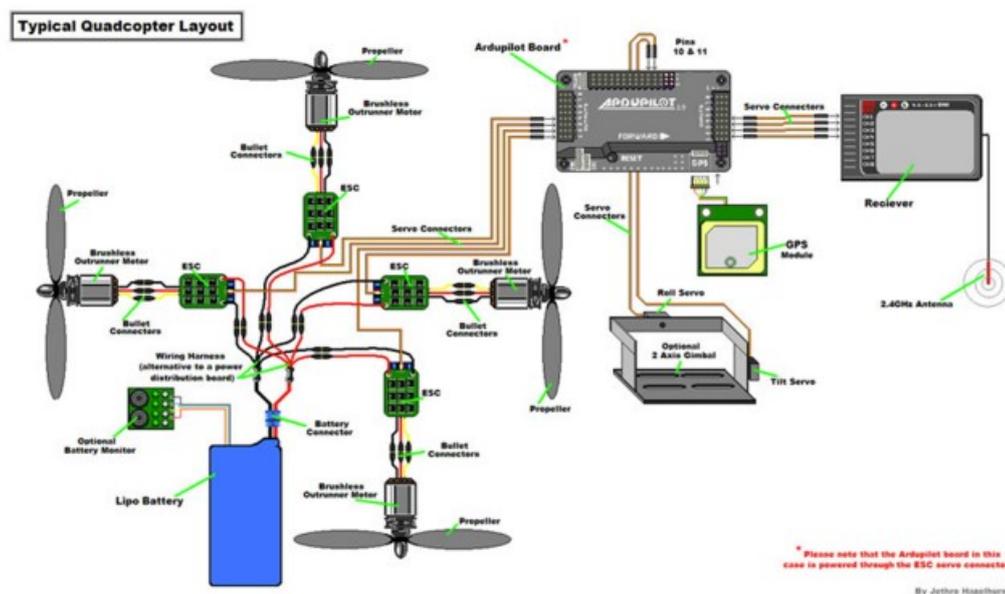
Una vez ensamblado todo el drone, se ha procedido a soldar los cables de los variadores a la placa de potencia, soldando el cable rojo al lado positivo, y el negro al negativo y manteniendo una relación de los dos variadores posteriores van soldados juntos en un extremo, y los anteriores en el otro extremo por si sucede alguna avería poder identificarlos rápidamente. También se ha soldado el conector XT60 de la batería a la placa para así poder alimentarla. Todo el cableado se ha dejado oculto de la mejor manera posible para que el drone se vea mucho más estético.



**Figura 3.18:** Soldadura de los variadores a la placa de potencia

El siguiente paso es la fijación del APM a la placa superior y su posterior conexión de todos los dispositivos (En la Figura 3.19 se puede observar el diagrama del conexionado). Este dispositivo se ha fijado con cinta de doble cara.

En primer lugar se procedió a instalar el receptor de radio y el GPS, colocandolos de una forma estratégica para así disponer del máximo rango de transmisión.



**Figura 3.19:** Diagrama de conexionado al APM

En la controladora de vuelo se han conectado los variadores siguiendo la guía de que el dron es un quadcóptero en forma de “X”, por lo que los variadores se han conectado tal y como se indica en la Figura 3.20 para que los motores 3 y 4 giren en sentido horario, y los motores 1 y 2 en sentido antihorario.



Figura 3.20: Conexión de variadores al APM

### 3.4. Configuración del dron

#### 3.4.1. Comprobación de la configuración de rotación de las hélices

Una vez todo conectado, se ha procedido a la comprobación del sentido de giro de los motores siempre sin las hélices instaladas para evitar accidentes. Realizando la comprobación, había un motor que no giraba en el sentido correcto, para corregirlo, se ha cambiado la polaridad cambiando la posición de dos cables de los variadores entre si. Una vez cambiado, todos los motores giraban de la manera adecuada.

#### 3.4.2. Conexión e instalación del firmware

Una vez instalado el *Mission Planner*, se ha de conectar el dron mediante un cable micro USB entre la controladora de vuelo y el ordenador. Una vez conectado se establece conexión seleccionando desde *Mission Planner* el puerto COM y 115200bps ya que está conectado mediante USB. Siempre que queramos vincular la APM con *Mission Planner* tendremos que hacer este proceso, solo cambiaría en el caso de que se quisiera conectar mediante telemetría, que se usaría 57600bps (Véase la Figura 3.21). La diferencia entre conectar el dron mediante USB o telemetría a Mission Planner son las opciones disponibles en el software, ya que mediante USB tenemos todas las opciones desbloqueadas y podemos hacer cambios en la configuración de la controladora, mientras que vía telemetría únicamente podemos obtener información del vuelo.



Figura 3.21: Conexión de la APM con Mission Planner

Respecto a la instalación del firmware, se realiza mediante la pestaña *Setup* --> *Install Firmware*, donde encontraremos distintos tipos de dispositivos para los cuales hay firmwares disponibles (En la Figura 3.22 se puede observar el menú de instalación de firmware) . En este caso se ha instalado el firmware **Copter 4.1.2 OFICIAL**, ya que corresponde al tipo de drone usado para el proyecto, un quadróptero en forma de “X”.

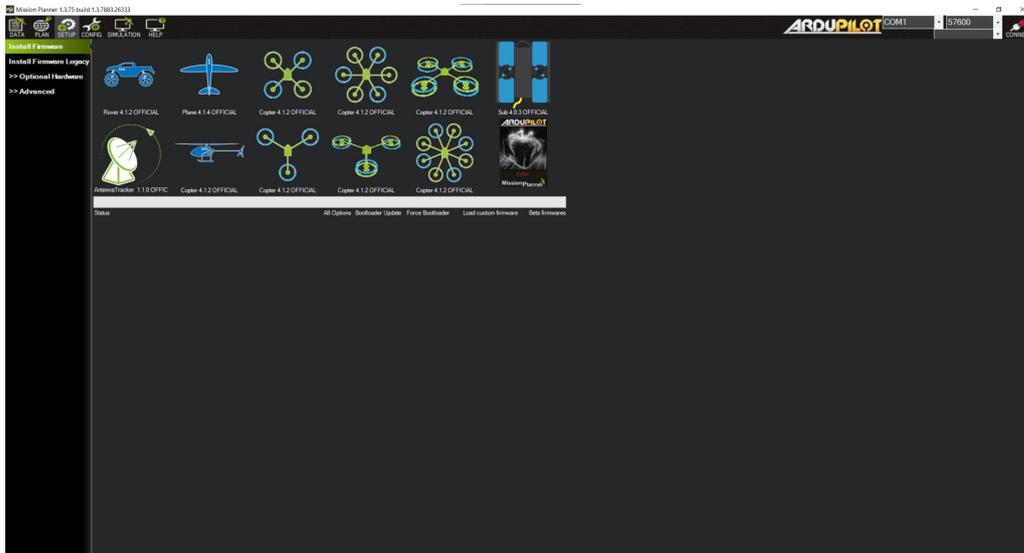


Figura 3.22: Instalación del *firmware*

#### 3.4.3. Calibración de los ESC

A pesar de que no siempre es necesaria esta calibración, hemos procedido a realizarla ya que es necesario que los variadores conozcan los valores máximos y mínimos de PWM que enviará la controladora de vuelo con tal de girar los motores a la velocidad deseada. Los pasos a seguir para la calibración son los siguientes:

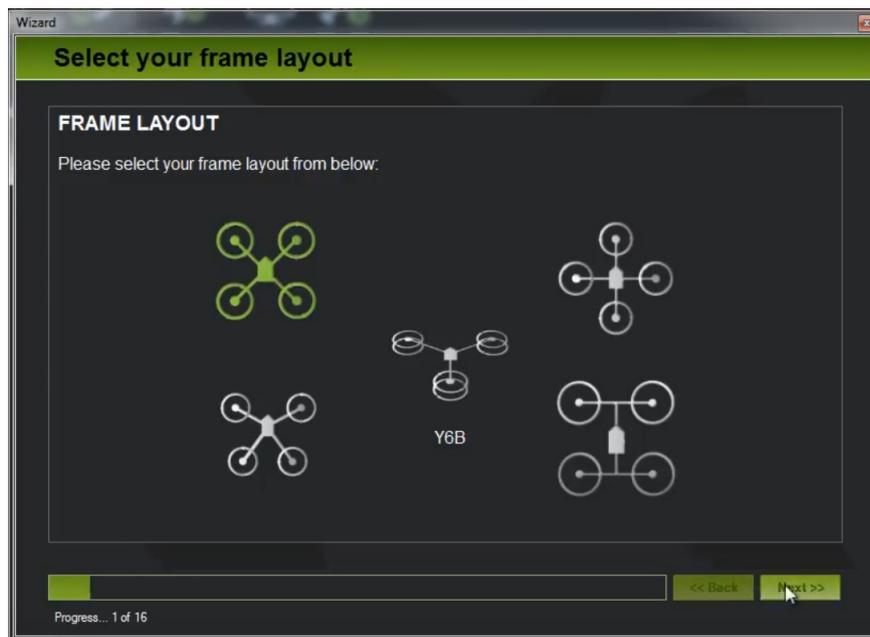
1. Retiraremos las hélices. desconectaremos la batería y encenderemos la emisora con el throttle al máximo.
2. Conectaremos la batería LiPo.
3. Con el throttle en la misma posición desconectaremos y volveremos a conectar la batería. Ahora mismo hemos entrado al modo de calibración de las ESC.
4. Cuando escuchemos un pitido, bajaremos el throttle al mínimo.
5. Los ESC emitirán un pitido largo indicando que se ha realizado la calibración correctamente.
6. Si subimos el throttle poco a poco veremos que los ejes de los motores empiezan a girar, y si lo bajamos dejan de hacerlo.

7. Para finalizar la calibración y salir del modo de calibración, bajamos el throttle y desconectamos la batería.

### 3.4.4. Calibración mediante Mission Planner

Para empezar con la calibración, nos vamos a la pestaña *Setup --> Wizard*, y una vez aquí se nos abrirá una ventana con un proceso de 16 pasos, donde calibraremos el acelerómetro, brújula, radio, etc.

El primer paso de la calibración consiste en la selección de que tipo de frame es el utilizado (Véase la Figura 3.23).



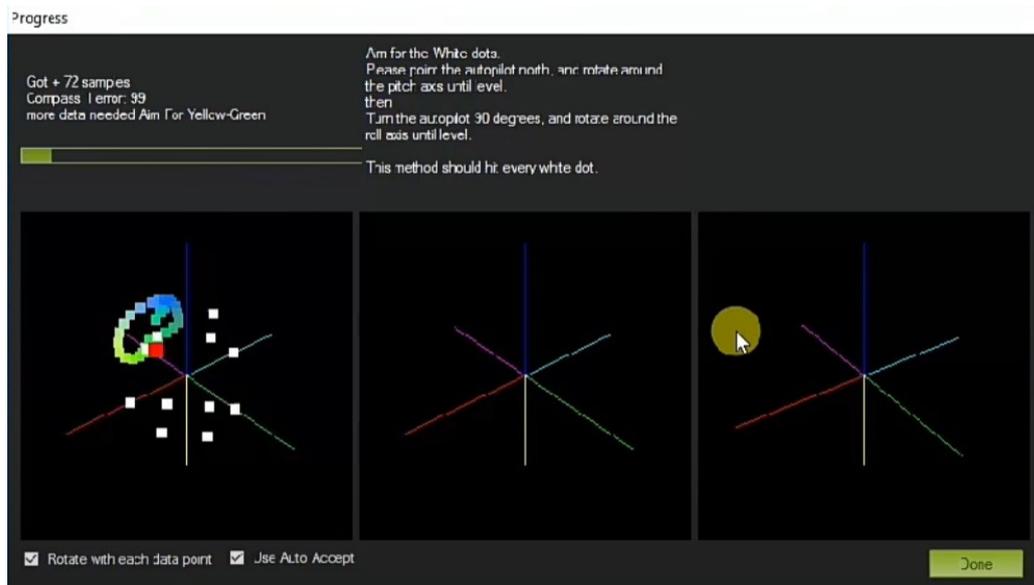
**Figura 3.23:** Selección del tipo de frame para la calibración

Una vez seleccionado el tipo de frame, pasamos al siguiente paso, que es la calibración de el acelerómetro. La calibración del acelerómetro consiste en poner el dron en distintas posiciones (Figura 3.24) que te va indicando el software para así lograr una buena calibración. De una forma indirecta, también se calibra el horizonte artificial, sabiendo en todo momento en que angulo respecto el suelo se encuentra el dispositivo.



**Figura 3.24:** Posiciones de calibración de el acelerómetro

Una vez calibrado el acelerómetro, se procede a calibrar el GPS. Para realizar este paso, se nos abre una ventana (Figura 3.25) donde nos aparecen tres gráficas de 3 ejes, donde mediante giros del drone, se tiene mover el vehículo en todos sus ejes. Este proceso es muy importante, ya que una buena calibración del GPS nos permitirá que el UAV en todo momento sepa su rumbo, y por lo tanto, que pueda funcionar en modo de piloto automático.



**Figura 3.25:** Calibración del GPS

El penúltimo paso de la calibración es la radio (Figura 3.26), el objetivo de esta calibración es que el transmisor reconozca en todo momento en qué posición se encuentran los joysticks. También podemos configurar desde el transmisor el modo de vuelo, controlar el movimiento y orientación del drone y activar o desactivar funciones auxiliares que podemos asignarle mediante la programación de estos. Es muy importante asegurarnos de que la batería no está conectada al drone, ya que al estar manipulando el transmisor, podríamos armar sin querer el dron y provocar un accidente. En la Figura se puede observar la ventana donde manipulando los botones, el programa adquiere el valor máximo y mínimo de los joysticks, y obtiene el valor medio para así saber en todo momento la posición de los mismos.

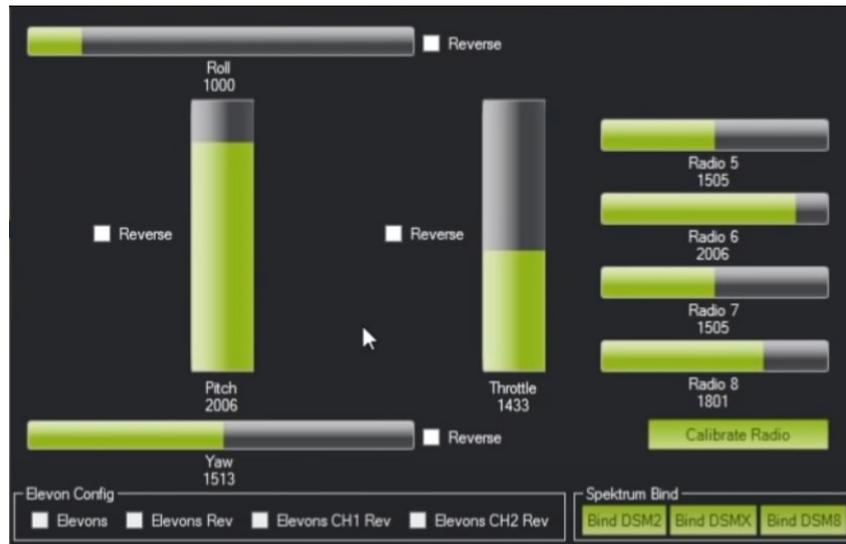


Figura 3.26: Calibración del transmisor

Mirando los menús del transmisor, se puede observar la asignación de canales. Ardupilot, por defecto, asigna los canales de la siguiente manera:

- Canal 1: Roll.
- Canal 2: Pitch.
- Canal 3: Throttle.
- Canal 4: Yaw.

Para finalizar con la calibración desde Mission Planner, el último paso es la asignación de los modos de vuelo. Ardupilot tiene integrados 23 modos de vuelo, de los cuales 10 son los que se utilizan de manera habitual.

Los modos de vuelo se pueden seleccionar tanto desde el transmisor como desde Mission Planner mientras estas conectado por telemetría. Para el desarrollo del proyecto hemos utilizado tres modos de vuelo, asignados al canal 5 del transmisor que se controlan mediante un conmutador.

Los modos son los siguientes:

**Loiter** Es el modo manual mas fácil de usar. Este modo se basa en intentar mantener automáticamente la ubicación, rumbo y altitud en todo momento. Se puede volar como si estuviese en modo manual, pero si no se mueven los joysticks del transmisor, el dron frena y mantiene su posición de vuelo. Es el modo ideal para poder realizar imágenes cuando el dron está quieto manteniendo la posición. Para usar este modo es muy importante tener una buena señal GPS y las menores interferencias posibles.

Los parámetros mas importantes (todos dejados por defecto) en este modo de vuelo son:

- *LOIT\_SPEED*: velocidad horizontal máxima en cm/s.
- *LOIT\_ACC\_MAX*: aceleración máxima en cm/s/s.
- *LOIT\_ANG\_MAX* : ángulo de inclinación máximo en grados.
- *LOIT\_BRK\_ACCEL*: aceleración máxima en cm/s/s durante la frenada.
- *LOIT\_BRK\_DELAY*: retardo en segundos antes de comenzar la frenada.
- *LOIT\_BRK\_JERK*: cambio máximo de aceleración en cm/s/s/s durante la frenada.

**Stabilize** Este modo de vuelo permite volar el dron manualmente, la controladora de vuelo solo se encarga de nivelar el roll y el pitch. El piloto en todo momento tiene que tener bajo control el roll y pitch contrarrestando el empuje del viento. También es necesario tener el control del acelerador para mantener la altura.

Los parámetros mas importantes (todos dejados por defecto) en este modo de vuelo son:

- *ANGLE\_MAX*: Controla el ángulo de inclinación máxima.
- *ANGLE\_RATE\_MAX*: Controla la velocidad de rotación máxima en roll y pitch.
- *ACRO\_YAW\_P*: Controla la rapidez a la que gira el dron en función del yaw de entrada.

**Auto** En modo automático, el vehículo sigue un vuelo preestablecido mediante un archivo cargado al controlador. En este modo el drone sigue unos puntos de coordenadas, denominados *waypoints* (Figura 3.27). Para finalizar el vuelo, al llegar al último waypoint se añade un RTL(Return-to-Launch) donde el drone vuelve a la posición desde la cual ha iniciado el vuelo.

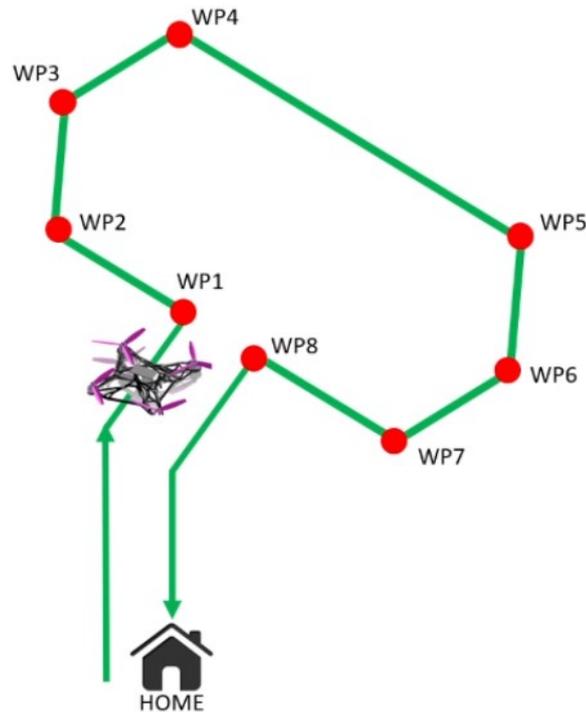
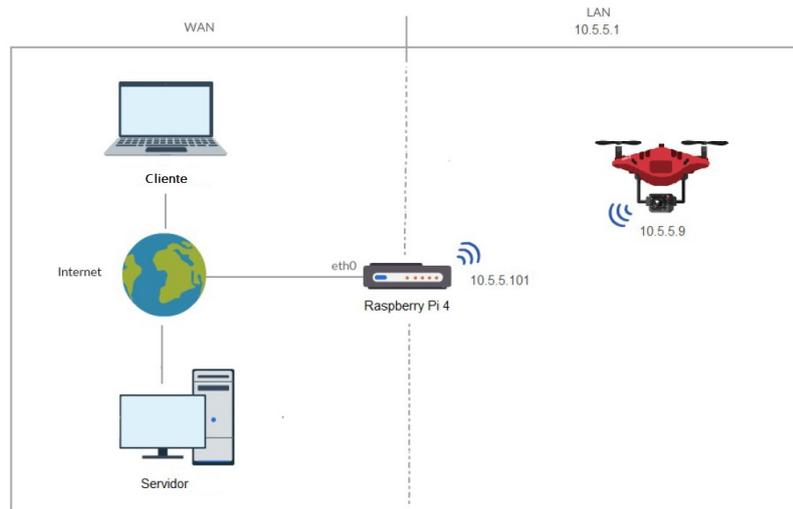


Figura 3.27: Modo Auto

### 3.5. Comunicación GoPro-Servidor

Comprobado el correcto funcionamiento del dron, y que es segura la instalación de la cámara sobre el Gimball, se procede al desarrollo de una aplicación en lenguaje python para establecer una comunicación entre cliente, en nuestro caso una Raspberry Pi, y un servidor implementado en una máquina remota. La finalidad de esta aplicación es tener el control de la cámara, realizar fotografías cada  $x$  segundos y enviarlas a un servidor para poder hacer el procesado de imagen donde se detectará si hay personas o no.

En la Figura 3.28 se muestra un diagrama de como están interconectados los diferentes dispositivos. Nuestro enlace para que se produzca la comunicación entre la GoPro y el servidor es una Raspberry Pi 4. Se han buscado otras alternativas como utilizar un **Dragino LG01-P IoT Gateway** pero estas pruebas no fueron satisfactorias a causa de no poder conectar al mismo tiempo el dispositivo a internet y a la GoPro, por lo que se descartó esta alternativa.



**Figura 3.28:** Comunicación GoPro-servidor

Para lograr establecer la conexión local con la GoPro y una conexión inalámbrica a internet con el servidor, se han utilizado dos scripts ya desarrollados[22], y se han adaptado y modificado a nuestras necesidades. Estos scripts desarrollados y adaptados en Python se pueden encontrar en [23]. Como se ha mencionado antes hay 2 scripts, uno se tiene que ejecutar en la base terrestre, en nuestro caso la Raspberry Pi 4 y el otro script, en un ordenador.

Brevemente, estos dos scripts se encargan de hacer este proceso, la Raspberry se encarga de establecer la conexión con la GoPro de forma local, pide que se realicen las imágenes y las envía al servidor, mientras que el ordenador se encarga de ejecutar el servidor y mediante este recibir y gestionar las imágenes recibidas.

### 3.5.1. Programación del script de la Raspberry Pi 4

Se ha desarrollado el script *client2.py* en Python[24], para que de manera autónoma se lleve a cabo la conexión entre la Raspberry y la GoPro, y se realice la captura de fotografías cada 1.5 segundos. Este código se encuentra presente en [23].

Se ha utilizado una librería ya desarrollada por un usuario, que nos permite el control de la GoPro mediante diversas funciones incluidas en el código que se encuentra en [25].

El proceso que sigue el script es el siguiente:

1. Conexión a la red WiFi de la GoPro.
2. Conexión con la GoPro mediante la librería GoProCamera.
3. Formateo de la tarjeta de memoria para tener únicamente las fotografías del vuelo realizado.

4. Mientras la cámara esté conectada, y cada 1.5 segundos:
  - a) Se envía una orden a la cámara para que haga una fotografía.
  - b) Descarga la fotografía en la memoria interna de la Raspberry.
  - c) Envía la fotografía al servidor mediante el método de solicitud POST a la dirección IP del servidor IP/uploadimage (mediante ethernet).
  - d) Borra la fotografía de la memoria interna de la cámara.
5. Finalmente, se hace una solicitud POST a la dirección IP del servidor IP/json con los siguientes parámetros: nombre de la Raspberry, nombre de la cámara, memoria disponible, % de batería de la cámara, SSID, número de serie y clientes conectados.

Se trata de una aplicación multidisciplinar y muy ampliable, ya que esta pensado para incluir mas funciones y robustas.

### 3.5.2. Programación del script del Servidor ejecutado en el Ordenador

Este script también está desarrollado en Python. La aplicación web está desarrollada con Flask, que nos permite desarrollar de manera muy sencilla una web con arquitectura MVC(Modelo-Vista-Controllador)(Figura 3.29). Esta arquitectura se divide en 3 aplicaciones interconectadas: Base de datos, interfaz de usuario y lógica de control[26].

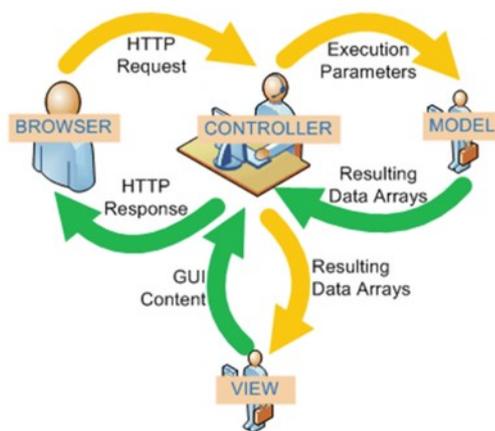


Figura 3.29: Flujo de control MVC

Lo primero para poder ejecutar el script, es instalar todas las dependencias de librerías que no tenemos instaladas, entre ellas la de Flask y para poder tener mas funcionalidades se utiliza la librería Flask-Sqlalchemy, donde se creara una base de

datos para poder guardar los datos de los dispositivos. Siempre funcionaremos en localhost para garantizar una conexión más segura.

El servidor tiene que tener las siguientes funcionalidades:

- Dos tipos de usuario, administrador y invitado.
- El administrador tiene acceso a todas las funcionalidades de la aplicación web.
- El usuario invitado puede enviar datos e imágenes al servidor.
- Gestión de errores.
- Base de datos para el almacenamiento y tener control de todos los datos.

El código *main.py* implementa la lógica de control, los archivos *.html* forman parte de la interfaz de usuario y *devices\_database.sqlite3* forman parte de la base de datos.

### Base de Datos - Modelo

El modelo es el responsable de acceder a la capa de almacenamiento de datos, definir la funcionalidad del sistema y de llevar un registro de las vistas y controladores del sistema.

La base de datos tiene una estructura como la del Cuadro 3.8.

Nom	Tipus	Descripció
id	Integer	Número de identificación único
name	String	Nombre del dispositivo conectado al servidor
camera	String	Nombre de la cámara conectada al dispositivo terrestre
ip	String	IP del dispositivo
connection	String	Fecha y hora de la última conexión
rem_space	String	Espacio disponible en la memoria de la cámara
battery_left	String	% de Bateria restante
clients_connected	String	Número de clientes conectados a la cámara
pictures_left	String	Número de fotografías que se pueden realizar
camera_SSID	String	SSID de la cámara
serial_number	String	Numero de série de la cámara

**Cuadro 3.8:** Estructura de la base de datos

### Interfaz de usuario - Vista

La vista es la responsable de recibir los datos del modelo y mostrarlos al usuario. Se ha desarrollado un entorno web fácil y organizado para el usuario utilizando la biblioteca Bootstrap, ya que contiene herramientas de código abierto para el diseño de aplicaciones web [27].

Todas las aplicaciones web, también necesitan archivos estáticos. Son los denominados archivos CSS, en esta carpeta será dónde se almacenarán nuestra imágenes.

## Lógica de control - Controlador

El controlador es el responsable de recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.).

El código está organizado en rutas `@app.route('/')` para indicar a Flask que URL ha de desencadenar nuestra función. Por defecto, una ruta solo responde a solicitudes de tipo GET, pero se pueden añadir otros métodos HTTP. Las rutas que se han desarrollado son:

- *signin*, métodos GET y POST.
  - Si el método es POST:
    - Comprueba que el usuario y contraseña coinciden con alguno de los registrados en la base de datos y comprueba los permisos otorgados a ese usuario. En caso de coincidir, crea una sesión y te redirige a la página de inicio, si no, te indica que vuelvas a introducir usuario y contraseña.
  - Si el método es GET:
    - Te redirige a la página de la Figura 3.30.

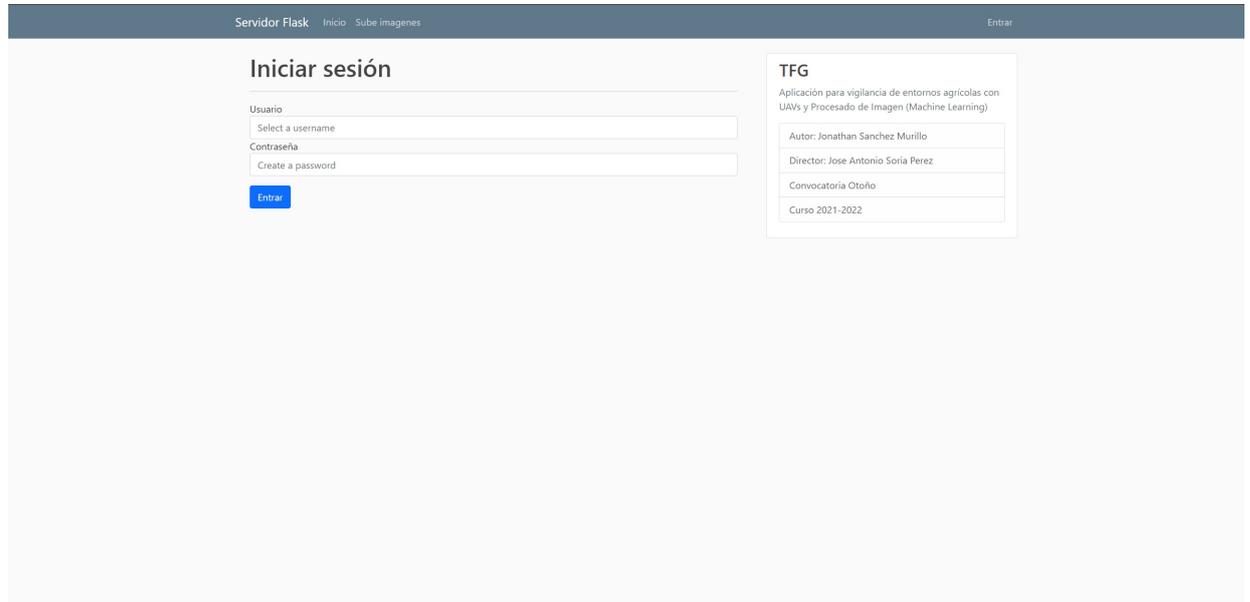
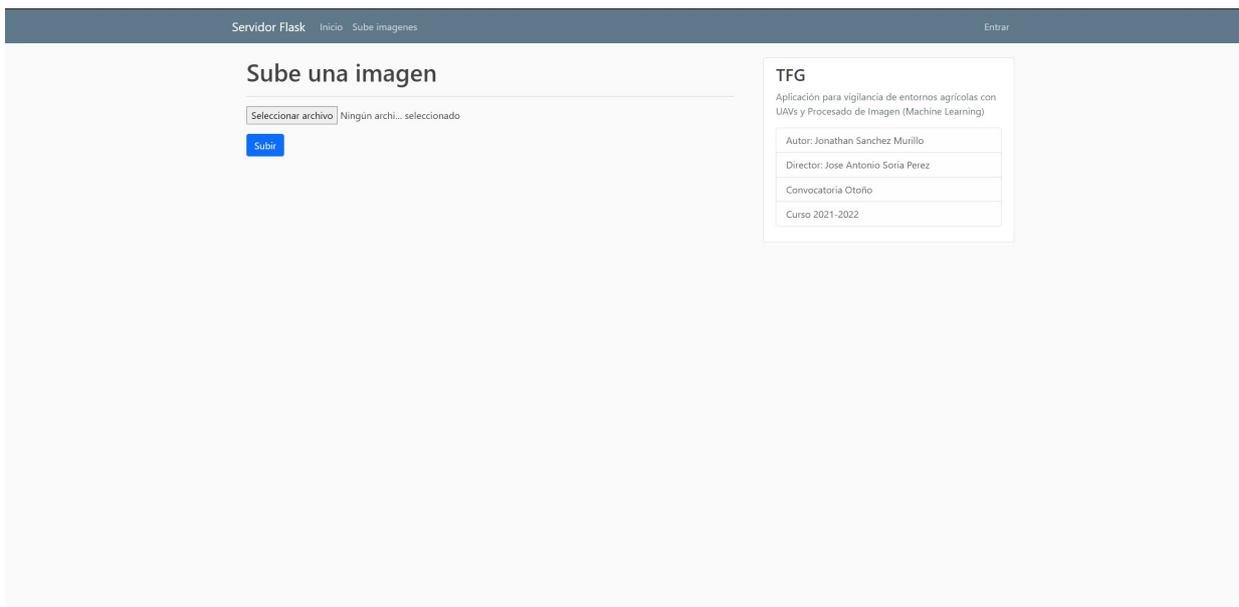


Figura 3.30: Página 'Iniciar sesión'

- *signout*, método GET.

- Borra la sesión y te redirige a la página 'Iniciar sesión'.
- *uploadimage*, métodos GET i POST.
  - Si el método es POST:
    - Se sube la imagen al servidor y se guarda en la carpeta static.
  - Si el método es GET:
    - Te redirige a la página de la Figura 3.31.



**Figura 3.31:** Página 'Sube una imagen'

- Página de inicio, métodos GET y POST.
  - Si el método es POST:
    - Si la sesión tiene permisos de administrador, se descargan todas las imágenes del servidor en un zip, sino, te redirige a la página 'Iniciar sesión'.
  - Si el método es GET:
    - Si la sesión tiene permisos de administrador, se muestran los datos de los dispositivos de la base de datos para poder monitorearlas tal y con se muestra en la Figura 3.32, sino, te redirige a la página 'Inicio'. También se muestran las imágenes.

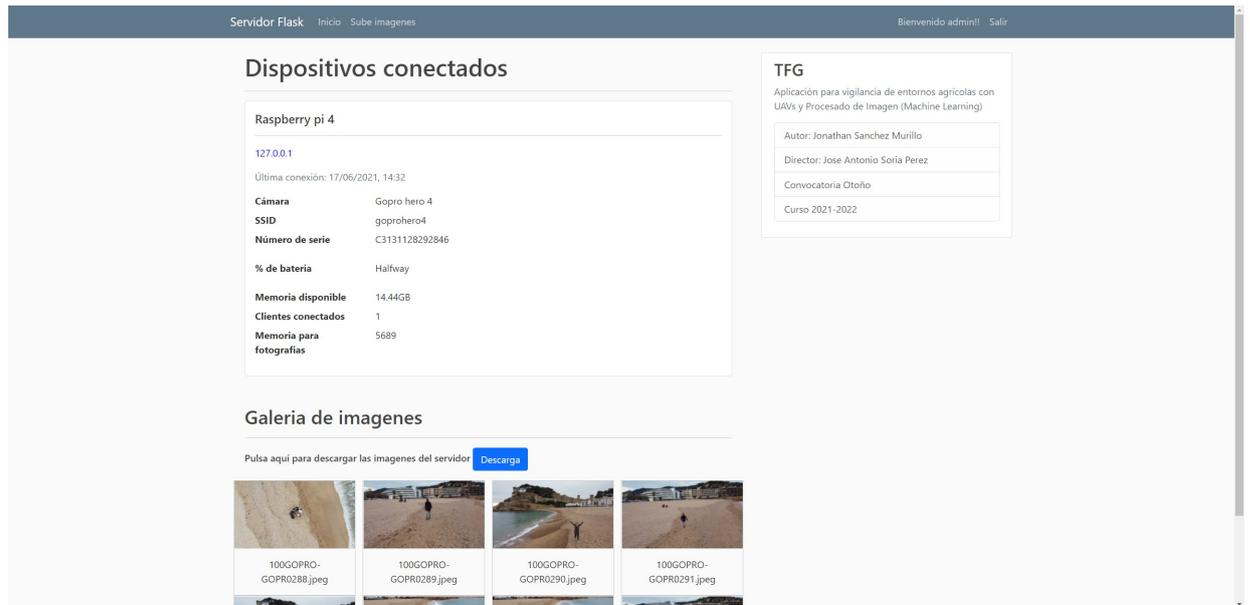


Figura 3.32: Pagina 'Inicio'

# 4 Machine Learning

En este capítulo hablaremos sobre que es el Machine Learning, y de que forma hemos aplicado esta técnica para conseguir un algoritmo que identifique personas a través de imágenes capturadas aéreamente.

## 4.1. ¿Que es el aprendizaje automático?

El machine learning, también llamado aprendizaje automático[28], pertenece al gran mundo de la inteligencia artificial, concretamente en una rama donde tenemos la libertad de que las maquinas aprendan sin ser precisamente programadas para ello. Esta basado en crear un sistema mediante patrones que se hayan en los datos introducidos para así poder lograr predicciones.

Un ejemplo mas claro lo podemos visualizar en la Figura 4.1, comparándolo con la programación clásica, mientras que en la programación clásica se introducen unas reglas manualmente,el programa, y unos datos que se procesaran siguiendo las reglas para obtener el resultado, en el aprendizaje automático o *machine learning* introducimos los datos y las respuestas que esperamos obtener de estos datos, de esta manera obtenemos como resultado las reglas. Una vez tenemos estas reglas, las podemos utilizar para cualquier nuevo dato, obteniendo como resultado la respuesta deseada mediante una predicción. Gracias a este proceso, podemos entender porque en un sistema de aprendizaje automático los expertos dicen que hay que “entrenar” dicho sistema.

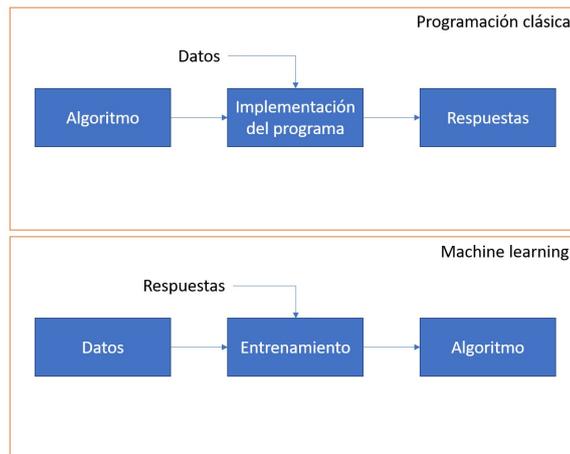


Figura 4.1: Programación clásica vs Machine learning

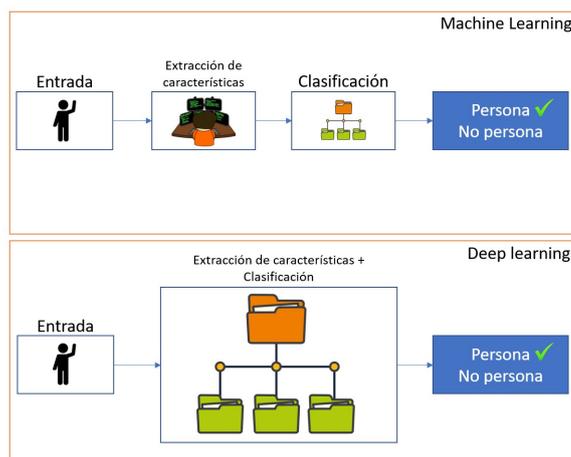
En el aprendizaje automático se trata con un volumen de datos muy grande, ya que cuanto mas grande sea esta base de datos, el entrenamiento que realicemos será mucho mas satisfactorio, y por lo tanto, obtendremos un modelo mucho más preciso a la hora de hacer predicciones.

Entonces, para llevar a cabo un algoritmo de machine learning necesitamos lo siguiente:

- **Datos** de entrada, en nuestro caso, imágenes.
- Imágenes con el **resultado esperado**, en nuestro caso, imágenes con una etiqueta de “persona”.
- Un **feedback** del algoritmo donde poder observar si el entrenamiento se está realizando de una forma satisfactoria, ya que compararemos el resultado obtenido con el esperado. A esto se le llama **aprendizaje**.

#### 4.1.1. Aprendizaje profundo

El aprendizaje profundo[29], también conocido como Deep learning, pertenece al gran mundo del aprendizaje automático. Este se basa en una RCNN conformada por distintos niveles jerárquicos, cuantos más niveles jerárquicos más profundo se considera el modelo. En el nivel inicial la red aprende algo muy simple y envía la información al siguiente nivel, de tal manera que cada vez la información es más compleja hasta llegar al último nivel donde obtenemos el modelo final. En la Figura 4.2 se puede observar las diferencias en el proceso de aprendizaje.



**Figura 4.2:** Proceso Machine Learning vs Deep learning

Por lo tanto, la diferencia entre Machine Learning y Deep Learning es la forma del aprendizaje, ya que el aprendizaje profundo intenta imitar la forma de aprender del cerebro humano por su funcionamiento como neuronas. Por otro lado, el Machine Learning en vez de usar redes neuronales usa arboles de decisión, que están menos evolucionadas que las RCNN.

## 4.2. Redes Neuronales Convolucionales

Las redes neuronales convolucionales o también conocidas como CNN[30] son reconocidas dentro de la categoría de redes neuronales por su efectividad en el reconocimiento de imágenes.

Una CNN esta compuesta por los siguientes procesos:

- Convoluciones
- Función de activación
- Submuestreo o *subsampling*
- Muestreo con Max-Pooling
- Clasificación

### 4.2.1. Convoluciones

Una convolución es un proceso que trata de coger *grupos de píxeles cercanos* de la imagen de entrada, y realizar el producto escalar contra una pequeña matriz que se llama kernel (figura 4.3). El Kernel tiene un tamaño de  $n \times n$  píxeles, y sirve para poder visualizar todas las neuronas de entrada y generar una nueva matriz de salida,

siendo así nuestra nueva capa de neuronas oculta. Por lo tanto, el objetivo que tenemos con la convolución es extraer características de la imagen de entrada teniendo en cuenta que cuantos más filtros tengamos, más características obtendremos durante el proceso de convolución. Según el tipo de Kernel obtenemos una característica diferente, consiguiendo así  $n$  características diferentes.

Una imagen se considera como una matriz de píxeles. Si vemos la figura 4.3, podemos observar que se trata de una matriz 5x5 en color verde, donde los valores de los píxeles son de únicamente 0's y 1's, aunque estos valores normalmente oscilan entre 0 y 255. También podemos observar en la figura 4.3 una matriz 3x3 en color amarillo[30].

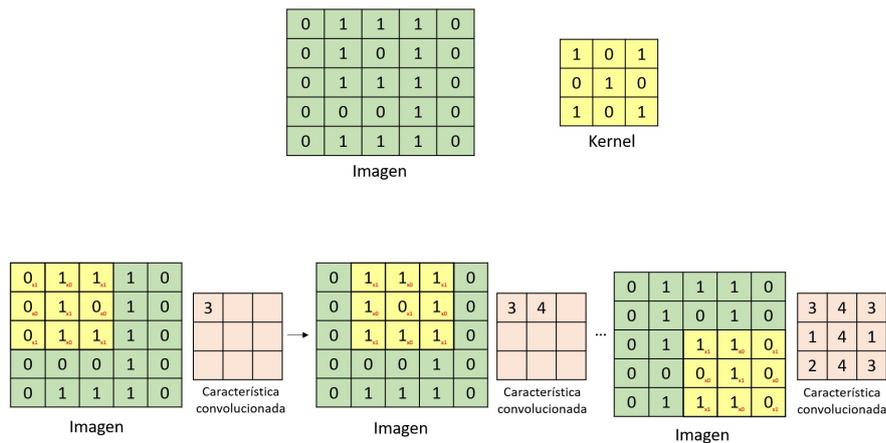


Figura 4.3: Convolución

Desde la parte teórica, la convolución consiste en lo explicado anteriormente, pero en la parte práctica la CNN aprende los valores de los kernels durante su proceso de entrenamiento, aunque si que debemos especificar parámetros como el número de filtros y su tamaño, la arquitectura de red, etc.

La característica convolucionada o mapa de características tiene un tamaño que viene definido por los siguientes parámetros:

- **Profundidad.** Este parámetro se define mediante al numero de kernels que se usan para la convolución.
- **Stride.** El stride es el número de píxeles por los que se desliza la matriz de kernel sobre la matriz de entrada durante el proceso de convolución. Por lo que, cuanto más alto es el stride, más pequeño es el tamaño del mapa de características.
- **Zero-Padding.** Para poder aplicar el filtro a los elementos de los bordes, es conveniente rellenar la matriz de entrada con ceros por el borde.

### 4.2.2. Función de activación

La función de activación mas usada en las CNN es la ReLu, *Rectifier Linear Unit*, y consiste en una función  $f(x) = \max(0, x)$ , tratándose  $f(x)$  de la salida y  $x$  de la entrada[31].

Esta función se aplica a cada elemento de los mapas de características, donde se reemplazan todos los valores de los píxeles negativos por ceros, véase la figura 4.4. La finalidad de este proceso es introducir la no linealidad en la CNN, ya que gran parte de los datos que queremos que la red aprenda son no lineales.

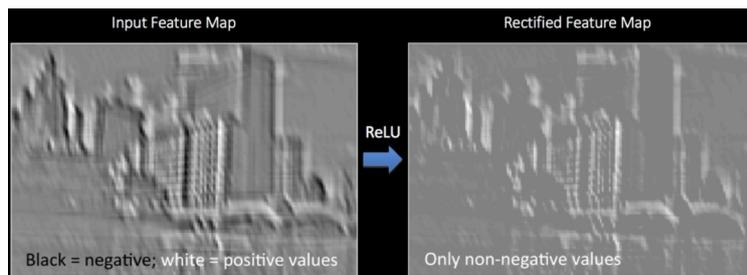


Figura 4.4: Ejemplo de aplicación de la función de activación ReLu

### 4.2.3. Submuestreo o subsampling

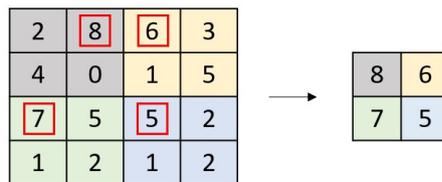
El subsampling es una operación que tiene como resultado reducir la dimensión de cada mapa de características, pero siempre manteniendo la información mas importante de cada uno. Esta operación se hace ya que si no cada capa requeriría un poder computacional muy elevado, llegando a un caso que sería imposible realizarlo.

### 4.2.4. Muestreo con Max-Pooling

El Max-Pooling es el muestreo mas utilizado en las CNN. Consiste en definir una ventana y se coge el elemento más grande de cada mapa de características, generando así una nueva matriz con los datos mas importantes. En la figura 4.5, se puede observar claramente este proceso, donde aplicando Max-Pooling en una Matriz 4x4 con un Stride de 2 y filtros de 2x2, conseguimos reducirla a una matriz 2x2.

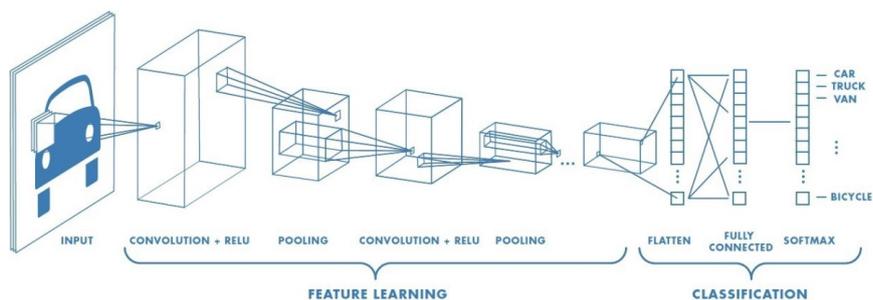
### 4.2.5. Clasificación

Llegados a este paso, se han obtenido varios mapas de activación después de pasar por varias capas. En la última capa tiene lugar la clasificación, donde esta capa esta **totalmente conectada**, esto quiere decir que cada neurona de la capa anterior esta conectada a todas las neuronas de esta última capa. Con esto se pretende usar



**Figura 4.5:** Ejemplo de Max-Pooling

las características de todas las capas anteriores para clasificar la imagen de entrada en alguna de las clases del conjunto de entrenamiento. En la figura 4.6 se puede observar claramente esta ultima capa de clasificación[32].



**Figura 4.6:** Proceso que sigue una imagen en una CNN hasta ser procesada

## 4.3. Implementación

En esta sección trataremos la implementación de una CNN en el proyecto, en nuestro caso se trata de Mask R-CNN, una CNN ya desarrollada por otros usuarios lista para realizar el entrenamiento aunque puedes aplicar modificaciones.

### 4.3.1. Tecnologías usadas para la implementación

Una vez situados en el contexto, pasamos a hablar que tecnologías se han usado para el desarrollo de esta parte del proyecto. Mask R-CNN está desarrollada con el lenguaje de programación **Python**, puesto que hemos usado el código como base, y vamos a tener que modificarlo y añadir elementos, también tenemos que usar este lenguaje. Para realizar las modificaciones hemos usado *Google Colaboratory*, que se trata de un entorno colaborativo que te permite trabajar con notebooks y en lenguaje de programación Python, por lo que esta herramienta nos facilitaba mucho las tareas de ejecución a pesar de sus limitaciones.

Estas limitaciones son las siguientes:

- Memoria de almacenamiento limitada según el volumen de usuarios activos.
- Memoria RAM de un máximo de 12 GB.
- 12 horas de ejecución continuas como máximo, pudiendo ser desconectado previamente si hay mucho volumen de usuarios.

Cabe decir que hay una versión **PRO** de *Google Colab*, que te da mejores prestaciones, pero que no está disponible actualmente en España.

### 4.3.2. Datasets utilizados

Para el desarrollo del proyecto, hemos creado un dataset que contiene 129 imágenes, de las cuales se utilizan 89 para el entrenamiento, 11 para la validación y 29 para el test. Sabemos que se trata de un dataset muy pequeño para realizar un entrenamiento, pero esto se debe a las limitaciones de Google Colab y de no disponer de una máquina para poder realizar el entrenamiento de la CNN.

Las imágenes para el entrenamiento las hemos etiquetado usando **Makesense.ia**, donde nos dejaba guardar directamente todo el dataset en formato **.json**.

En la figura4.7 podemos ver un ejemplo de como quedaría una imagen etiquetada.

### 4.3.3. Selección de clases a detectar

Mask R-CNN es capaz de detectar 80 clases por defecto, pero para el proyecto desarrollado unicamente necesitamos la clase *persona*, por lo que en las imágenes del dataset, unicamente se han introducido imágenes donde salen personas y están debidamente etiquetadas.

### 4.3.4. Entrenamiento de la R-CNN

Para realizar el entrenamiento de la R-CNN hemos utilizado el dataset mencionado anteriormente, y aplicando los siguientes valores a estos parámetros:

- Épocas: Una epoca se refiere a cuando un conjunto de datos entero se pasa hacia adelante y hacia atrás a través de la red neuronal una sola vez. Se trata de un proceso iterativo ya que el entrenamiento de la red neuronal también lo es.
- Pasos: Es un parámetro que va vinculado al numero de imágenes y ejemplos de entrenamiento en un solo lote, aunque nosotros hemos dejado fijado el valor de 500.
- Clases: Tenemos 2 clases ya que tratamos de clasificar entre persona y no persona, por lo tanto su valor es de 2.



Figura 4.7: Ejemplo de imagen etiquetada

### 4.3.5. Métricas de evaluación

Para evaluar un modelo en detección de objetos se pueden utilizar diferentes métricas que miden la calidad del modelo, y por lo tanto de sus predicciones. A continuación se explican 2 de las mas usadas, una de ellas la aplicada en el proyecto, la matriz de confusión.

#### 4.3.5.1 Incorporación de la matriz de confusión

Para el análisis de los resultados se empleará la matriz de confusión (Cuadro 4.1)[33]. Esta matriz de confusión evalúa los resultados obtenidos dividiéndolos en verdaderos positivos, falsos positivos, falsos negativos y verdaderos negativos. Gracias a estos valores se pueden calcular las métricas de *precision* y *recall*.

**Verdaderos positivos (TP):** Predicciones correctas respecto a los datos reales.

**Falsos positivos (FP):** La predicción corresponde a una clase que no se encuentra en los datos reales.

**Verdaderos negativos (FN):** Cuando en la predicción no se detecta la clase o se detecta una clase errónea y se clasifica como perteneciente a otra clase.

**Falsos negativos (TN):** Valores de detección que no son detectados y realmente no existen en los datos reales

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

**Cuadro 4.1:** Matriz de confusión

La métrica *Precision* se utiliza para medir el porcentaje de predicciones positivas verdaderas entre los casos positivos. Se calcula mediante la ecuación 4.1

$$Precision = \frac{TP}{TP+FP} \quad (4.1)$$

La métrica *Recall* tiene la finalidad de medir el porcentaje de predicciones correctas entre el total de predicciones realizadas. Se calcula mediante la ecuación 4.2

$$Recall = \frac{TP}{TP+FN} \quad (4.2)$$

La métrica *Accuracy* nos proporciona el porcentaje total de los aciertos de nuestro modelo y se calcula según la ecuación 4.3

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4.3)$$

Finalmente tenemos un ultimo parámetro que nos ayuda cuando dos modelos tienen baja *Precision* y alto *Recall*, o viceversa. Esta métrica se llama *F-score*, que nos ayuda a medir el *Recall* y *Precision* al mismo tiempo. Utiliza la media media armónica en vez de la media aritmética castigando así mas a los valores extremos. Se calcula mediante la ecuación 4.4

$$F - score = 2 * \frac{Recall * Precision}{Recall + Precision} \quad (4.4)$$

### 4.3.5.2 Intersection over Union

La IoU es una métrica que suele usarse en detección de objetos, esta se utiliza para medir la precisión de un detector de objetos en un conjunto de datos en particular.

Para aplicar este modelo se necesitan tanto las cajas reales, por lo tanto los que hemos etiquetado mediante la herramienta *makesense.io* y las cajas obtenidas mediante la predicción del modelo. El cálculo que se realiza consiste en una división del área sobrepuesta por las dos cajas entre el área total que ocupan las 2 cajas (Véase la Figura 4.8). Por lo que, un resultado satisfactorio será cuando el valor de la IoU sea mayor que un valor de la predicción[34].

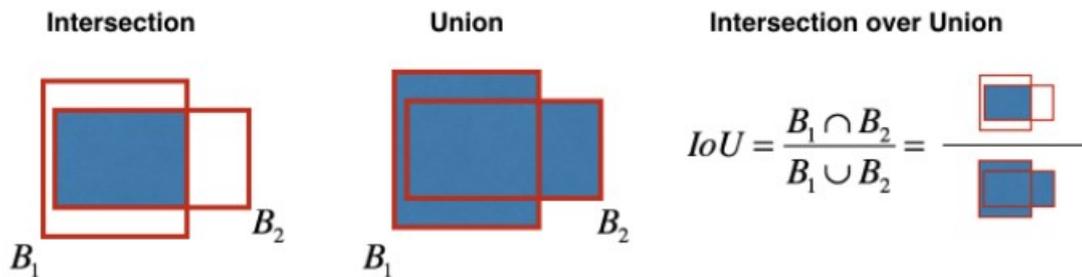


Figura 4.8: Cálculo de la métrica IoU

## 4.4. Desarrollo del código

En este apartado se explican los pasos a seguir para el entrenamiento del modelo.

### 4.4.1. Definición del modelo

Para el desarrollo del proyecto se ha utilizado el modelo Mask R-CNN adaptado con los parámetros que necesitamos[35]. Se trata de un modelo pre-entrenado, lo que se ha adaptado para únicamente la detección de 2 clases como se ha mencionado anteriormente, la detección del *background* y *persona*. Es un modelo implementado con Python 3, Keras y Tensorflow.

### 4.4.2. Definición del dataset

Como ya se ha mencionado en el apartado 4.3.2, el conjunto de datos se divide en dos subconjuntos:

- **Entrenamiento**, donde se encuentran el 90% de los datos empleados para enseñar a la red lo que queremos que aprenda a clasificar.

- **Validación**, se encuentran el 10% restante de los datos y se utilizan para validar el modelo, seleccionando así los hiperparámetros mas adecuados para obtener el modelo más óptimo.

### 4.4.3. Parámetros de la CNN

Los parámetros que hemos de tener en cuenta a la hora de obtener el modelo son los siguientes:

- **IMAGE\_MIN\_DIM**: Es el valor de la dimensión máxima que tendrá la imagen entrenada. Su valor es fijo, 512.
- **IMAGE\_MAX\_DIM**: Es el valor de la dimensión mínima que tendrá la imagen entrenada. Su valor es fijo, 512.
- **STEPS\_PER\_EPOCH**: Es el numero de veces que se repite el proceso en un solo conjunto de datos. Su valor es fijo, 500.
- **DETECTION\_MIN\_CONFIDENCE**: Es el valor que estipula si esta detectando o no, ya que cuando el porcentaje de precisión es superior a lo establecido, considera que si que pertenece a esa clase. Su valor es fijo, 0.9.
- **epochs**: Es el valor que define el numero de veces que se repite el proceso hasta acabar el entrenamiento. Su valor es fijo, 100.
- **num\_classes**: Es el número de objetos a clasificar, en nuestro caso 2, clasificar entre persona y no persona.



# 5 Resultados

En este capítulo se describen las pruebas realizadas para comprobar el correcto funcionamiento del dron, del servidor y de la red neuronal y también los resultados de estos. Se han ejecutado un total de cuatro vuelos en diferentes modos de funcionamiento, tanto en manual como automático. Se han analizado una serie de datos de los vuelos obtenidos mediante telemetría para verificar que el dron funciona en condiciones óptimas. Respecto al servidor, se ha verificado su funcionamiento en local. En cuanto a la red neuronal, se han pasado imágenes obtenidas mediante la comunicación entre la GoPro y la unidad terrestre para poder analizar su rendimiento.

## 5.1. Primera prueba de vuelo

La primera prueba de vuelo se ha realizado en la pista polideportiva del campus de la ESEIAAT en Terrassa, un espacio delimitado y protegido con vallas y redes, de tal forma de que si el dron se descontrola o desestabiliza no pueda salir de la zona y así evitar posibles accidentes. El objetivo de este primer vuelo es comprobar que el dron responde correctamente a todas las órdenes mandadas por el transmisor.

Iniciamos el vuelo con el modo *stabilize*, que nos permite controlar el dron de manera manual. Se ha cargado la batería al nivel máximo de carga para realizar la prueba. El primer paso es armar el dron, esto se consigue orientando ambos joysticks a la parte inferior central del transmisor (Figura 5.1).



Figura 5.1: Como armar drone

Una vez hemos armado el drone, subiendo el throttle se han comprobado los movimientos de *Roll*, *pitch*, *yaw*. El drone tiene el sentido de una de las hélices en sentido inverso, para corregir este sentido, se han intercambiado dos cables de las ESC para cambiar la polaridad y se ha corregido el error.

Una vez comprobado estos movimientos el dron ya puede volar de manera manual.

## 5.2. Segunda prueba de vuelo

El objetivo de esta segunda prueba es realizar un vuelo en el modo asistido *Loiter* descrito en la sección 3.4.4. Esta prueba también se ha realizado en la pista polideportiva del campus de la ESEIAAT en Terrassa. Esta vez la prueba de vuelo ha sido satisfactoria, el drone responde de manera correcta a todos los comandos mandados por el transmisor. El siguiente paso es configurar un plan de vuelo y realizar un vuelo de manera automatizada, para ver si cumple con nuestras necesidades. Para realizar esta prueba se tiene que realizar en un espacio más abierto, para así poder comprobar su correcto funcionamiento.

## 5.3. Tercera prueba de vuelo

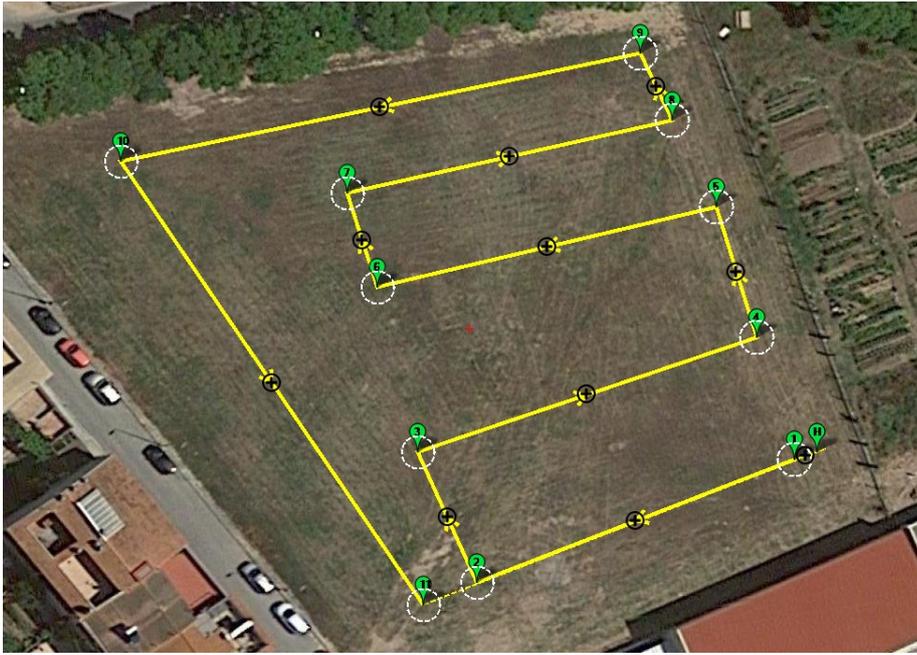
El objetivo de la tercera prueba de vuelo es analizar el rendimiento de los modos asistidos *Loiter* y *Auto*. Se han realizado las pruebas en una zona aislada en Vilanova i la Geltrú 5.2. Para la realización de la prueba de vuelo, primero hay que definir

el área donde se va a realizar el vuelo automático. Una vez definido el área, se ha preparado un plan de vuelo cubriendo toda la zona, estimando el tiempo de vuelo y modificando los parámetros de vuelo.



**Figura 5.2:** Zona de pruebas de vuelo

Para poder conseguir el mejor rendimiento de vuelo es muy importante tener una buena señal GPS, poca interferencia electromagnética y con ráfagas de viento bajas. La señal GPS es limitada, únicamente se localizaban 13 señales GPS, cuando lo mínimo recomendable para realizar un vuelo en buenas condiciones y seguro, debe ser de 10 satélites. Como es una zona con huertos privados, es un buen entorno para probar el modo automático. Para la realización de un vuelo automático primero es necesario crear un plan de vuelo en pestaña Flight Plan de Mission Planner. En la siguiente imagen 5.3 podemos observar el plan creado para el vuelo automático.



**Figura 5.3:** Plan de vuelo

El plan de vuelo forma un conjunto de 11 waypoints. El primer waypoint está configurado como Take off. Los waypoints están definidos a una altura de 15 metros. El último waypoint está definido como Return to Launch.

Durante la realización del vuelo se pudo observar que realizaba correctamente los distintos waypoints, pero al no realizar una parada y orientarse hacia el siguiente waypoint el drone se desorientó y perdió su ubicación. Al realizar los giros acabó desorientándose y se tuvo que finalizar haciendo un aterrizaje forzado. Al ver el resultado del vuelo, se modificó el plan de vuelo, generando 3 waypoints a una altura de 5 metros bastantes alejados entre ellos y en línea recta para evitar giros pronunciados. A la hora de realizar el siguiente plan de vuelo, el drone ignoró por completo el plan de vuelo previamente creado. Procedió a volar de manera autónoma a una altura por encima de la establecida. Seguidamente, se forzó el cambio de modo a manual para poder controlarlo y aterrizarlo de una forma segura. A pesar del descontrol, se consiguió aterrizar el drone con un pequeño impacto contra el suelo.

Cuando se recuperó el drone, conectamos el ardupilot a la aplicación Mission Planner para visualizar el registro de datos. En ellos se observó un error en la brújula tal y como se muestra en la Figura 5.4.

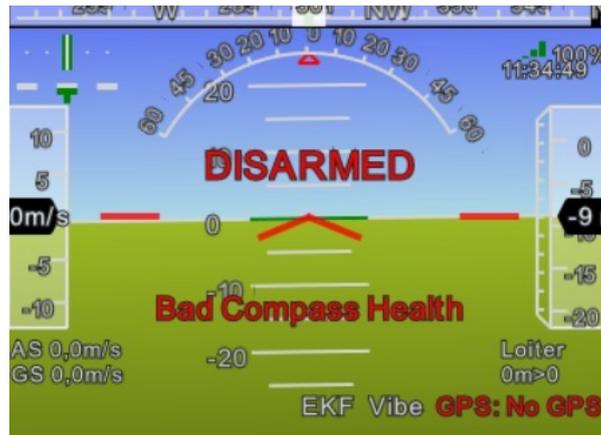


Figura 5.4: Error Mission Planner

## 5.4. Cuarta prueba de vuelo

En la última prueba de vuelo se ha realizado en la misma ubicación que el tercer vuelo. A causa del impacto recibido por el aterrizaje forzoso de la tercera prueba de vuelo, se ha tenido que cambiar el ardupilot instalado, para garantizar un correcto funcionamiento del drone.

Para la realización de un vuelo automático es necesario volver a generar un plan de vuelo tal y como se muestra en la siguiente Figura 5.5, se ha generado un plan de vuelo mucho más sencillo que el del anterior vuelo por seguridad.

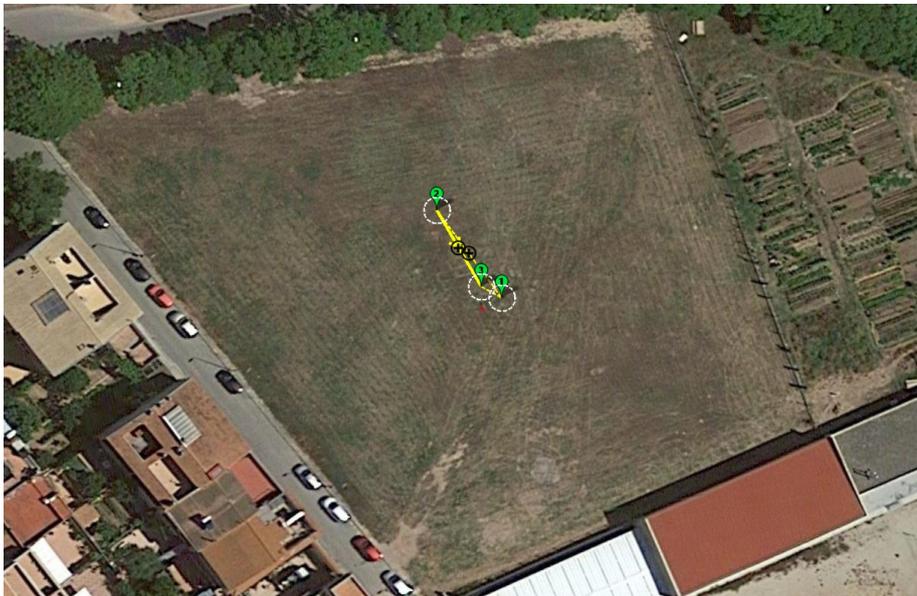


Figura 5.5: Plan de vuelo

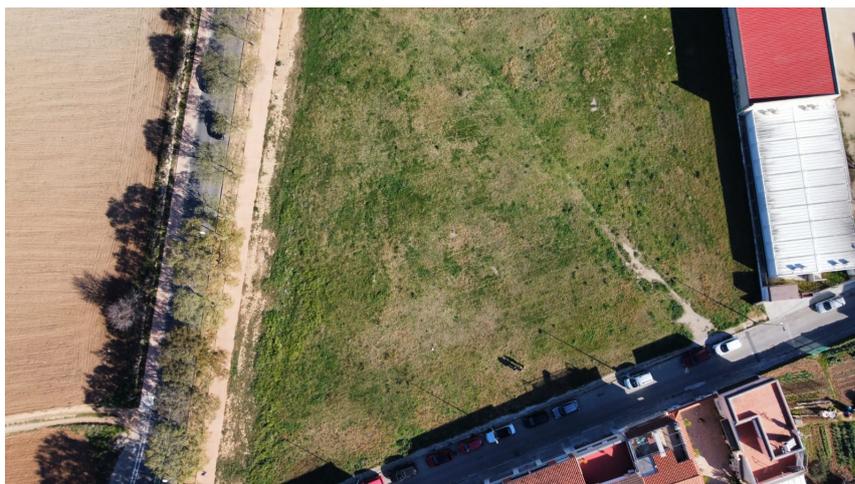
## Resultados

Esta vez el vuelo realizado de manera automatizado ha sido satisfactorio. En el siguiente Cuadro 5.1 se pueden observar alguno de los datos del vuelo realizado.

Área mapeada	150.04m <sup>2</sup>
Velocidad	4 m/s
Distancia recorrida	52.12 m
Tiempo total	0:01:34
Número total de Waypoints	3
Altura máxima	10 m

**Taula 5.1:** Datos del vuelo

Durante la realización del vuelo se han capturado imágenes de la ubicación (Figura 5.6). A partir de aquí el agricultor puede tener una vista área de los cultivos para la vigilancia. Gracias al Gimbal se puede obtener imagenes como la Figura 5.7.



**Figura 5.6:** Imagen aérea



Figura 5.7: Imagen aérea del terreno

## 5.5. Procedimientos para realizar vuelos

A continuación se muestran los procedimientos que se han de seguir para de realizar vuelos en piloto automático y recibir los datos e imágenes en el servidor.

1. Preparar el plan de vuelo y obtener el archivo '.waypoints'.
2. Una vez ubicados en la zona de vuelo, conectar la batería y telemetría al dron.
3. Escribir en el piloto automático el archivo '.waypoints'.
4. Leer el archivo '.waypoints' para comprobar que la ruta se ha escrito correctamente.
5. Colocar la GoPro en el Gimball y activar el WiFi de la cámara.
6. Encender la Raspberry y arrancar el programa.
7. Comprobar en que botón del transmissor está configurado el modo Auto.
8. Armar el dron y subir la palanca de gases.
9. Activar el modo Auto.

## 5.6. Análisis de los registros de datos

Los registros de datos o *Dataflash Logs* se almacenan en la memoria del piloto y se pueden descargar después de un vuelo. Por defecto, se crean después de armar el vehículo por primera vez. El procedimiento para obtener los datos del piloto automático es:

1. Conectar el dron a la estación terrestre utilizando el cable micro USB.

## Resultados

2. Seleccionar la ventana "DataFlash Logs".
3. Pinchar en el botón "Download DataFlash Log Via Mavlink".

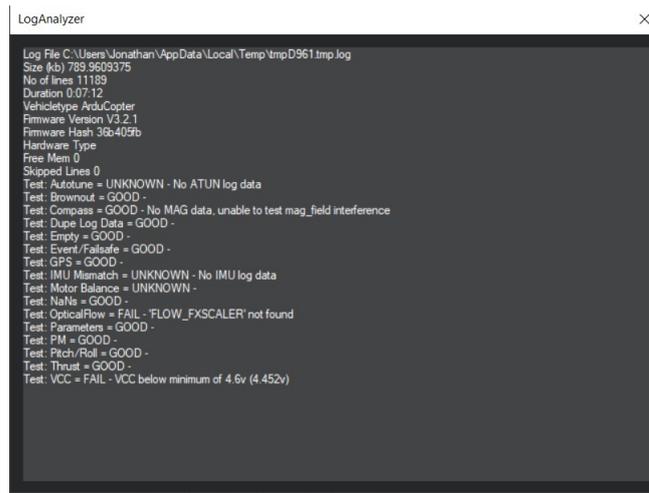
Por defecto los archivos vienen en formato .bin o .log., pero *Mission Planner* permite convertirlos en formato .mat compatible con MATLAB. Se ha utilizado el programa localizado en [23]. El registro de datos también incorpora un fichero .klm que permite visualizar la ruta realizada con Google Earth en 3D (Figura 5.8).



**Figura 5.8:** Ruta realizada en 3D

### 5.6.1. Análisis automático de datos de telemetría

El análisis más sencillo consiste en generar un informe automatizado básico que ponga en conocimiento las áreas problemáticas más habituales. Por eso, es necesario acceder a "Análisis de registros" y seleccionar el registro correspondiente del directorio de MissionPlanner / logs. Una vez seleccionado el registro que se genera el informe de la Figura 5.9.



```

LogAnalyzer
Log File C:\Users\Jonathan\AppData\Local\Temp\tmpD961.tmp.log
Size (kb) 789.9609375
No of lines 11189
Duration 0:07:12
VehicleType ArduCopter
Firmware Version V3.2.1
Firmware Hash 36-405b
Hardware Type
Free Mem 0
Skipped Lines 0
Test: Autotune = UNKNOWN - No ATUN log data
Test: Brownout = GOOD -
Test: Compass = GOOD - No MAG data, unable to test mag_field interference
Test: Dupe Log Data = GOOD -
Test: Empty = GOOD -
Test: Event/Falsafe = GOOD -
Test: GPS = GOOD -
Test: IMU Mismatch = UNKNOWN - No IMU log data
Test: Motor Balance = UNKNOWN -
Test: NaNs = GOOD -
Test: OpticalFlow = FAIL - FLOW_FXSCALER not found
Test: Parameters = GOOD -
Test: PM = GOOD -
Test: Pitch/Roll = GOOD -
Test: Thrust = GOOD -
Test: VCC = FAIL - VCC below minimum of 4.6v (4.452v)

```

**Figura 5.9:** Log Analyzer

Como se puede observar, no todos los parámetros que el piloto guarda se han registrado. En cambio, los de las áreas más relevantes se han registrado correctamente y este primer análisis los da como buenos (*GOOD*).

### 5.6.2. Calidad del GPS

Cuando el dron se encuentra en modos autónomos (Loiter, RTL, Auto, etc.), el GPS juega un papel fundamental, ya que es quien suministra la geolocalización de la aeronave. Los errores de posición del GPS pueden hacer pensar que se encuentra en otra posición de forma repentina y provocar movimientos agresivos para poder corregir el error en la posición, tal y como nos sucedió en las pruebas de vuelo.

La Figura 5.10 muestra los valores de hdop y número de satélites que recibe el GPS tanto por un entorno urbano. Un valor hdop inferior a 1,5 puede considerarse muy bueno mientras que valores superiores a 2 pueden indicar posiciones del GPS no muy buenas. El valor hdop está relacionado con el número de satélites captados. Como puede observarse, en un entorno urbano se llegan a captar 14 satélites.

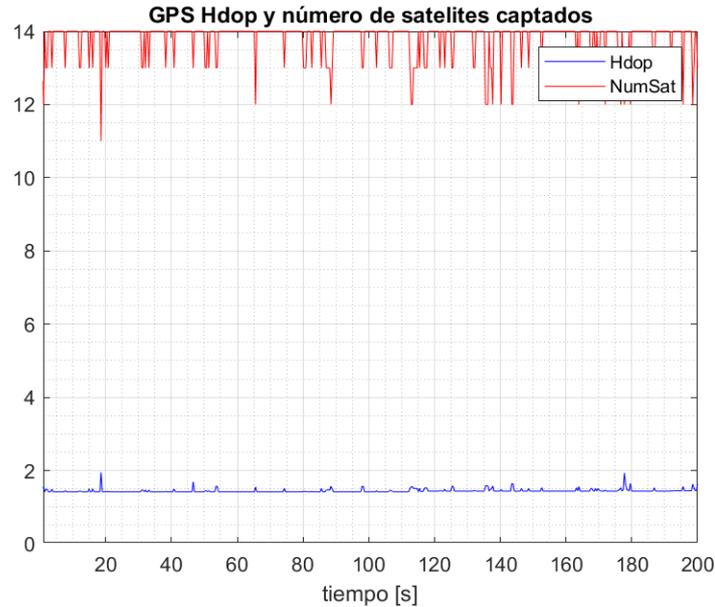


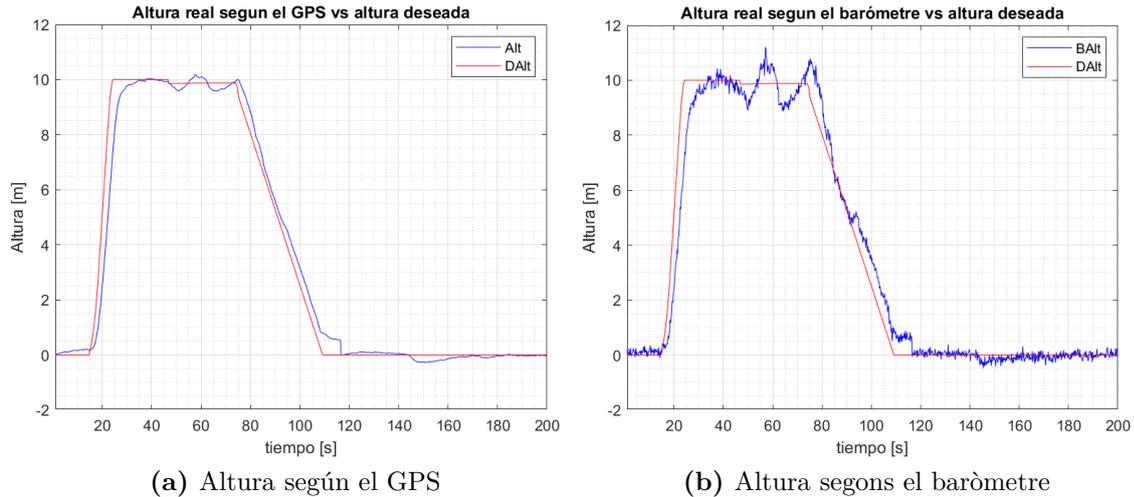
Figura 5.10: GPS Hdop y número de satélites

### 5.6.3. Altura

La Figura 5.11 muestra la altura real de vehículo respecto al terreno (azul) medida por el barómetro y la altura medida por el GPS. Se puede comprobar que el vehículo sigue la altura programada en la sección 4.2 de 15 metros con ligero desvío. Se puede comprobar también que la altura proporcionada por el GPS es más precisa que en el barómetro, pasando de tener errores de 0.5 m a errores de hasta 2 m. Como el procedimiento de aterrizaje se hace también en automático es importante que la altura proporcionada por el GPS sea precisa por carecer de aterrizajes bruscos. Para reducir este error debería ajustarse los parámetros del PID de la controladora

## 5.7. Comunicación GoPro-Servidor

Se ha comprobado el funcionamiento del servidor adaptado para la comunicación entre la GoPro y el servidor lanzándolo desde la Raspberry Pi y manteniendo el dron en tierra en estas primeras pruebas. Se ha conectado la Raspberry a un pequeño monitor para poder visualizar la interfaz. Cuando se ejecuta el código, la Raspberry se conecta a la red WiFi de la cámara y formatea la memoria de esta para disponer de todo el espacio disponible a la hora de realizar el vuelo. Cada 2 segundos se realizan fotografías, y éstas nada más realizarse tardan como máximo 0.5 segundos en transferirse al servidor, este tiempo depende en todo momento del tamaño de la imagen.



**Figura 5.11:** Altura de vuelo

Se ha verificado que únicamente el usuario “admin” es capaz de acceder a la página principal, y que es imposible acceder con otro usuario o contraseña que no esté registrado. Una vez iniciada sesión, como se puede ver en la Figura 5.12, las fotografías se han recibido correctamente y podemos visualizar datos como la SSID, el nombre de la cámara, la memoria disponible, etc.

El servidor funciona correctamente y está preparado para futuras aplicaciones como la implementación de la detección de personas en el servidor, o la recepción de datos de sensores colocados por los campos de cultivo y conectados a la Raspberry sin tener que hacer ninguna modificación en el código.

## 5.8. Procesado de imagen

En esta sección analizaremos los resultados obtenidos durante la etapa del procesado de imagen. El entrenamiento con Google Collab no ha sido del todo satisfactorio, ya que esta herramienta limita el entrenamiento a 12 horas, y te desconecta de sus servidores cuando están colapsados para dar prioridad a los usuarios con suscripción prémium. Esta suscripción no está disponible en España, así que el entrenamiento máximo logrado ha sido de 19 épocas, por lo tanto, se usarán los pesos obtenidos en esta época para la obtención de los resultados del procesado de imagen.

### 5.8.1. Matriz de confusión

Para poder evaluar los resultados de la matriz de confusión, evaluaremos el conjunto de imágenes usadas para *Train* y *Test*. (Cuadro 5.2)

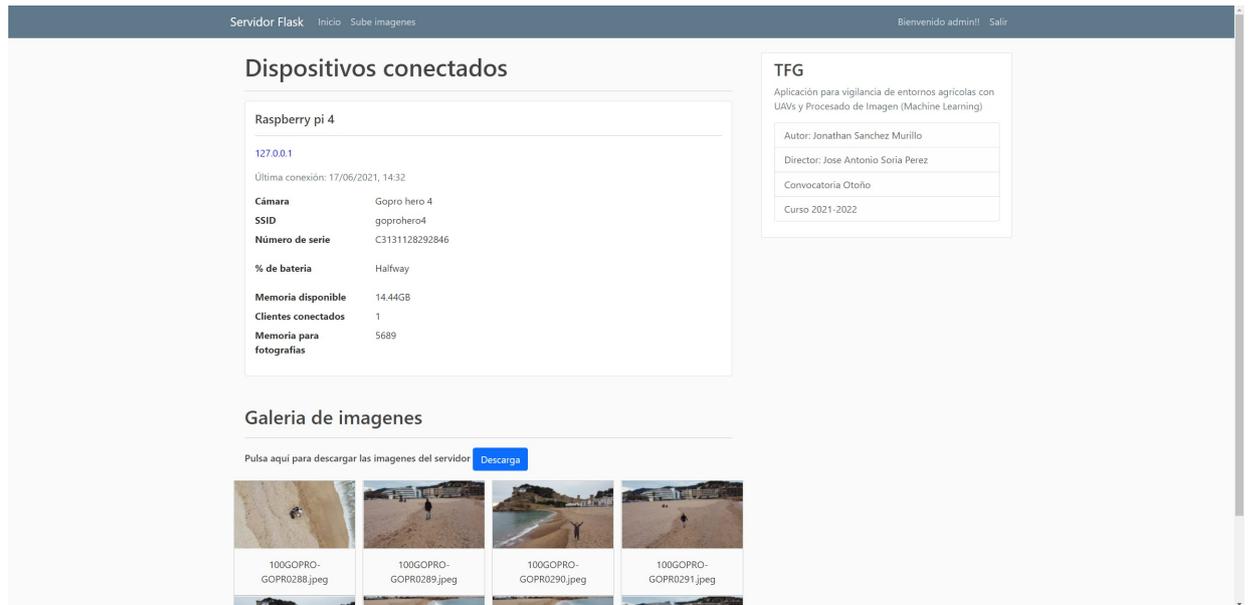


Figura 5.12: Página principal del servidor

(a) Train

		Predicción	
		Background	Persona
Realidad	Background	16	2
	Persona	24	47

(b) Test

		Predicción	
		Background	Persona
Realidad	Background	5	1
	Persona	6	17

Taula 5.2: Matriz de confusión

Una vez tenemos las matrices de confusión, podemos calcular distintas métricas que nos van a proporcionar que tan bueno es nuestro modelo para la detección de personas.

**Precision** Con esta métrica medimos la calidad del modelo. Usaremos la fórmula indicada en el apartado 4.1:

$$precision - train = \frac{47}{47 + 2} = 0.9591, \text{ para el conjunto de datos de train}$$

$$precision - test = \frac{17}{17 + 1} = 0.9444, \text{ para el conjunto de datos de test}$$

Con estos resultados podemos observar es bastante preciso a la hora de detectar la clase persona, obteniendo ms de un 94% de precisión en ambos conjuntos de datos. Cabe recalcar que el conjunto de datos es muy reducido, para evaluar correctamente el funcionamiento, pero son unos datos muy positivos.

**Recall** Con esta métrica medimos la cantidad que el modelo es capaz de identificar. Usaremos la fórmula indicada en el apartado 4.2:

$$recall - train = \frac{47}{47 + 24} = 0.6619, \text{ para el conjunto de datos de train}$$

$$recall - test = \frac{17}{17 + 6} = 0.7391, \text{ para el conjunto de datos de test}$$

El modelo es capaz de identificar en el peor de los casos, en un 66%, a personas en las imágenes. Por lo que este modelo detecta aproximadamente 3 de cada 5 personas que pasan por el procesado de imagen.

**Accuracy** La exactitud mide el porcentaje de casos que el modelo ha acertado. Usaremos la fórmula indicada en el apartado 4.3:

$$accuracy - train = \frac{16 + 47}{47 + 24 + 16 + 2} = 0.7078, \text{ para el conjunto de datos de train}$$

$$accuracy - test = \frac{17 + 5}{17 + 6 + 5 + 1} = 0.7586, \text{ para el conjunto de datos de test}$$

El modelo en el peor de los casos, acierta un 70% de las veces, pero es una métrica muy engañosa, ya que al ser un estudio entre 2 clases, es muy fácil decantarte por una u otra.

**F-Score** El valor F-Score o también llamado F1 se utiliza para combinar las métricas precisión y recall en un solo valor, comparando así el rendimiento combinado de la precisión y la exhaustividad. Usaremos la fórmula indicada en el apartado 4.4:

$$F1 - train = 2 * \frac{0.9591 * 0.6619}{0.9591 + 0.6619} = 0.7832, \text{ para el conjunto de datos de train}$$

$$F1 - test = 2 * \frac{0.9444 * 0.7391}{0.9444 + 0.7391} = 0.8292, \text{ para el conjunto de datos de test}$$

Esta métrica asume que nos importan por igual tanto la precisión como la exhaustividad.

### 5.8.2. Detección correcta de personas

En esta subsección se mostrarán diferentes imágenes(Figura 5.13) después del procesado de imagen, donde se puede observar que la detección es correcta y muestra únicamente a la persona resaltada.



**Figura 5.13:** Detección de una única persona

Cabe recalcar que las condiciones de iluminación son favorables en todos estos resultados.

### 5.8.3. Detección parcial en una imagen

En esta subsección, observaremos resultados no satisfactorios. Estos se deben a las condiciones del entrenamiento de la CNN. Como se puede observar en las imágenes(Figura 5.14), la CNN después de procesar la imagen, detecta parcialmente a las

personas, o realiza la bounding box englobando a más de una persona dentro de ella.



**Figura 5.14:** Detección parcial de personas

Estos resultados mejorarían con un entrenamiento mucho más robusto, incluyendo un dataset exhaustivo.

#### 5.8.4. Detección incorrecta a causa de condiciones desfavorables

En esta subsección se podrán observar resultados no satisfactorios a causa de condiciones desfavorables a la hora de realizar las fotografías. Como se puede observar en las imágenes(Figura 5.15), la CNN después de procesar la imagen, detecta sombras de personas como si fuera otra persona, esto se debe a causa de la similitud que tiene la sombra con una persona, y la limitación mencionada a la hora del entrenamiento.



**Figura 5.15:** Detección incorrecta a causa de condiciones desfavorables



# 6 Conclusiones y futuros trabajos

## 6.1. Conclusiones

La finalidad del trabajo ha sido montar y realizar la puesta a punto de un dron, con vuelo automático planificado mediante planes de vuelo, para la adquisición de imágenes aéreas y que estas se envíen de forma automática a una unidad terrestre remota, en nuestro caso la Raspberry Pi 4, la cual está ejecutando un servidor web al cual se puede acceder mediante el navegador web de cualquier dispositivo que este conectado a la misma red usando el protocolo HTTP. La aplicación de este trabajo está destinada a la videovigilancia de los terrenos agrícolas.

Una vez la GoPro envía las imágenes al servidor, estas son descargadas y pasan por el procesamiento de imagen realizado con la red neuronal Mask R-CNN, donde se realiza la detección de si hay personas o no. Consiguiendo así la videovigilancia de los campos de cultivo en apenas unos minutos.

Finalmente, se ha demostrado el correcto funcionamiento de las 2 aplicaciones, analizando resultados del procesamiento de imagen, y el correcto funcionamiento del dron a la hora de realizar tanto vuelos manuales como vuelos automatizados.

## 6.2. Futuros trabajos

Después de realizar el proyecto, de toda la información recaptada a la hora de realizarlo, y de la visión de futuro del entorno de los UAV, he llegado a la conclusión de que los drones proporcionaran un gran avance al sector agrícola, permitiendo desde el control del regado, la maduración de los productos conreados, o como en el caso de este proyecto, la videovigilancia de los terrenos. Este avance provocará un aumento de la productividad y la calidad de los productos, y por lo tanto un desarrollo en muchas técnicas que hasta ahora siguen un proceso muy tradicional.

Teniendo en cuenta el potencial que tienen los UAVs, nos damos cuenta de que la industria electrónica necesita tener mas en cuenta estos dispositivos, y desarrollar componentes de sensórica, y cámaras para aplicaciones de aeronaves en cualquier ámbito, para así poder hacer trabajos mucho más precisos y obtener unos mejores resultados. Por otro lado, la industria de los drones, tiene que tener en cuenta el desarrollo de softwares con una interfaz muy sencilla para cualquier tipo de usuario, facilitando así el uso a todo el mundo, y por lo tanto la introducción de estos dispositivos en muchos mercados.

## *Conclusiones y futuros trabajos*

Teniendo en cuenta el potencial del proyecto, han quedado unos puntos claramente a mejorar, sobretodo enfocados en las aplicaciones. En primer lugar, unificar las 2 aplicaciones en una sola, incluyendo el procesado de imagen en la aplicación del servidor, consiguiendo así una optimización del proceso. En segundo lugar, ubicar el servidor en un dominio público, para así no tener que acceder de manera local al servidor, y poder acceder desde cualquier punto a la supervisión y poder controlar más de un campo de conreo al mismo tiempo. El tercer punto y muy importante, es la realización de un entreno a la red neuronal mucho más extenso, con un dataset mucho mas completo, unos parámetros de entrenamiento más extensos, y así conseguir una mejora en la detección, y por lo tanto una mejor funcionalidad.

Respecto al drone y la Raspberry Pi, sería un gran avance la instalación de módulos 4G, para así tener conexión en cualquier momento, sin la necesidad de tener una conexión vía Ethernet, y eliminamos limitaciones como por ejemplo, la extensión de campo que puede supervisar en la actualidad, teniendo en cuenta la batería, ya que instalar un módulo en el drone significa más peso y consumo de la batería, por lo que habría que realizar un estudio de como afectaría la introducción de este módulo al dispositivo.

# Bibliografía

- [1] A. Lewis. How iot is minimising the damage of climate change on the agriculture industry. <https://www.iot-now.com/2019/05/16/95838-iot-minimising-damage-climate-change-agriculture-industry/>, May 2019. Consultat el 16-06-2021.
- [2] C.M. Donald and J. Hamblin. The convergent evolution of annual seed crops in agriculture. volume 36 of *Advances in Agronomy*, pages 97–143. Academic Press, 1983.
- [3] Abhishek Khanna and Sanmeet Kaur. Evolution of internet of things (iot) and its significant impact in the field of precision agriculture. *Computers and Electronics in Agriculture*, 157:218–231, 2019.
- [4] Adrian De Blas Canovaca. Aplicacion para vigilancia de entornos agricolas con uavs y procesamiento de imagen(desarrollo servidor web). Technical report, 2022.
- [5] Matterport. Mask r-cnn for object detection and segmentation. <https://github.com/matterport/MaskRCNN>. Consultado el 17/06/2021.
- [6] Carlos bueno. Tecnologia drone. [www.eleconomista.es/tecnologia/noticias/10302607/01/20/En-2050-los-drones-generaran-mas-de-150000-empleos-segun-la-UE.html](http://www.eleconomista.es/tecnologia/noticias/10302607/01/20/En-2050-los-drones-generaran-mas-de-150000-empleos-segun-la-UE.html), 2020. Consultado el 15/11/2021.
- [7] Eduardo berra villasenor. Control pid para drone. [www.reaxion.utleon.edu.mx/Articulo\\_de\\_variables\\_de\\_control\\_PID\\_para\\_drones\\_cuadcopter.html](http://www.reaxion.utleon.edu.mx/Articulo_de_variables_de_control_PID_para_drones_cuadcopter.html), 2015. Consultado el 15/11/2021.
- [8] Los mejores chasis para drone de carreras y freestyle. <https://chickenfpv.com/chickenfpv/los-mejores-componentes-para-drones-de-carreras-y-freestyle/los-mejores-chasis-para-drones-de-carreras-y-freestyle/>, 2020. Consultado el 13/11/2021.
- [9] Gabriel. Como funcionan las helices de un drone. <https://www.comprardrones.online/como-funcionan-las-helices-de-un-drone/>, 2018. Consultado el 23/11/2021.
- [10] Motor sin escobillas o brushless. <https://guiadrones.com/base-de-conocimiento/motor-sin-escobillas-o-brusless/>, 2019. Consultado el 29/11/2021.
- [11] volandocondrones. Mejores controladores de vuelo. <https://volandocondrones.com/mejores-controladores/>. Consultado el 2/12/2021.
- [12] Fpvmax. Variador electronico. <https://www.fpvmax.com/uncategorized/variador-electronico-esc-funciona/>, 2016. Consultado el 3/12/2021.

- [13] BigdataAILab. ¿Qué es la segmentación semántica, la segmentación de instancias, la segmentación panorámica? <https://my.oschina.net/u/876354/blog/3055850>, 2019. Consultado el 12/12/2021.
- [14] Shilpa Ananth. Faster r-cnn for object detection. <https://towardsdatascience.com/faster-r-cnn-for-object-detection-a-technical-summary-474c5b857b46>, 2019. Consultado el 12/12/2021.
- [15] Elisha Odemakinde. Mask r-cnn: A beginner's guide. <https://viso.ai/deep-learning/mask-r-cnn/>, 2021. Consultado el 12/12/2021.
- [16] Juan Ignacio Bagnato. Breve historia de las redes neuronales artificiales. <https://www.aprendemachinelearning.com/breve-historia-de-las-redes-neuronales-artificiales/>, 2018. Consultado el 20/12/2021.
- [17] Juan Carlos Macho. Lo que hay que saber para elegir los escs para un cuadricóptero. <https://www.prometec.net/esc-para-drones/>. Consultado el 01/12/2021.
- [18] Gregorio Castro. Qué debes conocer de las baterías lipo. <https://dronprofesional.com/blog/que-debes-conocer-de-las-baterias-lipo/>. Consultado el 04/12/2021.
- [19] Vistronica. Controlador de vuelo ardupilot apm 2.8. <https://www.vistronica.com/aeromodelismo/controlador-de-vuelo-ardupilot-apm-2-8-detail.html>. Consultado el 04/12/2021.
- [20] Ardupilot. Archived:apm 2.5 and 2.6 overview. <https://ardupilot.org/copter/docs/common-apm25-and-26-overview.html>. Consultado el 04/12/2021.
- [21] Drones.MX. Mission planner 1.3.16 build 1.1.5456.14788. <https://drones.mx/soporte/mission-planner/:text=Misi> Consultado el 07/12/2021.
- [22] Carlos Aparisi. Flask-server. <https://github.com/carlosaparisi/Flask-server>, June 2021. Consultado el 16/10/2021.
- [23] Jonathan Sanchez. Flask-server. <https://github.com/byjoony/Flask-server.git>, Diciembre 2021. Consultado el 16/12/2021.
- [24] Carlos Aparisi Cantero. Desenvolupament d'un dron customitzat per us en aplicacions d'agricultura. Technical report, UPC, 2021.
- [25] KonradIT. gopro-py-api. <https://github.com/KonradIT/gopro-py-api>. Consultado el 16/12/2021.
- [26] Universitat de Alacant. Modelo vista controlador (mvc). <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>. Consultado el 18/12/2021.
- [27] Bootstrap. <https://getbootstrap.com/>. Consultado el 16/10/2021.
- [28] Lilian Judith Sandoval. Algoritmos de aprendizaje automático para análisis y predicción de datos. *REVISTA TECNOLÓGICA*, (11):36–40, ENERO-DICIEMBRE 2018. Consultado el 27/11/2021.

- [29] B. Walker. *Aprendizaje profundo con Python: Guía completa para principiantes para aprender y comprender los reinos del aprendizaje profundo con Python (Libro En Español/Deep Learning with Python Spanish Book)*. Aprendizaje Profundo con Python Series. Independently Published, 2019.
- [30] Ujjwal Karn. An intuitive explanation of convolutional neural networks. <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>. Consultado el 24/11/2021.
- [31] Juan Barrios. Redes neuronales convolucionales. <https://www.juanbarrios.com/redes-neurales-convolucionales/>. Consultado el 19/11/2021.
- [32] Sumit Saha. A comprehensive guide to convolutional neural networks â the eli5 way. <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>, 2018. Consultado el 11/12/2021.
- [33] Sarang Narkhede. Understanding confusion matrix. <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>. Consultado el 27/11/2021.
- [34] O'Reilly. Evaluating detection (intersection over union). <https://www.oreilly.com/library/view/hands-on-convolutional-neural/9781789130331/a0267a8a-bd4a-452a-9e5a-8b276d7787a0.xhtml>. Consultado el 27/11/2021.
- [35] Jonathan Sanchez. Mask r-cnn. [https://github.com/byjoony/TFG\\_jonathan\\_sanchez.git](https://github.com/byjoony/TFG_jonathan_sanchez.git), 2022.

