



# Centro Universitario de la Defensa en la Escuela Naval Militar

## TRABAJO FIN DE GRADO

*Aplicación de técnicas de aprendizaje profundo a la localización  
y seguimiento de objetos flotantes empleando drones con cámara*

## Grado en Ingeniería Mecánica

**ALUMNO:** José Álvarez San Martín

**DIRECTORES:** José Antonio González Prieto  
José Ignacio Valles Cancela

**CURSO ACADÉMICO:** 2022-2023

Universida<sub>de</sub>Vigo





# Centro Universitario de la Defensa en la Escuela Naval Militar

## TRABAJO FIN DE GRADO

*Aplicación de técnicas de aprendizaje profundo a la localización  
y seguimiento de objetos flotantes empleando drones con cámara*

**Grado en Ingeniería Mecánica**  
Intensificación en Tecnología Naval  
Cuerpo General

UniversidadeVigo



# RESUMEN

Este TFG surge de la necesidad por parte de las marinas, tanto civiles como militares, de disponer de medios prácticos y asequibles que les permitan alcanzar un mayor espectro de visión y comprensión de su entorno marino. Para afrontar esta situación, este trabajo busca desarrollar un modelo de aprendizaje automático que sea integrable a una plataforma aérea de pequeño porte; el fin será, desde una perspectiva aérea, localizar, identificar, y realizar seguimiento de los distintos objetos que se encuentren flotando en la mar. Para la creación de este modelo se aplican técnicas de aprendizaje profundo, trabajando con el algoritmo YOLO, y se le entrena sobre bases de datos creadas para focalizar al sistema en el entorno marítimo, haciendo especial énfasis en su ámbito militar. Así mismo, se realizan actualizaciones de estas bases de datos, según se observan carencias en el modelo, y se reentrena para integrar las nuevas aportaciones. Este sistema será exportado a una plataforma portátil que pueda acoplarse a un dron de pequeño tamaño, que se concluye como la plataforma que mejor cumple con los requerimientos circunstanciales. Una vez desarrollado el modelo, se analizan los resultados de su aplicación y se plantean futuras mejoras y líneas de actuación.

## PALABRAS CLAVE

Aprendizaje profundo, dron, YOLO, *tracking*, etiquetado



# AGRADECIMIENTOS

Ante todo, quiero expresar mi gratitud a mis tutores José Antonio y José Ignacio, que tanto se han implicado en este trabajo, y quienes me han introducido a este interesante mundo del aprendizaje automático.

También quiero darle las gracias a mi familia, por su apoyo incondicional durante todos estos años, por la formación que me han dado, por su paciencia y por su comprensión, que me han permitido llegar hasta donde estoy.

Por último, agradezco a todos mis amigos y compañeros de la promoción 423-153 el haber estado siempre ahí durante estos cinco años. Sin ellos, el paso por la escuela no habría sido igual.



## CONTENIDO

|                                                     |    |
|-----------------------------------------------------|----|
| Contenido .....                                     | 1  |
| Índice de Figuras .....                             | 5  |
| Índice de Tablas.....                               | 7  |
| 1 Introducción y objetivos .....                    | 9  |
| 1.1 Contexto y motivación .....                     | 9  |
| 1.2 Objetivos .....                                 | 11 |
| 1.3 Estructura de la memoria .....                  | 11 |
| 2 Estado del arte .....                             | 13 |
| 2.1 <i>La inteligencia artificial</i> .....         | 13 |
| 2.2 <i>Machine Learning</i> .....                   | 14 |
| 2.2.1 Definición .....                              | 14 |
| 2.2.2 Origen del Machine Learning .....             | 14 |
| 2.2.3 Aprendizaje profundo .....                    | 15 |
| 2.2.4 Redes neuronales artificiales .....           | 16 |
| 2.2.5 Redes convolucionales.....                    | 18 |
| 2.2.6 Visión por computador .....                   | 18 |
| 2.2.7 YOLO .....                                    | 21 |
| 2.3 Vehículos aéreos no tripulados .....            | 24 |
| 2.3.1 Origen de los drones .....                    | 24 |
| 2.3.2 Los drones en la actualidad.....              | 26 |
| 2.3.3 Los drones en el mundo civil.....             | 26 |
| 2.4 Los drones en las instituciones del Estado..... | 27 |
| 2.4.1 Dirección General de Tráfico .....            | 28 |
| 2.4.2 Vigilancia Aduanera .....                     | 29 |
| 2.4.3 Salvamento Marítimo .....                     | 29 |
| 2.4.4 Policía Nacional.....                         | 29 |
| 2.4.5 Guardia Civil .....                           | 30 |
| 2.4.6 Ejércitos e Infantería de Marina .....        | 30 |
| 2.4.7 Cuerpo General de la Armada .....             | 38 |
| 3 Desarrollo del TFG.....                           | 41 |
| 3.1 Medios empleados.....                           | 41 |
| 3.1.1 Anaconda .....                                | 41 |
| 3.1.2 Label Studio.....                             | 42 |

|                                                                                                |    |
|------------------------------------------------------------------------------------------------|----|
| 3.1.3 Ultralytics HUB .....                                                                    | 42 |
| 3.1.4 Google Colab .....                                                                       | 42 |
| 3.1.5 Google Drive .....                                                                       | 42 |
| 3.1.6 Codium .....                                                                             | 43 |
| 3.1.7 Coral USB Accelerator .....                                                              | 43 |
| 3.2 Desarrollo de sistema de identificación de objetos desde una posición fija .....           | 43 |
| 3.2.1 Obtención de datos .....                                                                 | 44 |
| 3.2.2 Etiquetado .....                                                                         | 45 |
| 3.2.3 Entrenamiento .....                                                                      | 46 |
| 3.3 Desarrollo de sistema de identificación de objetos desde una perspectiva móvil aérea ..... | 51 |
| 3.3.1 Obtención de datos .....                                                                 | 51 |
| 3.3.2 Etiquetado .....                                                                         | 53 |
| 3.3.3 Entrenamiento .....                                                                      | 56 |
| 3.3.4 Aplicación del modelo .....                                                              | 57 |
| 3.4 Mejora del modelo .....                                                                    | 58 |
| 3.4.1 Ampliación del dataset .....                                                             | 59 |
| 3.4.2 Actualizaciones del modelo .....                                                         | 60 |
| 3.4.3 Entrenamiento final .....                                                                | 63 |
| 3.5 Integración del sistema a la plataforma Coral .....                                        | 64 |
| 4 Obtención y análisis de resultados .....                                                     | 65 |
| 4.1 Análisis del modelo .....                                                                  | 65 |
| 4.2 Aplicación y análisis de resultados .....                                                  | 69 |
| 4.2.1 Video parada naval .....                                                                 | 69 |
| 4.2.2 Video externo .....                                                                      | 71 |
| 5 Conclusiones y líneas futuras .....                                                          | 75 |
| 5.1 Conclusiones .....                                                                         | 75 |
| 5.2 Líneas futuras .....                                                                       | 76 |
| 6 Bibliografía .....                                                                           | 77 |
| Anexo I: Implicaciones Sociales, Económicas, y Ambientales .....                               | 83 |
| Anexo II: Reflexiones Éticas y Sociales .....                                                  | 85 |
| Anexo III: Catálogo de Siglas .....                                                            | 87 |
| Anexo IV: Script de organización de imágenes .....                                             | 89 |
| Anexo V: Script de extracción de imágenes .....                                                | 91 |
| Anexo VI: Script de entrenamiento del modelo .....                                             | 93 |

|                                              |    |
|----------------------------------------------|----|
| Anexo VII: Script de ejecución YOLOv8n ..... | 95 |
|----------------------------------------------|----|



## ÍNDICE DE FIGURAS

|                                                                                                                  |    |
|------------------------------------------------------------------------------------------------------------------|----|
| Figura 2-1 Integrantes de la conferencia de Dartmouth, 1956 [9] .....                                            | 14 |
| Figura 2-2 El <i>Deep Learning</i> encuadrado en el <i>Machine Learning</i> y la IA [elaboración propia] ...     | 15 |
| Figura 2-3 Esquema de una red neuronal artificial básica [elaboración propia].....                               | 17 |
| Figura 2-4 Esquema de red neuronal artificial [elaboración propia].....                                          | 18 |
| Figura 2-5 Diferencia gráfica entre las tareas de clasificación, localización y segmentación [16]..              | 19 |
| Figura 2-6 Imagen médica analizada por visión por computador [16] .....                                          | 21 |
| Figura 2-7 Proceso de funcionamiento YOLO [20] .....                                                             | 22 |
| Figura 2-8 Línea temporal del desarrollo del algoritmo YOLO [elaboración propia] .....                           | 22 |
| Figura 2-9 Globos austriacos sobre Venecia, 1849 [25] .....                                                      | 24 |
| Figura 2-10 Misil guiado V1 alemán [27].....                                                                     | 25 |
| Figura 2-11 UAV israelí IAI Scout [29].....                                                                      | 25 |
| Figura 2-12 Tipologías de drones [31] .....                                                                      | 26 |
| Figura 2-13 Dron proyecto <i>Delorean</i> de Correos [34].....                                                   | 27 |
| Figura 2-14 Dron <i>Matrice</i> 200 de la DGT [35].....                                                          | 28 |
| Figura 2-15 Dron del CNP [43].....                                                                               | 29 |
| Figura 2-16 Clasificación de los RPAS según estándares OTAN [47].....                                            | 31 |
| Figura 2-17 Operador con un <i>Black Hornet</i> [50] .....                                                       | 32 |
| Figura 2-18 Preparación de <i>Raven</i> para su despliegue [53] .....                                            | 33 |
| Figura 2-19 <i>Tucán</i> sobre su plataforma de despegue [63] .....                                              | 34 |
| Figura 2-20 Dron <i>Fulmar</i> de THALES España [66].....                                                        | 34 |
| Figura 2-21 <i>Atlantic</i> en el GAIL [71].....                                                                 | 35 |
| Figura 2-22 <i>Searcher</i> MK.III [53].....                                                                     | 36 |
| Figura 2-23 <i>Predator</i> B del Ejército del Aire y del Espacio [78] .....                                     | 37 |
| Figura 2-24 Escudo de la 11ª Escuadrilla de Aeronaves [83] .....                                                 | 38 |
| Figura 2-25 <i>Scan Eagle</i> de la Armada [84] .....                                                            | 39 |
| Figura 2-26 <i>Scan Eagle</i> en plataforma de lanzamiento [83] .....                                            | 39 |
| Figura 2-27 M5D <i>Airfox</i> en su lanzadera [87] .....                                                         | 40 |
| Figura 3-1 Imágenes empleadas del puerto de Cangas de Morrazo [91] .....                                         | 44 |
| Figura 3-2 Imágenes etiquetadas de Cangas [elaboración propia].....                                              | 46 |
| Figura 3-3 Distribución de imágenes del <i>dataset</i> en <i>Ultralytics</i> [fuente: <i>Ultralytics</i> ] ..... | 47 |
| Figura 3-4 Puerto de Cangas analizado por el modelo [elaboración propia] .....                                   | 48 |
| Figura 3-5 Métrica del modelo [elaboración propia] .....                                                         | 49 |
| Figura 3-6 Box Loss [elaboración propia] .....                                                                   | 49 |

|                                                                                                   |    |
|---------------------------------------------------------------------------------------------------|----|
| Figura 3-7 Class Loss [elaboración propia] .....                                                  | 50 |
| Figura 3-8 Object Loss [elaboración propia] .....                                                 | 50 |
| Figura 3-9 Diferentes ángulos y perspectivas en un video del <i>dataset</i> [fuente: Armada]..... | 52 |
| Figura 3-10 MK-41 de una F-100 durante el lanzamiento de un ESSM [92] .....                       | 53 |
| Figura 3-11 Pareja de <i>Canisters</i> [93] frente a un lanzador MK-13 [94] .....                 | 54 |
| Figura 3-12 Plano frontal de una F-100 antes y después del etiquetado [elaboración propia] .....  | 55 |
| Figura 3-13 Imagen con etiquetas rotadas [elaboración propia] .....                               | 55 |
| Figura 3-14 Imagen con etiquetas corregidas [elaboración propia] .....                            | 56 |
| Figura 3-15 Aplicación inicial del modelo [elaboración propia] .....                              | 57 |
| Figura 3-16 Portaaeronaves analizado por el modelo [elaboración propia].....                      | 58 |
| Figura 3-17 Fragata analizada por el modelo [elaboración propia] .....                            | 58 |
| Figura 3-18 BAM analizado por el modelo [elaboración propia] .....                                | 60 |
| Figura 3-19 BAM con etiquetas corregidas [elaboración propia] .....                               | 60 |
| Figura 3-20 BAM analizado con el modelo reentrenado [elaboración propia].....                     | 61 |
| Figura 3-21 Diagrama del proceso de desarrollo [elaboración propia].....                          | 62 |
| Figura 3-22 L-61 analizado en distintos estados del entrenamiento [elaboración propia].....       | 62 |
| Figura 3-23 Comandos en <i>Colab</i> [elaboración propia] .....                                   | 63 |
| Figura 3-24 <i>Coral USB Accelerator</i> [90].....                                                | 64 |
| Figura 4-1 <i>Precision-Recall</i> [elaboración propia].....                                      | 66 |
| Figura 4-2 Matriz de confusión [elaboración propia] .....                                         | 67 |
| Figura 4-3 Características de los objetos analizados [elaboración propia] .....                   | 68 |
| Figura 4-4 Elementos identificados durante el entrenamiento [elaboración propia] .....            | 68 |
| Figura 4-5 Pasada del L-61 [elaboración propia].....                                              | 69 |
| Figura 4-6 Pasada de una F-100 [elaboración propia] .....                                         | 70 |
| Figura 4-7 Análisis de pasada con YOLOv8n [elaboración propia] .....                              | 71 |
| Figura 4-8 Primeros instantes análisis RIMPAC_1 [elaboración propia].....                         | 72 |
| Figura 4-9 RIMPAC_1 [elaboración propia] .....                                                    | 72 |
| Figura 4-10 RIMPAC_2 [elaboración propia] .....                                                   | 73 |
| Figura 4-11 Últimos instantes análisis RIMPAC_2 [elaboración propia] .....                        | 73 |

## ÍNDICE DE TABLAS

|                                                                       |    |
|-----------------------------------------------------------------------|----|
| Tabla 2-1 Modelos de YOLOv8 [22] .....                                | 23 |
| Tabla 2-2 Drones de la DGT (tomada de [35]) .....                     | 28 |
| Tabla 2-3 Características de los drones de la Guardia Civil [46]..... | 30 |



# 1 INTRODUCCIÓN Y OBJETIVOS

## 1.1 Contexto y motivación

La capacidad humana de entender el entorno donde uno se desenvuelve, depende de la forma en que lo percibe. Esta capacidad de percibir el entorno se adquiere mediante mecanismos naturales a los seres vivos, y a los que denominamos sentidos. Los sentidos nos permiten conocer el mundo más allá de nuestra imaginación, y de entre los cinco que posee el ser humano, priman por su aportación el sentido del oído y el de la vista [1]. Pero el sentido del oído, si bien es esencial para la completa comprensión de la situación en la que se encuentra una persona, es así mismo una de las herramientas esenciales para la comunicación humana y la formación y comprensión del lenguaje. No obstante, el sonido es de menor utilidad que la vista a la hora de conocer entornos de grandes superficies como puede ser la mar. Es por esto que la vista adquiere una importancia capital en el entorno marino.

El alcance de la visión humana es, no obstante, finito, y sus límites naturales a veces resultan demasiado restrictivos como para que esta capacidad satisfaga por si sola y sin ayudas ciertos cometidos a los que se puede enfrentar una persona. Es por ello que, en la mar, a pesar de garantizar la vista una mayor concienciación del entorno general, y aun siendo su aportación la de mayor peso, se manifiesta la problemática de cómo aumentar la capacidad visual cuando surgen situaciones en la navegación que requieren de una ayuda complementaria. Este Trabajo de Fin de Grado (TFG) se encuadra en el desarrollo de una solución que facilite los distintos cometidos en la navegación que precisen de asistencia visual, obtenida a partir medios que aporten una perspectiva aérea.

A lo largo de la historia, según la navegación ha ido evolucionando, han evolucionado así mismo las formas con las que se afrontaba la limitación visual. A mediados de la era de las grandes exploraciones oceánicas, entre finales del siglo XVI y principios del siglo XVII, se llevó a cabo la invención del catalejo [2], que se empleó en las marinas de todas las naciones europeas del momento y demás navegantes. Antes de ese momento se dependía exclusivamente en la agudeza visual de los vigías, pero con su desarrollo, el catalejo se estableció como el instrumento de recurrencia para la vigilancia en la mar durante un extenso periodo de tiempo. Esta fase histórica de las ayudas visuales a la navegación se prolongó hasta el siglo XIX, cuando esta ayuda fue desplazada a un segundo plano debido a la llegada de los binoculares [3]. A día de hoy se sigue empleando este instrumento en la mar, no obstante, con la llegada a los buques de cámaras de video, infrarrojas, y demás instrumentación actual, los binoculares ya no ocupan el monopolio de las ayudas a la visión en la mar.

Si bien estas herramientas conllevan una mejora significativa a la comprensión del entorno marino en que se encuentra un buque, con el exponencial avance tecnológico del último siglo, en la actualidad los buques pueden llegar a emplear medios aéreos que les permiten ampliar su campo de visión de forma

drástica. Brindando estos una perspectiva aérea, el espectro visual que el buque pasa a tener a su disposición se ve potenciado radicalmente. Los medios aéreos empleados en la mar son múltiples, especialmente los desplegados por buques portaaviones. Desde aeronaves de ala fija de porte considerable, como los aviones E2 *Hawkeye* estadounidenses de envergadura de más de 24 metros, a aeronaves de ala rotatoria, como los SH60 *Seahawk* empleados por la Armada Española, estas herramientas permiten al barco conocer que tiene a su alrededor en un radio de decenas de millas. La gama de medios es amplia y variada, al igual que las capacidades que estos aportan. No obstante, si bien estas aeronaves tradicionales son capaces de alcanzar largas distancias y permanecer en vuelo largos periodos de tiempo, pocas son las embarcaciones civiles capaces de cumplir con los requerimientos necesarios para su despliegue. Ciertamente, estas aeronaves se emplean en el mundo civil, como en la pesca del atún, por ejemplo, donde es común que la flota pesquera lleve integrada en una de sus unidades un helicóptero para ayudar en el avistamiento del cardumen. Sin embargo, a excepción de la marina de guerra y de la científica, donde su empleo es más recurrente, en la marina mercante, en la pesquera y en la deportiva el empleo de aeronaves no suele ser habitual.

Debido a un presupuesto limitado, las embarcaciones de esta índole se ven privadas de facilidades que podrían suponer una gran diferencia en su travesía y quehaceres marineros. Muchas son las vicisitudes a las que un buque puede verse enfrentado durante una navegación. En un ámbito general, un barco puede verse inmerso en una situación de tráfico complicada que requiera de gran atención para evitar contactos. En otros ámbitos más específicos, como por ejemplo durante maniobras de ataque y desataque, desembarcos anfibios, aprovisionamientos en la mar y otros, surgen nuevas problemáticas de diversas índoles.

Empero, en los últimos años, con el auge de los vehículos aéreos no tripulados, los barcos de menor porte se están comenzando a dotar de drones que les aportan esa perspectiva aérea a cuyo acceso no estaban teniendo previamente. Si bien esta integración todavía se está llevando a cabo y es de reciente aplicación, las ventajas que aporta son significativas. No obstante, la capacidad de hacer frente de forma eficaz a los posibles problemas expuestos anteriormente no está tan desarrollada; los sensores de los drones, ya sean cámaras electroópticas, cámaras infrarrojas, radares de apertura sintética, o cualquier otro, aportan información de gran utilidad al controlador del sistema si bien, en muchos casos, resulta compleja de analizar o rebasa la capacidad de atención del operador. El desarrollo de una solución práctica a esta problemática es un tema sobre el que pivota el planteamiento de este TFG.

Por otra parte, en la esfera militar del mundo náutico, además, se presenta otro factor de importancia para este TFG: la recolección de información para su aplicación a la inteligencia naval. Este ámbito se centra principalmente en la situación geográfica, los equipos que portan, y el estado de los buques de otras marinas; datos de relevancia táctica y estratégica para las armadas. Respecto a la situación, ya existen sistemas aéreos específicos y de gran fiabilidad que desempeñan estas tareas de localización mediante procedimientos de búsqueda y reconocimiento de probada suficiencia, como por ejemplo el *Hawkeye* mencionado anteriormente. Para poder identificar los diferentes equipos de un buque militar, ya sea armamento, sensores o demás potenciadores de capacidades, se requieren no obstante aeronaves capaces de captar imágenes de gran detalle y cercanía de las embarcaciones, al igual que un profundo conocimiento de la plataforma por parte del piloto o sus operadores de equipos. Estos factores conllevan que tanto el piloto y su dotación deban ser adiestrados en la identificación de los equipos de relevancia, como que la aeronave tenga que posicionarse a una distancia suficientemente cercana como para suponer un gran riesgo y peligro para su seguridad. Respecto a la segunda problemática, la solución es la misma que la planteada en el párrafo anterior, es decir, el empleo de drones, evitando así poner en peligro vidas humanas. Respecto a la primera, la identificación de equipos, se busca abordar su solución en este trabajo.

La motivación de este TFG se asienta en la búsqueda de una ayuda práctica para los operadores de los drones navalizados. Se persigue aportar una herramienta adicional que les facilite el desempeño de sus cometidos a la hora de apoyar a la navegación y las tareas de sus buques nodriza, haciendo especial

énfasis en la aplicación de sistemas de inteligencia artificial creados para resolver el problema de la identificación de equipos presentes en buques militares.

## 1.2 Objetivos

El objeto principal de este trabajo es el desarrollo de un modelo de aprendizaje automático, empleando técnicas de aprendizaje profundo, que permita la localización y el seguimiento de objetos flotantes mediante el uso de drones con cámara. Este modelo tiene como aspiración última facilitar las labores del piloto de dron a la hora de detectar, localizar, reconocer, identificar y hacer seguimiento de los distintos barcos, focalizándose en embarcaciones de guerra, captados con la cámara de video de su dispositivo aéreo, así como identificar elementos característicos que definen las capacidades del sistema observado por el dron: antenas de comunicaciones satélite, radares de navegación, armamento, u otros equipos de relevancia.

Para cumplir con la finalidad de este TFG, se busca, como objetivos, alcanzar los siguientes hitos:

- La formación en *Machine Learning*, focalizándose en el aprendizaje profundo y la visión por computador.
- La familiarización con la metodología a ejecutar en el desarrollo del modelo. Esto se llevará a cabo mediante un estudio previo de la misma, que finalizará con la construcción de un modelo similar simplificado de menores prestaciones.
- La adquisición y el etiquetado de una base de datos lo suficientemente nutrida como para generar un sistema de aprendizaje profundo que aporte los resultados buscados.
- Entrenamiento del modelo de aprendizaje mediante el algoritmo YOLO sobre la base de datos generada.
- Aplicación del modelo sobre datos no empleados en su entrenamiento.
- Integración del modelo en un dispositivo portátil que permita su instalación en drones de pequeño porte.
- Análisis del funcionamiento y rendimiento del modelo desarrollado.

## 1.3 Estructura de la memoria

Habiendo dispuesto los hitos y objetivos que busca observar este trabajo, y tras su contextualización, se detalla a continuación la estructura en que se organizará la presente memoria. Para aportar una visión clara y ordenada, el trabajo se divide en cinco capítulos diferenciados a los que se añade, a la finalización del último de estos, una bibliografía, que testifica las fuentes empleadas en el desarrollo del trabajo, y una serie de anexos complementarios. Los distintos apartados son los siguientes:

- Capítulo 1: Introducción y objetivos. En este primer capítulo se introduce el contexto del trabajo, aportando las motivaciones y la justificación para el desarrollo del mismo. Además, se establecen los objetivos a alcanzar y se refleja la estructura de la memoria.
- Capítulo 2: Estado del arte. El segundo capítulo de la memoria estudia diferentes aspectos y materias cuyo conocimiento es necesario para el desarrollo práctico del trabajo y su comprensión. También se analiza el estado actual en que se encuentra el sector de la aviación tripulada de forma remota, aspecto de relevancia en este TFG. Así, las subdivisiones de este capítulo tratan los siguientes temas:
  - Breve introducción y estudio de la inteligencia artificial y sus principales modalidades.
  - Análisis del aprendizaje automático, haciendo especial énfasis en el aprendizaje profundo y la visión computarizada.
  - Historia y actualidad de los vehículos aéreos tripulados de forma remota.
  - Presencia de los drones en las instituciones del Estado, realzando la situación y uso de los mismos en los Ejércitos y la Armada.

- Capítulo 3: Desarrollo del TFG. En este tercer apartado se indican las herramientas empleadas para la construcción del modelo de aprendizaje. Así mismo, se refleja el proceso de desarrollo llevado a cabo y los diferentes aspectos contemplados en las distintas fases de su ejecución. A continuación, se detalla el empleo del modelo construido y su integración en un dispositivo portátil.
- Capítulo 4: Obtención y análisis de resultados. En este capítulo, tras el desarrollo del modelo, se analizan los resultados obtenidos. También, se aplica el modelo a nuevos retos y se estudia su comportamiento frente a estas situaciones, reflejándose un juicio sobre su rendimiento y funcionamiento.
- Capítulo 5: Conclusiones y líneas futuras. En el último capítulo se completa el trabajo con una vista general del mismo y una serie de conclusiones. Por último, se ponderan posibles líneas futuras de desarrollo e investigación que contribuyan a la consecución del fin en que se encuadra la temática global de este proyecto.

Para finalizar, la última sección de la memoria contiene una bibliografía, donde se reflejan todas las fuentes consultadas para la ejecución de este TFG, y se adjuntan también los diferentes anexos que tienen por función apoyar el desarrollo del trabajo.

## 2 ESTADO DEL ARTE

En este capítulo se abordará el concepto de inteligencia artificial y sus características generales para dar lugar a una posterior explicación del aprendizaje automático y su desarrollo, desglosándolo y profundizando en el aprendizaje profundo y su implementación mediante las redes neuronales artificiales. Se tratarán las redes convolucionales y se estudiará el algoritmo YOLO sobre el que se centrará el presente TFG. Así mismo, se tratará de explicar en qué se basa la visión por computador y, por otro lado, se llevará a cabo una evaluación del presente estado del arte relativo a la situación de los drones a día de hoy, al igual que su empleo por parte de las instituciones del Estado, en especial en tareas que involucren a la visión computarizada.

### 2.1 La inteligencia artificial

La inteligencia artificial (IA) es una rama de las ciencias computacionales que busca desarrollar algoritmos y sistemas que imiten la inteligencia humana y puedan realizar tareas que normalmente requieren de intervención del hombre, tales como la toma de decisiones, el razonamiento o el aprendizaje. Para ello, la IA se basa en el uso de datos y algoritmos para entrenar modelos y sistemas que puedan aprender a realizar tareas complejas. Estos modelos pueden ser supervisados, en los que se utiliza un conjunto de datos etiquetados para entrenar al modelo, no supervisados, en los que el modelo aprende a partir de los datos sin la intervención humana y modelos de aprendizaje por refuerzo:

- **Aprendizaje supervisado:** En este modelo, la máquina aprende con el ejemplo utilizando un algoritmo supervisado, es decir, un algoritmo disponible con datos de entrada, que representa la pregunta planteada al sistema, y datos de salida, que representa la respuesta que deseamos para la pregunta planteada. Al procesar grandes cantidades de datos de entrada en varias capas, el sistema identifica similitudes y diferencias relevantes hasta que es capaz de predecir estos patrones de clasificación y proporcionar respuestas por sí mismo [4].
- **Aprendizaje no supervisado:** En este método de aprendizaje, solo están disponibles los datos de entrada. El propósito del aprendizaje es identificar las estructuras o patrones presentes en los datos de entrada, permitiéndonos así estudiar el comportamiento del sistema con respecto a los datos específicos que serán introducidos. Un sistema típico en este campo es un algoritmo de agrupación, que determina el número de categorías en las que se puede organizar la información de entrada y a qué categoría pertenecen los nuevos datos [5].
- **Aprendizaje por refuerzo:** esta tipología de aprendizaje, a diferencia de las anteriores, se basa en llevar a cabo una serie de acciones, en un entorno determinado, sobre las que se recibe una retroalimentación en forma de recompensa en caso de ser acciones acertadas. El objetivo del aprendizaje por refuerzo es maximizar la recompensa total recibida por el sistema a lo largo del tiempo. Mediante este sistema, no solo se busca aprender a identificar

patrones en los datos de entrada, sino también a tomar decisiones óptimas en función de la retroalimentación recibida del entorno. Un ejemplo de aplicación de este tipo de aprendizaje es el desarrollo de sistemas de inteligencia artificial para juegos, donde el sistema aprende a tomar decisiones óptimas para ganar el juego [6].

El empleo de esta tecnología se expande en ámbitos comprendidos desde la robótica hasta la seguridad nacional pasando la medicina o las finanzas, transformando la forma en que las empresas operan buscando mejorar la eficiencia y la productividad de sus productos y servicios.

## 2.2 Machine Learning

### 2.2.1 Definición

El *Machine Learning* o aprendizaje automático es una rama de la inteligencia artificial que se enfoca en el desarrollo de técnicas que permiten a las máquinas aprender y mejorar con la experiencia a partir de datos. Es por ello que algunos autores denominan a esta área como *Data-Driven Model*, es decir, generación de modelos a partir de datos. Su objetivo es la creación de un sistema que sea capaz de generar un conocimiento que lo capacite para poder acometer tareas para las cuales no estaba inicialmente programado, es decir, que aprenda a hacer algo que inicialmente no sabe hacer.

Para ello las máquinas emplean algoritmos para el procesamiento de grandes cantidades de datos o *big data*, a partir de los cuales modifican sus propiedades. La aplicación de estas técnicas permite identificar patrones que surgen de la información obtenida y a partir de estos patrones generar modelos que permiten elaborar predicciones partiendo de ellos [7].

### 2.2.2 Origen del Machine Learning

El origen del *Machine Learning* se remonta a la década de 1950, cuando se iniciaron los primeros estudios e investigaciones sobre la inteligencia artificial. Es en 1956 cuando se celebra la primera conferencia donde se discute el campo de la inteligencia artificial tal y como lo conocemos a día de hoy, estableciendo, acotando y definiendo este término: “Este estudio procederá sobre la base de que todos los aspectos del aprendizaje o de rasgos de la inteligencia pueden, en principio, ser descritos de una forma tan precisa que se puede crear una máquina que los simule” [8]. En esta conferencia que se llevó a cabo en la Universidad de Dartmouth en Nuevo Hampshire, se aborda así mismo en primera instancia el aprendizaje automático. En la Figura 2-1 se puede ver a 7 de los 11 participantes en la conferencia. El grupo estaba conformado por *Marvin Minsky*, *John Holland*, *Julian Bigelow*, *Ray Solomonoff*, *John McCarthy*, *Claude Shannon*, *Nathanial Rochester*, *Oliver Selfridge*, *Allen Newel*, *Herbert Simon* y el profesor *D.M. Mackay*. A lo largo de los siguientes años y durante la década de los 60 se desarrollan varios algoritmos de aprendizaje supervisado, incluyendo y destacando el algoritmo de regresión lineal y el algoritmo de árboles de decisión.



Figura 2-1 Integrantes de la conferencia de Dartmouth, 1956 [9]

En los años 80 y a lo largo de los 90, el aprendizaje automático experimentó un auge en el interés y la investigación, lo que conllevó el desarrollo de nuevos algoritmos y técnicas de aprendizaje supervisado, no supervisado y por refuerzo. El auge del *big data* a finales de siglo y la cada vez mayor disponibilidad de ingentes conjuntos de datos ha impulsado el crecimiento y la popularidad del aprendizaje automático. Las empresas y organizaciones han comenzado a utilizar el *Machine Learning* para mejorar sus procesos, aumentar su eficiencia y tomar decisiones basadas en datos.

El aprendizaje automático también ha tenido un impacto significativo en la investigación y el desarrollo en áreas como la robótica, la visión por computadora, la biología, la farmacología, el procesamiento del lenguaje natural y muchas otras, pudiendo aplicarse esta modalidad tecnológica a casi cualquier campo de trabajo y despertando aún más el interés en su consecuente desarrollo y mejora. No obstante, a día de hoy y como consecuencia de su mayor presencia en multitud de ámbitos, también surgen nuevas preocupaciones acerca la privacidad y la seguridad de los datos puestos a disposición de estas tecnologías, factores sobre los que se están estudiando políticas de subsanación.

Con el aumento de la disponibilidad de datos y la mejora de la tecnología, se espera una mayor adopción y aplicación del aprendizaje automático en el futuro.

### 2.2.3 Aprendizaje profundo

El aprendizaje profundo es una modalidad integrada en el aprendizaje automático, basada en el empleo de redes neuronales artificiales que se construyen mediante el apilamiento secuencial de múltiples capas intermedias, las cuales se encargan de procesar la salida de la capa anterior para aumentar el nivel de abstracción de la información que representan, de forma que en la capa final se desarrolle una representación numérica a la respuesta que se desea obtener. A cada capa de esta secuencia se la conoce como extractor de propiedades (*feature extractor*), ya que el tipo de patrones de información localizados en las distintas capas comparten niveles de complejidad, y esta aumenta a medida que se avanza en la estructura. Esta modalidad se enfoca en el desarrollo de algoritmos de inteligencia artificial que imitan la estructura y función de la corteza cerebral. Es un enfoque innovador que ha revolucionado la forma en que las máquinas aprenden y han mejorado la precisión y la efectividad en una amplia variedad de tareas, incluyendo la visión por computadora, el procesamiento del lenguaje natural y la robótica.

Al estar el aprendizaje profundo, o *Deep Learning*, dentro del paraguas del *Machine Learning*, como se representa en la Figura 2-2, este aprendizaje cuenta con una vasta cantidad de características que en nada se diferencian del aprendizaje automático clásico. No obstante, si las propiedades fuesen idénticas no cabría lugar a un subcampo como es el *Deep Learning*. Por ende, el aprendizaje profundo se ha de separar ligeramente de su rama troncal a partir de las estructuras de aprendizaje propuestas.

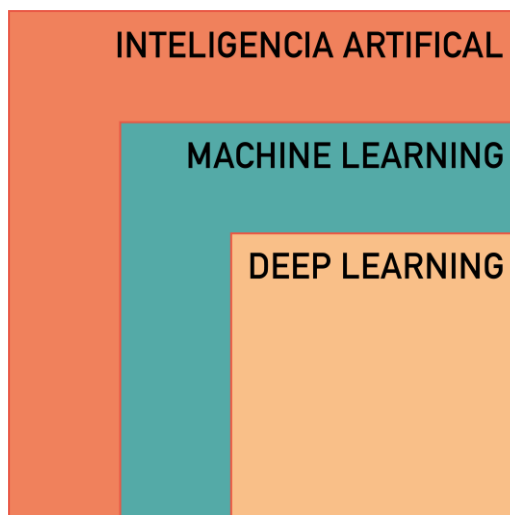


Figura 2-2 El *Deep Learning* encuadrado en el *Machine Learning* y la IA [elaboración propia]

La principal diferencia entre el *Machine Learning* y el *Deep Learning* es que, este último, emplea un gran número de capas consecuentes, mientras que el primero engloba otras tipologías estructurales diferentes. Un modelo puede ser *wide* (ancho) o *deep* (profundo), y lo que determina eso son las capas. Cuantos más elementos tenga una capa, más ancho será el sistema. Por otro lado, cuantas más capas tenga, más profundo será, creando así un modelo de *Deep Learning*.

Las distintas capas y sus elementos se estudiarán en el siguiente apartado y, como se aclarará más adelante, su principal característica funcional es la de conseguir un gran nivel de abstracción [10].

El aprendizaje profundo se ha utilizado con éxito en una amplia variedad de tareas, incluyendo la identificación de objetos en imágenes, en cuya aplicación se centrará el presente TFG, así como la traducción automática de idiomas, la detección de fraudes o la recomendación de productos entre otras muchas aplicaciones. Además, el aprendizaje profundo también ha mejorado la capacidad de las máquinas para comprender y generar texto, música y video, lo que ha llevado a nuevos avances en el campo de la creatividad artificial.

Para poder proporcionar esta ventaja que ha significado tantos avances en el campo de la automatización, el *Deep Learning* se basa, como se ha mencionado anteriormente, en el empleo de redes neuronales artificiales y la idea de que estas puedan aprender a realizar tareas complejas a través de la formación de una jerarquía de características y representaciones [11]. A continuación, se estudiarán estas redes.

#### 2.2.4 Redes neuronales artificiales

Como se ha introducido en el apartado anterior, encuadrándolas dentro del aprendizaje profundo tenemos las redes neuronales artificiales o *Artificial Neural Networks* (ANN). Estas se pueden entender como la unidad funcional o básica del *Deep Learning* [12].

Para cumplir con su cometido, las redes neuronales se disponen de forma que se establecen diferentes capas, a través de las cuales, pasa el flujo de información, de forma que cada capa añade un nuevo nivel de abstracción de la información. Primero en forma de los datos iniciales, que se van empleando y modificando capa tras capa, y al final al pasar la última capa salen en forma de la respuesta buscada.

Estas redes persiguen simular el complejo entretejido neuronal humano, y están compuestas, al igual que este, de pequeñas unidades individuales denominadas neuronas, en las cuales se llevan a cabo las diferentes operaciones. Cada capa por la que pasa la información cuenta con un número de neuronas, desde una capa de una sola neurona a una capa con una cantidad ingente de ellas, proporcional a la complejidad del sistema que se busca, al igual que el número de capas.

Siguiendo esta estructuración, las capas se pueden clasificar de forma general en dos grupos: las capas visibles, que son aquellas en los extremos de la red y que resultan accesibles directamente, y las ocultas, en las que opera la máquina produciéndose el nuevo conocimiento y que no son observables de forma directa.

De manera más específica, las capas se clasifican en tres diferentes tipos: de entrada, ocultas y de salida. En la Figura 2-3 se pueden diferenciar las distintas capas y su posicionamiento en lo que sería una estructura básica de red neuronal.

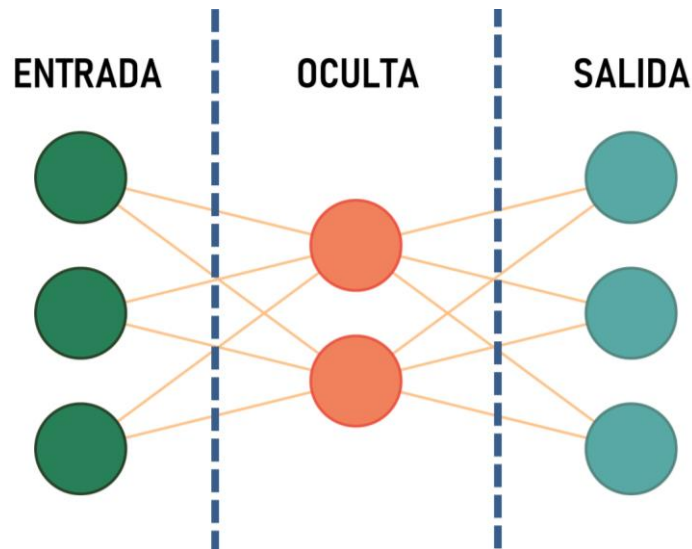


Figura 2-3 Esquema de una red neuronal artificial básica [elaboración propia]

- **Capa de entrada**, o *input layer*, es la primera capa del sistema. Es aquí donde se introducen los datos provenientes de una fuente externa. En esta fase del sistema, de características singulares, se dispondrá de un número de neuronas igual al número de variables que se tendrán en los datos a procesar y las cuales se limitarán a transferir esta información a la siguiente capa, sin llevar a cabo ninguna operación computacional.
- **Capas ocultas**, o *hidden layers*, son las capas intermedias que se encuentran al sobrepasar la capa de entrada. Es en estas capas donde se lleva a cabo todo el procesamiento de los datos mediante las operaciones computacionales que se hayan preestablecido. Como se puede ver en la Figura 2-4, puede haber multitud de capas ocultas interconectadas entre sí, que se dedican a buscar características diferentes en los datos de entrada. Como ejemplo, se puede dar una situación de procesamiento de imágenes por medio de redes neuronales en la que las primeras capas ocultas se ven encargadas de las tareas más generales del proceso, como identificar bordes o límites entre objetos, mientras que las capas finales se ven envueltas en tareas más específicas y de mayor complejidad como identificar a una persona o a un barco.
- **Capa de salida**, u *output layer*, es la última fase del sistema. Al llegar a este punto, basándose en lo obtenido de las capas anteriores se lleva a cabo una predicción final a partir de lo que ha aprendido y se aporta un resultado. La capa de salida es esencial para el buen funcionamiento de una red neuronal y es donde se combinan y procesan todas las señales que se originan en las capas ocultas. Las redes neuronales pueden tener diferentes tipos de capas de salida dependiendo de la tarea que se esté realizando. Por ejemplo, en un problema de clasificación, la capa de salida puede tener una o varias neuronas que utilicen una función de activación para producir una probabilidad de pertenencia a una determinada clase. En otras aplicaciones, como la generación de imágenes o texto, la capa de salida puede tener una forma diferente, como una matriz de píxeles o un vector de caracteres. Es importante destacar que la capa de salida es el resultado final de un proceso de entrenamiento, en el que se ajustan los pesos de las conexiones entre las neuronas de diferentes capas. Durante el entrenamiento, los pesos se actualizan de manera iterativa para minimizar el error en las predicciones y mejorar el desempeño de la red neuronal.

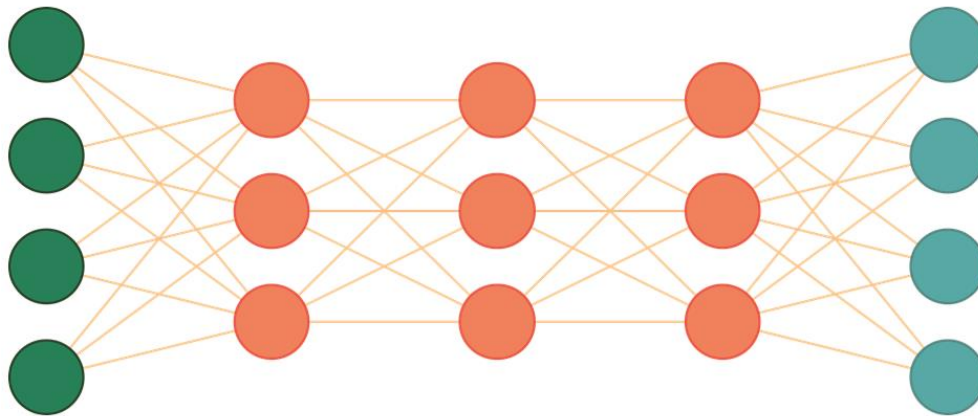


**Figura 2-4 Esquema de red neuronal artificial [elaboración propia]**

A medida que se aumente el número de capas en la red, se aumentará así mismo la capacidad de abstracción de esta, consiguiendo alcanzar a aprender conceptos más complejos [13].

### *2.2.5 Redes convolucionales*

Las redes neuronales se han desarrollado de multitud de maneras y arquitecturas en función de los tipos de problemas. De estas redes artificiales surgen las redes neuronales convolucionales (CNN, por sus siglas en inglés), las cuales se emplean principalmente para el análisis de imágenes y video, estando diseñadas para replicar la forma en que el cerebro humano procesa la información visual [14].

Una CNN consta de una serie de capas de neuronas que procesan la información visual en una forma que representa cada vez más características detalladas de la imagen. Las primeras capas de la red procesan la información en forma de pequeños fragmentos o "filtros". Cada filtro detecta características específicas en la imagen, como líneas, curvas, texturas y demás matices de relevancia. Las capas subsecuentes combinan estos filtros para formar representaciones más complejas y detalladas de la imagen. Un ejemplo podría ser la combinación de varios filtros de líneas y curvas llegando a formar una representación de un objeto específico, como una cara humana.

Además de las capas de procesamiento, las CNN también incluyen una capa de *pooling* o reducción de dimensiones, que reduce la resolución de la imagen sin perder la información relevante. Esto ayuda a la red a concentrarse en las características más importantes y significativas de la imagen. La fase final de una red convolucional es la capa de salida, que utiliza la información procesada para realizar una tarea específica intencionada, como la clasificación de imágenes o la detección de objetos [15].

Las CNN son muy efectivas para tareas de procesamiento de imágenes, como la clasificación de imágenes, la detección de objetos, la segmentación de imágenes y la generación de imágenes. Esto se debe en parte a su capacidad para aprender características y patrones en las imágenes de forma autónoma, sin la necesidad de una supervisión directa.

### *2.2.6 Visión por computador*

La visión computarizada o por computador es un campo de la inteligencia artificial que se enfoca en desarrollar sistemas informáticos que puedan "ver" y comprender el mundo visual en una manera similar a la forma en la que lo hacen los humanos. Mediante los recientes avances en el aprendizaje profundo tratado anteriormente y su constante desarrollo y mejora, la visión por computador se ha logrado aplicar de forma práctica combinando técnicas de procesamiento de imágenes, aprendizaje automático y algoritmos de detección de patrones consiguiendo que las máquinas puedan analizar, comprender y responder al procesamiento de imágenes y vídeos en tiempo real. En resumen, se trata de enseñar a las máquinas a ver e interpretar un entorno visual.

La visión por computador basa su funcionamiento en tres etapas globales: adquisición de imágenes, procesamiento de imágenes y comprensión de imágenes [16]. La primera fase es la de mayor simplicidad tecnológica de las tres, ya que basta con la obtención de fotografías o videos, bien sea mediante una cámara, un escáner o una imagen ya existente en el sistema, mientras que las fases restantes conforman un mayor reto computacional. Para procesar una imagen partimos de que está compuesta por una serie de píxeles que representan la unidad básica de información visual. Cada pixel de una imagen tiene asignados tres valores distintos que representarán la intensidad de cada uno de los tres colores primarios en la tonalidad de su composición, siendo estos colores el rojo el verde y el azul. De esta forma cada imagen pasa a ser una matriz tridimensional (ancho, alto y 3 canales de colores), y el sistema pasará a estudiar la matriz para comprender la imagen y aprender a extraer las respuestas deseadas. Los algoritmos de visión computarizada más básicos se fundamentan en el empleo de álgebra lineal o la probabilidad, pero a medida que aumenta la complejidad de la matriz, y se busca también una mayor capacidad de diferenciación de los diferentes elementos en la imagen, se comienzan a emplear operaciones más complejas como las convoluciones, desarrolladas en el apartado anterior.

Mediante estas metodologías, la visión computarizada se emplea en multitud de problemas que aumentan de forma constante. Además, debido a la perseverante evolución de esta tecnología, surgen nuevos desafíos que afrontar de forma asidua. Entre las tareas de las que se ocupa este campo de investigación, se encuentran las siguientes [16]:

- **Segmentación.** La segmentación implica la separación de una imagen en diferentes segmentos o regiones. Su objetivo es separar los objetos de interés del fondo o de otros objetos en la imagen. Por ejemplo, en una imagen de un automóvil en la calle, la segmentación ayudaría a separar el automóvil del asfalto, los edificios y otros objetos en la imagen.
- **Clasificación.** El objetivo de la clasificación es asignar una categoría a una imagen o video. Esto se hace mediante el entrenamiento del sistema para que pueda distinguir entre diferentes categorías de objetos, personas, animales o cualquier elemento que se pretenda clasificar. Por ejemplo, en un conjunto de imágenes de coches y motos, el algoritmo debe poder etiquetar correctamente cada imagen como "coche" o "moto".
- **Localización.** La localización busca detectar, identificar y ubicar elementos específicos en una imagen o video. Esto significa que no solo se debe detectar la presencia de un objeto, sino también identificar su naturaleza y ubicarlo exactamente. Un ejemplo aplicable al mundo militar de esta tarea es la localización de objetivos de artillería. La diferencia entre esta modalidad y la clasificación y segmentación de imágenes se puede ver en la Figura 2-5.



Figura 2-5 Diferencia gráfica entre las tareas de clasificación, localización y segmentación [16]

- **Keypoint detection.** Este problema se centra en la detección de puntos clave en las imágenes analizadas. Implica la identificación, en una fotografía o un video, de puntos específicos como las esquinas de un objeto, sus bordes o las articulaciones del cuerpo humano. Esto puede ser útil para la identificación de características específicas de un objeto, como el contorno de una cara. Su empleo se puede dar en una amplia variedad de campos, como el reconocimiento facial, la detección de objetos en imágenes médicas o la robótica.
- **Reconocimiento facial.** Esta aplicación busca identificar y verificar la identidad de una persona a partir de una imagen o una secuencia de imágenes en las que aparezca. Se emplea principalmente en aplicaciones de seguridad, como la identificación de sospechosos en las cámaras de vigilancia o el reconocimiento personal para acceder a dispositivos o elementos privados, como puede ser un teléfono móvil.
- **Detección de gestos.** La detección de gestos implica la identificación de patrones específicos de movimiento en una imagen o video. Esto puede ser útil para identificar y clasificar gestos de la mano, como la acción de hacer un puño o la señal de "ok". Es muy útil en una amplia variedad de aplicaciones, como la interacción humano-computadora en el desarrollo de la inteligencia artificial aplicada al ámbito de la robótica, o también en la realidad virtual.
- **Detección de objetos en tiempo real.** La detección de objetos en tiempo real conlleva aplicar la tarea de localización a un ámbito de tiempo real. Esto implica detectar y localizar objetos en video en lugar de simplemente identificarlos en una imagen fija. Es útil en aplicaciones de robótica y de vehículos autónomos, así como en la detección de objetos en videos de vigilancia. Este TFG busca desarrollar un modelo que pueda aplicar esta capacidad.
- **Tracking.** El *tracking* o seguimiento de objetos se refiere a la tarea de seguir la ubicación y el movimiento de un objeto específico en un video a lo largo del tiempo. Es decir, mantener identificado al mismo elemento durante la totalidad de su presencia en el video analizado, sin confundirlo con otro elemento similar ni aun cuando se ve ocluido durante un determinado momento. Esto sirve para analizar el comportamiento de objetos en movimiento o para realizar seguimiento de objetos en movimiento en tiempo real.

Existen multitud de aplicaciones más, como la reconstrucción de imágenes de baja calidad en resoluciones de mayor detalle o la transformación de una imagen en 2D a una escena en 3D, pasando por el reconocimiento de escenas en el que el modelo de visión es capaz de identificar si en una imagen se retrata una ciudad, un valle o una playa, y otras tareas más variadas.

Los sistemas de visión por computador se utilizan en una amplia gama de entornos, desde la seguridad y la vigilancia hasta la robótica y la medicina. En seguridad y vigilancia, los sistemas de visión por computadora pueden analizar imágenes y vídeos en tiempo real para detectar eventos o comportamientos sospechosos. En robótica, la visión por computadora permite a los robots "ver" y comprender su entorno, lo que les permite realizar tareas complejas de manera autónoma. En medicina, como se observa en la Figura 2-6, los sistemas de visión por computador se utilizan para analizar imágenes médicas, como radiografías y tomografías, para detectar enfermedades o lesiones y también para analizar grandes conjuntos de datos médicos y encontrar patrones y tendencias que puedan ayudar a mejorar la atención médica. También se utiliza en aplicaciones comerciales, como la detección de objetos en imágenes publicitarias o la recomendación de productos en línea basada en la detección de objetos en imágenes.

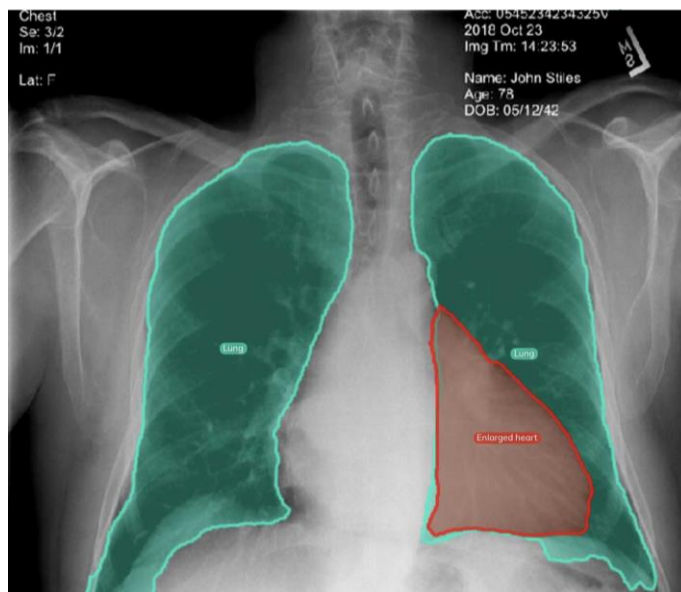


Figura 2-6 Imagen médica analizada por visión por computador [16]

Sin embargo, a pesar de los muchos avances en la visión por computador, aún existen desafíos importantes que deben abordarse. Uno de los mayores desafíos es la variabilidad en las imágenes y vídeos, que pueden ser afectados por factores como la iluminación, la perspectiva, la distorsión y la oclusión de objetos. Además, también existe el desafío de garantizar la privacidad y la seguridad de los datos, especialmente cuando se utilizan sistemas de visión por computadora en aplicaciones de vigilancia o seguridad.

### 2.2.7 YOLO

YOLO (*You Only Look Once*) es un algoritmo de aprendizaje automático de detección de objetos en imágenes y vídeos. Se diferencia de la mayoría de algoritmos de detección de objetos en que solo hace una "mirada" a la imagen para detectar y clasificar objetos, en lugar de hacer una exploración exhaustiva previa, del conjunto de imágenes y etiquetas empleadas en el entrenamiento, para obtener una lista de cajas de marcado de objetos candidatas para ser empleadas como respuesta [17]. Este algoritmo ha supuesto un importante avance en la visión por computadora y es conocido por su velocidad y precisión en la detección de objetos.

Dentro de la detección de objetos en la visión por computador existen, como se ha mencionado, dos tipos de algoritmos. El primero, en el que se encuadra YOLO, se denomina *Single Shot Object Detectors* y funciona realizando un único procesamiento de la imagen y, a continuación, efectuando una predicción de los objetos que hay en ella. Esta metodología genera una solución práctica para las aplicaciones en tiempo real de modelos de detección de objetos, ya que permite un análisis rápido de las imágenes. No obstante, este método es por norma general menos preciso, especialmente en la detección de objetos de pequeño tamaño, debido a la rapidez de su procesamiento. Estas características se contraponen con las del segundo tipo de algoritmos: los *Two Shot Object Detectors*. Estos realizan dos procesamientos de las imágenes a analizar para identificar los objetos en ellas. Tras el primer procesamiento, el algoritmo genera una serie de propuestas para la identidad y localización de posibles objetos. Esto se complementa cuando, tras volver a procesar la imagen, se asientan o se descartan estas propuestas, siendo las reafirmadas las predicciones finales generadas por el sistema. Mediante este procedimiento se alcanzan cotas de precisión mayores que con modelos de una sola pasada, si bien se requiere una capacidad computacional mayor y el tiempo de análisis es también significativamente superior [18]. Por ello, la aplicación de los sistemas de dos pasadas es más idónea para situaciones en que se requiera una certeza de identificación superior, frente a los de una pasada que son idóneos para empleo en equipos de análisis en tiempo real, como será el caso desarrollado en este TFG.

El algoritmo YOLO emplea redes convolucionales para la detección de objetos. Este busca la delimitación de los diferentes objetos y su clasificación en distintos tipos. Para ello, basa su funcionamiento en la división de la imagen en una cuadrícula conformada por celdas de igual tamaño. Cada celda genera, junto a otras, conjuntos de cajas delimitadoras o *bounding boxes* con formas rectangulares que serán las que contengan a los objetos detectados en la imagen. Para determinar que cajas contienen o no a un objeto en la imagen, se aplica la red convolucional que extrae las características de la imagen, tales como la probabilidad que tiene cada una de las múltiples cajas generadas de que haya un objeto dentro suyo. Según se detectan superposiciones en las cajas, se aplica el algoritmo denominado *Non Maximum Suppression*, un método que elimina una a una las cajas que coinciden con otras y cuya puntuación es menor que estas. Así, se van eliminando las cajas que menos probabilidades tienen de contener objetos, o que los contienen, pero con peor ajuste, hasta que al final solo quedan las *bounding boxes* que mejor delimitan los objetos que deseamos localizar en la imagen, como se ve en la Figura 2-7. Por último, tras identificar los objetos, el algoritmo clasifica el tipo de objeto al que pertenece cada *bounding box* que las capas anteriores han generado como predicción de los objetos localizados en la imagen, siendo esta tarea ejecutada en la última capa de la red convolucional [19]. Esta red, por tanto, genera como respuesta a cada objeto localizado (la red, además, debe decidir el número de objetos localizados mediante el algoritmo *Non-Maximum Supression* comentado anteriormente) un vector de 5 valores: las coordenadas x e y del centro de la caja, su ancho y su altura, y el tipo de objeto. En función de la versión del algoritmo, algunas de las cuales se explicarán brevemente a continuación, se puede aportar un sexto dato: la probabilidad de acierto por cada objeto detectado y clasificado.

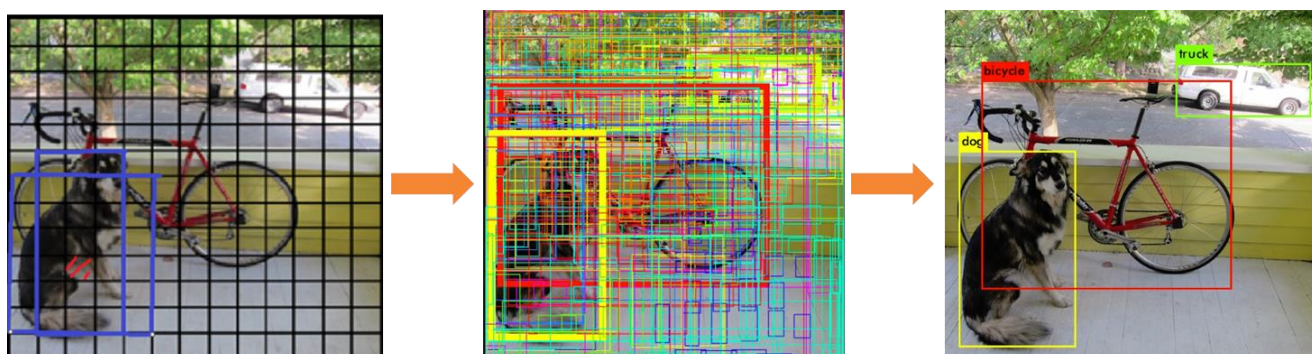


Figura 2-7 Proceso de funcionamiento YOLO [20]

Desde su primera introducción en 2015, este algoritmo ha estado en constante estudio y desarrollo [21]. Se han creado multitud de nuevas versiones en las que cada una ha aportado una mejora o característica diferenciadora, pero basándose todas en el mismo método de funcionamiento. Como se puede ver en la Figura 2-8, el desarrollo es reciente y constante, habiéndose introducido la última versión a día de redacción de este trabajo, YOLOv8, en enero de este mismo año. A continuación, se hablará de la quinta y la octava versión del algoritmo, ya que serán las empleadas en este TFG.

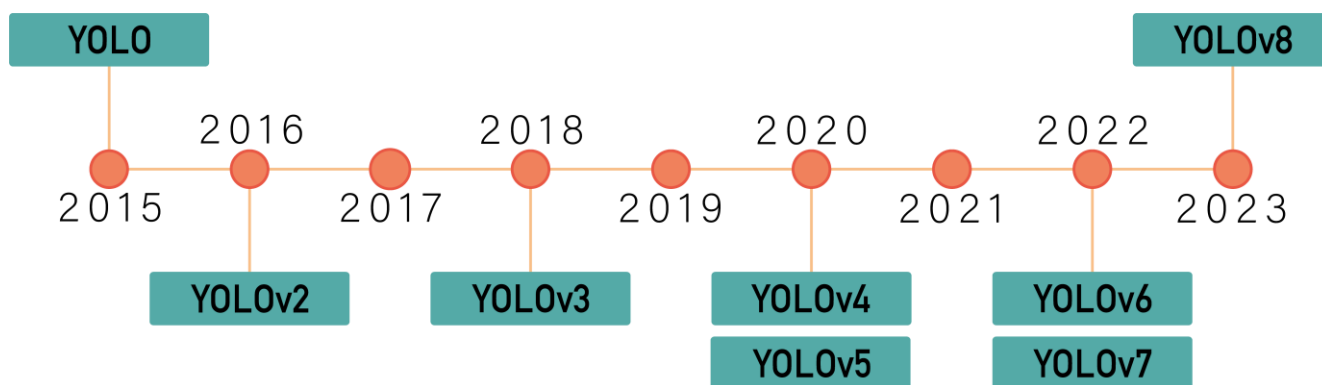


Figura 2-8 Línea temporal del desarrollo del algoritmo YOLO [elaboración propia]

- **YOLOv5.** Partiendo de las mejoras en detección y rapidez desarrolladas en las versiones anteriores, la versión 5 de YOLO implementa además una arquitectura diferente y más compleja, denominada *EfficientDet*, que genera una precisión mayor y un tiempo de ejecución significativamente menor que YOLOv4 y YOLOv3. También mejora la facilidad de uso [18]. Esta versión, al igual que YOLOv8 sí que aporta la fiabilidad de clasificación de objetos planteada anteriormente.
- **YOLOv8.** La versión más actual del algoritmo. Uno de sus aspectos de mayor relevancia para este TFG es la capacidad integrada de realizar seguimiento de objetos en vídeos, que es el objetivo básico del desarrollo de este TFG. Mediante la detección en cada fotograma de los distintos objetos, las demás versiones podían emplearse para, con el apoyo de sistemas auxiliares, realizar su seguimiento. Esta función la integra directamente YOLOv8 mediante dos algoritmos que pueden ser seleccionados: *BoT-SORT* y *ByteTrack*. Así mismo, hay una mejora exponencial respecto al rendimiento y precisión de detección. Esto se ha conseguido con el aumento de la profundidad y la complejidad de la red, así como con la integración de otras técnicas de visión computacional como la segmentación semántica. No obstante, en la fecha en que se redacta este trabajo, aún están pendientes de publicación los artículos científicos que defiendan y desarrollen esta versión, si bien su empleo ya se está ejecutando [22].

Independientemente de la versión del algoritmo, estas se dividen, así mismo, en función de su número de capas y complejidad computacional. Así, las distintas versiones pueden ser *nano*, *small*, *medium*, *large*, y *extra large*; no obstante, no todas las versiones contemplan todas las variantes. Con cada variante, aumenta el número de parámetros y mejora la precisión de detección. Sin embargo, también aumenta la lentitud de procesado y su peso computacional [23]. Así, los algoritmos empleados a lo largo del desarrollo de este TFG serán YOLOv5s (*small*), YOLOv8n (*nano*) y YOLOv8s (*small*). En la Tabla 2-1, se pueden ver los distintos modelos desarrollados para la última versión del algoritmo YOLO. Los valores representados, a excepción del tamaño, que hace referencia al máximo que puede tener una imagen para ser procesada por el algoritmo, indican diferentes aspectos computacionales. Como se puede observar, a medida que se pasa de un modelo al siguiente, aumenta el valor de los distintos parámetros. Así, la velocidad de computación se incrementa, pero también lo hace el número de parámetros que emplea el modelo, resultando esto en un aumento considerable de la carga computacional. El mAP es un valor que hace referencia a la precisión del modelo. Se estudiará con mayor detalle más adelante, en el apartado 3.2.3.

| Modelo  | Tamaño<br>(píxeles) | mAP<br>0,5:0,95 | Velocidad<br>CPU ONNX<br>(ms) | Velocidad<br>A100 TensorRT<br>(ms) | Parámetros<br>(Millones) |
|---------|---------------------|-----------------|-------------------------------|------------------------------------|--------------------------|
| YOLOv8n | 640                 | 37,3            | 80,4                          | 0,99                               | 3,2                      |
| YOLOv8s | 640                 | 44,9            | 128,4                         | 1,20                               | 11,2                     |
| YOLOv8m | 640                 | 50,2            | 234,7                         | 1,83                               | 25,9                     |
| YOLOv8l | 640                 | 52,9            | 375,2                         | 2,39                               | 43,7                     |
| YOLOv8x | 640                 | 53,9            | 479,1                         | 3,53                               | 68,2                     |

Tabla 2-1 Modelos de YOLOv8 [22]

## 2.3 Vehículos aéreos no tripulados

Un vehículo aéreo no tripulado (VANT), también referido como UAV por sus siglas en inglés y más comúnmente denominado como dron, palabra registrada en 2014 en la 23ª edición del Diccionario de la lengua española de la Real Academia Española, es una aeronave capacitada para operar en ausencia de un piloto que se encuentre físicamente en la misma. Cuando se hace referencia tanto a la aeronave como a su equipo de control, también se denomina a estos ingenios como RPAS, abreviatura de la terminología inglesa *Remotely Piloted Air System*.

### 2.3.1 Origen de los drones

El comienzo de los drones se justifica en la búsqueda de nuevas formas de hacer la guerra. En julio del año 1849 se ideó por vez primera, que se tenga registrado, el empleo de medios aéreos no tripulados. La ciudad de Venecia se encontraba sitiada cuando el ejército austriaco empleó con el fin de obtener su capitulación una flotilla de 200 globos aerostáticos incendiarios sin ningún tipo de dotación a bordo [24]. La operación fue un fracaso ya que debido a las corrientes de viento apenas una sola bomba cayó sobre la ciudad, pero esta acción sirvió para constituir el primer bombardeo aéreo de la historia y la primera concepción de un vehículo aéreo no tripulado. En la Figura 2-9 se puede ver un grabado de los globos austriacos no tripulados sobre la ciudad asediada de Venecia.



Figura 2-9 Globos austriacos sobre Venecia, 1849 [25]

Algo menos de medio siglo más tarde, en 1898, el inventor de origen serbio Nikola Tesla crearía un sistema basado en el empleo de radiofrecuencias para poder controlar remotamente una pequeña embarcación. No obstante, no fue hasta que el ingeniero español Leonardo Torres Quevedo idea su propio sistema de control a distancia llamado “*Telekino*” en 1903, el cual es capaz de realizar movimientos mecánicos a distancia y que funciona por medio de ondas electromagnéticas, que se contempla esta modalidad de control para la aviación al buscar el ingeniero probar sus diseños propios de globos aerostáticos sin tener que arriesgar vidas humanas [26].

Durante ambas guerras mundiales, y durante el periodo de entreguerras, se lleva a cabo un tímido avance en el ámbito de los vehículos aéreos no tripulados, potenciado por el exponencial aumento en la financiación para el desarrollo de tecnologías bélicas y la constante, cada vez más emergente, aplicación al campo de batalla de nuevas invenciones científicas. Enfocado el desarrollo principalmente en la producción de blancos aéreos, se llegó a crear armamento aéreo no tripulado, como son ejemplo los

misiles V-1 alemanes que se pueden ver en la Figura 2-10, los cuales tenían un sistema de guiado que les permitía maniobrar como un piloto lo haría, aunque no se puede decir que fuesen tripulados remotamente ya que su sistema de control no era capaz de alterar el punto de destino sino tan solo maniobrar para alcanzarlo.

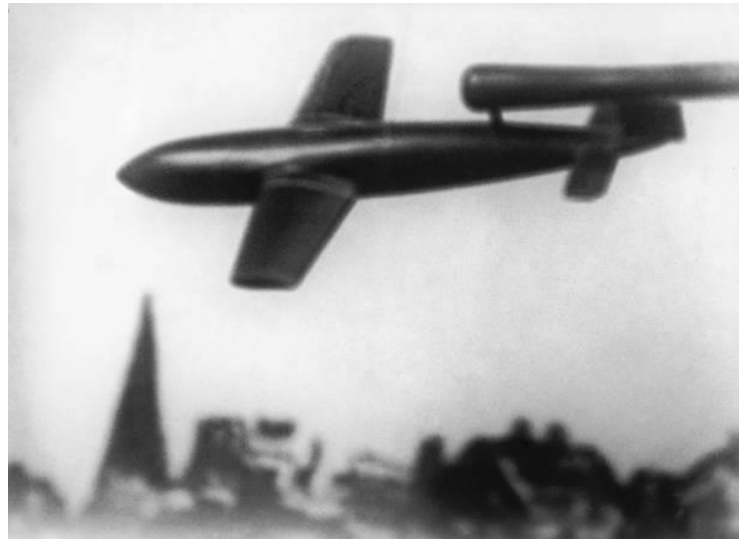


**Figura 2-10 Misil guiado V1 alemán [27]**

En los años de posguerra y durante los comienzos de la Guerra Fría se desarrolló con gran similitud tecnológica a sus compañeras tripuladas, aeronaves no tripuladas a reacción. No obstante, estas eran poco más que aeronaves y sus cometidos eran muy limitados. Es a partir de mediados de los años 60 y durante los años 70 que se comienzan a emplear estos VANTs en misiones de combate y de reconocimiento con gran precisión fotográfica, siendo finalmente desarrollado por Israel el primer dron con capacidad de vigilancia en tiempo real, el IAI *Scout* [28], que se puede ver en la Figura 2-11.



**Figura 2-11 UAV israelí IAI Scout [29]**

A partir de los años 80 y 90 la tecnología desarrollada para los UAVs comenzó a ser equiparable al resto de aeronaves de vigilancia y algunos comenzaron a incorporar armamento de gran calibre para misiones de combate, como es el caso del estadounidense MQ-1 *Predator*, puesto en servicio en 1995 y al cual se le equipaba con misiles aire-tierra de guiado laser AGM-114 *Hellfire* [30]. Con la entrada del nuevo milenio, la tecnología cada vez más avanzada ha normalizado la presencia de los drones en multitud de ámbitos de la sociedad.

### 2.3.2 Los drones en la actualidad

En la actualidad, el avance tecnológico de los drones es vertiginoso. Estos han experimentado un desarrollo exponencial gracias a su integración y aplicación en multitud de ámbitos, desde levantamientos cartográficos a control del tráfico e inspecciones industriales o agrícolas entre otros muchos ejemplos. Su campo de aplicación se extiende cada vez más gracias a la facilidad, sencillez y bajo coste con que estos equipos son capaces de desarrollar sus cometidos, llegando a lugares de difícil o complejo acceso para los medios tradicionales y permitiendo hacerlo con mayor comodidad que el resto de aeronaves, que no son tripuladas remotamente.

Debido a su versatilidad, la comercialización de los drones ha conllevado un aumento significativo de la importancia de esta herramienta en diferentes ámbitos de la sociedad. Esto ha resultado en una constante campaña de desarrollo e innovación de los sistemas aéreos pilotados remotamente, dando lugar a multitud de categorías y arquitecturas estructurales en que se encuadran los distintos UAVs. La diversa gama de tipologías da lugar a un extenso paraguas de empleos y prestaciones, confiriendo gran adaptabilidad y flexibilidad a este tipo de aeronaves [31]. En la Figura 2-12 se pueden ver las diferentes categorías en que se encuadran los drones a día de hoy en función de su arquitectura, pudiendo luego el tamaño de estos variar de forma significativa independientemente de su forma estructural.

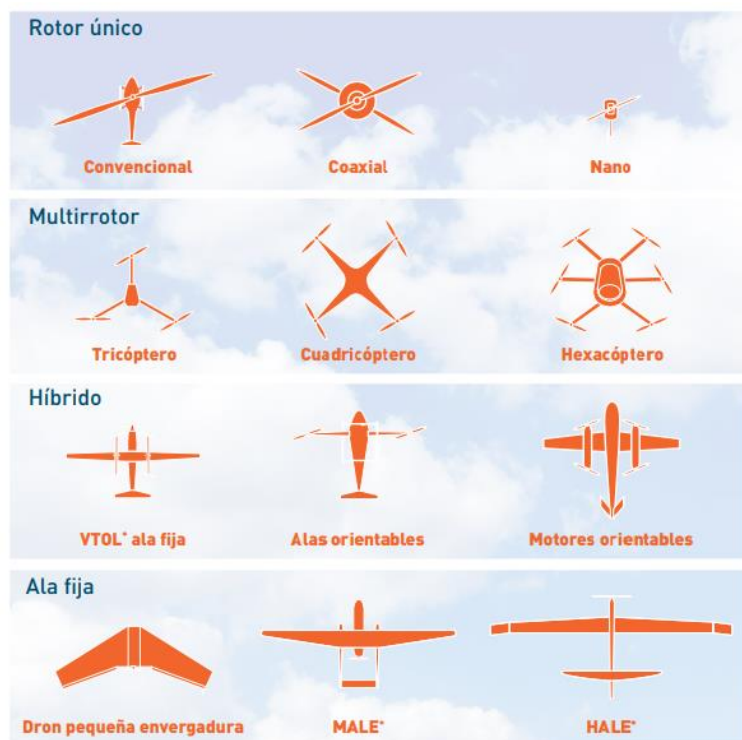


Figura 2-12 Tipologías de drones [31]

### 2.3.3 Los drones en el mundo civil

Como se ha constatado en el apartado anterior, los drones se han hecho con un papel de importancia en la sociedad civil en los últimos años. Los distintos tipos de drones civiles se emplean en aplicaciones tan variadas como la entrega de paquetes o el monitoreo de incendios forestales [32]. Estas aeronaves no tripuladas de uso comercial están diseñadas para ser fáciles de emplear y seguras, lo que las hace accesibles a una amplia variedad de usuarios, desde empresas de gran entidad hasta usuarios particulares.

En la entrega de paquetes, los drones han demostrado ser una alternativa competitiva y en algunos casos más rápida y eficiente que los métodos tradicionales. Empresas como *Amazon* y *DHL* están liderando el camino en la implementación de estas herramientas en el sector, lo que puede reducir significativamente el tiempo de entrega y los costos de envío.

Otro de los usos más fomentados de estos sistemas, especialmente en el ámbito lúdico y recreativo, es la fotografía aérea. Gracias al empleo de los drones se ha revolucionado la manera en que se capturan imágenes. Con el constante desarrollo y exponencial mejora de las cámaras aéreas, esta modalidad de fotografía se ha vuelto cada vez más de mayor accesibilidad al público general, lo que ha llevado a aumentar la demanda de compra conllevando el subsecuente desarrollo de un mayor catálogo de oferta por parte de distintas empresas enfocadas en el empleo recreativo de drones a un nivel de complejidad del sistema y características operacionales limitado. También desde una perspectiva más profesional, los drones han significado un importante avance a la hora de realizar estudios de la tierra. Estos medios pueden proporcionar imágenes de alta resolución y con una amplia perspectiva que sería imposible obtener con cámaras terrestres, o de gran coste con medios aéreos convencionales. En otros campos de importancia, este avance ha permitido a arquitectos, ingenieros o topógrafos realizar inspecciones más precisas y eficientes de edificios, puentes, carreteras y demás obras civiles.

Los drones también son útiles en situaciones de emergencia, como incendios forestales y desastres naturales, donde son empleados para la monitorización de la situación y guiar a los equipos de respuesta [32].

Sin embargo, los drones a día de hoy también plantean desafíos en términos de privacidad y seguridad. El público general presenta consternación por la posibilidad de que los drones sean empleados para espiar o invadir la privacidad de las personas, así como la capacidad que estos equipos poseen de entorpecer o perjudicar el correcto funcionamiento y la seguridad de los vuelos cerca de aeropuertos y aviones comerciales. A raíz del desarrollo de estas problemáticas, se han establecido regulaciones gubernamentales en multitud de países para garantizar la seguridad y privacidad que los drones pueden poner en entredicho. Las limitaciones establecidas comprenden restricciones de vuelo en ciertas áreas, requisitos de registro, limitaciones en la altura de vuelo y demás condicionamientos que se desarrollan e implementan cada día.

## 2.4 Los drones en las instituciones del Estado

Para cumplir con su función de manera más eficaz, las diferentes instituciones públicas se han ido adaptando al compás de los tiempos y a las nuevas tecnologías que el desarrollo ha traído consigo. En los últimos años, el auge de los drones ha despertado en multitud de organismos la curiosidad por descubrir si su implantación a su estructura operacional contribuiría al aumento de sus capacidades funcionales. A día de hoy, son varias las carteras institucionales que han decidido llevar a cabo la integración de esta capacidad aérea. Desde el Ministerio de Defensa hasta las unidades más pequeñas de dependencia estatal, como pueden ser los ayuntamientos, han ejercido su empleo [33], pasando por los presentes desarrollos de programas en todos los espectros del Estado, como el proyecto *Delorean* de Correos [34] que se puede ver en la Figura 2-13 pero que todavía no se encuentra operativo.



Figura 2-13 Dron proyecto *Delorean* de Correos [34]

### 2.4.1 Dirección General de Tráfico

La Dirección General de Tráfico tiene por objetivo la ejecución de la política vial en las carreteras españolas, comprendiendo esto su seguridad y la comodidad en las vías. Para ejecutar sus cometidos en cuanto a la seguridad en la carretera concierne, este organismo autónomo dependiente del Ministerio del Interior cuenta con una flotilla de aparatos de vuelo que se encargan de patrullar los caminos de titularidad estatal de España. Estas aeronaves hasta 2018 [35] se trataban de helicópteros, pero a partir de la fecha se comenzó la distribución por el territorio nacional de 39 drones de vigilancia finalizando en 2021 [35]. Las características de los drones de los que dispone la DGT a día de hoy aparecen reflejadas en la Tabla 2-2.

| Modelo             | Unidades | Peso   | Autonomía  | Alcance | Sistema óptico                                     |
|--------------------|----------|--------|------------|---------|----------------------------------------------------|
| <i>THYRA V109</i>  | 20       | 6,2 kg | 40 minutos | 10 km   | Videocámara con zoom digital x6<br>zoom óptico x40 |
| <i>MATRICE 200</i> | 3        | 6,1 kg | 40 minutos | 7 km    | Videocámara con zoom digital x6<br>zoom óptico x30 |
| <i>PHANTOM 4</i>   | 8        | 1,4 kg | 20 minutos | 3 km    | Videocámara sin zoom                               |
| <i>THYRA V105</i>  | 8        | 1,4 kg | 20 minutos | 3 km    | Videocámara sin zoom                               |

**Tabla 2-2 Drones de la DGT (tomada de [35])**

La altura efectiva de todos los dispositivos es de 120 metros y alcanzan velocidades de hasta 80 km/h sin poder operar por fuera del rango de temperatura comprendido entre los 20 y los 45 grados centígrados [35]. En la Figura 2-14 se puede apreciar uno de estos aparatos.



**Figura 2-14 Dron Matrice 200 de la DGT [35]**

Mediante esta capacidad aérea, ha aumentado exponencialmente la detección del uso de telefonía móvil al volante, así como los adelantamientos a ciclistas sin resguardo de distancias mínimas de seguridad y el empleo incorrecto del cinturón de seguridad y de sistemas de retención infantil. No obstante, cabe reseñar que estos equipos no están equipados con los sistemas necesarios para detectar el exceso de velocidad en los vehículos en circulación.

### 2.4.2 Vigilancia Aduanera

El Servicio de Vigilancia Aduanera (SVA) es una Dirección Adjunta dependiente tanto funcional como orgánicamente del Departamento de Aduanas e Impuestos Especiales, integrado en la Agencia Estatal de Administración Tributaria (AEAT) del Ministerio de Hacienda [36]. Esta institución es un cuerpo armado y sus agentes tienen carácter de agentes de la autoridad como policía fiscal y judicial y, por ello, están autorizados a portar armamento reglamentario para acometer con sus funciones institucionales y líneas de acción. Estas son el resguardo aduanero, la vigilancia marítima y las investigaciones y pesquisas necesarias resultantes de la lucha contra el contrabando, el blanqueo de capitales, el fraude fiscal y la seguridad en sus áreas de operación [37].

Para llevar a cabo sus cometidos, Vigilancia Aduanera cuenta con una plantilla de 1850 funcionarios que se encargan entre otras cosas de mantener y operar una flota de medios marítimos y aéreos. Con estos medios se efectúan misiones de vigilancia aeronaval a la que desde hace poco se han integrado unidades aéreas no tripuladas, que proporcionan mayor discreción y seguridad que los helicópteros con los que operan de forma más frecuente [38]. Entre los drones que operan destacan las unidades comerciales de la marca DJI, como el *Matrice 200*, operado también por la DGT.

### 2.4.3 Salvamento Marítimo

Esta institución estatal tiene por fin último el salvamento de vidas humanas en la mar. Así mismo, también realiza tareas de prevención y lucha contra la contaminación del medio marino, el remolque de embarcaciones auxiliadas y el seguimiento y ayuda al tráfico marítimo, fomentando la seguridad marítima y a la navegación. Estos deberes los cumple en la totalidad de las aguas territoriales españolas gracias a una importante capacidad de medios aeronavales [39].

Si bien Salvamento Marítimo, a día de redacción, no está aún equipado con medios aéreos tripulados remotamente, ha realizado multitud de ejercicios integrando estas herramientas gracias a la colaboración de agencias europeas e internacionales como la EMSA (*European Maritime Safety Agency*) que aportan sus drones con el fin de la mejora en las capacidades de la institución [40]. A raíz de estos, Salvamento Marítimo ha puesto en marcha el proyecto iSAR, que busca entre otras cosas el desarrollo de drones de altas prestaciones para la potenciación de su capacidad de búsqueda de naufragos en la mar [41].

### 2.4.4 Policía Nacional

El Cuerpo Nacional de Policía (CNP) ha utilizado drones para una amplia variedad de tareas por su capacidad para brindar una perspectiva aérea. Tanto en situaciones de emergencia, como para monitorizar áreas de interés, con la policía judicial o en operaciones. En su expediente operativo constan, entre otras, misiones de búsqueda y rescate, vigilancia, control del tráfico, e investigación criminal. La gama de modelos que emplea la policía es amplia aportan capacidades muy variadas, priorizando algunos modelos la calidad de las imágenes, frente a otros que priorizan el tiempo de vuelo o la velocidad máxima. Al igual que muchas instituciones y agencias del Estado, como se puede ver en la Figura 2-15 sus modelos proceden del mercado comercial, resaltando la presencia de los drones de la marca DJI [42].



**Figura 2-15 Dron del CNP [43]**

### 2.4.5 Guardia Civil

La Guardia Civil ha dotado a todas las Comandancias de España con UAS de la marca comercial DJI modelos *Zoom* y *Enterprise Advanced*. Estos drones de desarrollo civil se encuentran desplegados por todo el territorio nacional para el uso diario de los guardias ya que aportan una buena capacidad de reacción ante cualquier incidencia, pudiéndose desplegar en un breve espacio de tiempo y con una superficie de plataforma de despegue reducida. Así mismo, para las Comandancias de Ceuta y Melilla se ha adquirido también un DJI *Matrice 300 RTK* para el control de la inmigración y el apoyo a la vigilancia fronteriza.

La institución, además, ha dotado a unidades específicas con drones que se ajustan de forma más precisa a sus necesidades del servicio. Estas unidades son el Grupo de Apoyo Operativo (GAO), la Unidad Especial de Intervención (UEI), la Policía Judicial y la Unidad Central Operativa (UCO), que usan además de los equipos mencionados anteriormente, otro tipo de RPAS más específicos como son el *Indago II*, el *Matrice 300 RTK*, el *Matrice 600 PRO* y el *Black Hornet* del cual se hablará, más detalladamente, en el siguiente apartado [44]. Así mismo, también emplea la división de tráfico los drones *Matrice 200*, *Phantom 4* y *Thyra V109* que usa la DGT [45]. En la Tabla 2-3 se pueden ver las características más relevantes de los principales drones que emplea la Guardia Civil.

| Modelo                     | Peso   | Autonomía  | Alcance | Sistema óptico                  |
|----------------------------|--------|------------|---------|---------------------------------|
| <i>ZOOM</i>                | 0,9 kg | 31 minutos | 8 km    | Videocámara con zoom óptico x2  |
| <i>ENTERPRISE ADVANCED</i> | 0,9 kg | 31 minutos | 8 km    | Videocámara con zoom óptico x32 |
| <i>MATRICE 300 RTK</i>     | 9 kg   | 55 minutos | 15 km   | Videocámara con zoom óptico x3  |
| <i>INDAGO II</i>           | 1,8 kg | 50 minutos | 7 km    | Videocámara sin zoom            |
| <i>MATRICE 600 PRO</i>     | 9,5 kg | 35 minutos | 3,5 km  | Videocámara sin zoom            |

Tabla 2-3 Características de los drones de la Guardia Civil [46]

### 2.4.6 Ejércitos e Infantería de Marina

El Ejército de Tierra y el Ejército del Aire y del Espacio son las instituciones armadas encargadas de velar por la defensa de España y de sus intereses en el ámbito terrestre y en el ámbito aéreo respectivamente. Así mismo, la Infantería de Marina de la Armada lleva a cabo labores de naturaleza similar a estos ejércitos. Para poder cumplir, estas instituciones, con sus cometidos, además de disponer de una amplia variedad de medios de diferentes características, en los últimos años han integrado en sus unidades sistemas de vuelo aéreo no tripulado en multitud de modelos.

Respecto a su clasificación, los RPAS españoles se encuadran en las Clases I, II y III de características de drones establecidas por la OTAN (Organización del Tratado del Atlántico Norte), como se puede ver en la Figura 2-16, y definidas en función del peso, distancia y altitud que alcanzan los equipos. La primera clase encuadra a los sistemas de menor porte y se divide a su vez en tres subcategorías: *Micro*, *Mini* y *Small*. La primera comprende hasta los 2 kg de peso, la segunda hasta los 15 y la última llega hasta los 150 kg. Los drones de esta clase se emplean en el teatro de operaciones de una forma táctica y de último escalón por las unidades desplegadas en el escenario. Los drones de Clase II se pueden emplear a un nivel tanto táctico como operacional, y los de Clase III se utilizan para misiones de ámbito operacional o hasta estratégico. Esta tercera clase contemplada se utiliza para

designar a aquellas aeronaves que superan significativamente las capacidades de los RPAS de las clases inferiores, pudiéndose dividir en MALE (del inglés *Medium Altitude Long Endurance*) o HALE (*High Altitude Long Endurance*) y rivalizando con aeronaves tripuladas estratégicas de largo alcance.

| NATO UAS CLASSIFICATION       |                  |                                          |                           |                       |                             |                  |
|-------------------------------|------------------|------------------------------------------|---------------------------|-----------------------|-----------------------------|------------------|
| Class                         | Category         | Normal Employment                        | Normal Operating Altitude | Normal Mission Radius | Primary Supported Commander | Example Platform |
| Class III<br>(> 600 kg)       | Strike/Combat*   | Strategic/National                       | Up to 65,000 ft           | Unlimited (BLOS)      | Theatre                     | Reaper           |
|                               | HALE             | Strategic/National                       | Up to 65,000 ft           | Unlimited (BLOS)      | Theatre                     | Global Hawk      |
|                               | MALE             | Operational/Theatre                      | Up to 45,000 ft MSL       | Unlimited (BLOS)      | JTF                         | Heron            |
| Class II<br>(150 kg - 600 kg) | Tactical         | Tactical Formation                       | Up to 18,000 ft AGL       | 200 km (LOS)          | Brigade                     | Hermes 450       |
| Class I<br>(< 150 kg)         | Small (>15 kg)   | Tactical Unit                            | Up to 5,000 ft AGL        | 50 km (LOS)           | Battalion, Regiment         | Scan Eagle       |
|                               | Mini (<15 kg)    | Tactical Subunit (manual or hand launch) | Up to 3,000 ft AGL        | Up to 25 km (LOS)     | Company, Platoon, Squad     | Skylark          |
|                               | Micro ** (<66 J) | Tactical Subunit (manual or hand launch) | Up to 200 ft AGL          | Up to 5 km (LOS)      | Platoon, Squad              | Black Widow      |

Figura 2-16 Clasificación de los RPAS según estándares OTAN [47]

Dentro de los planes de integración de unidades no tripuladas de ámbito aéreo en las FFAA, se está llevando a cabo un programa de desarrollo y adquisición de drones Clase I de fabricación nacional para su despliegue con unidades de acción. Denominado *Rapaz*, esta iniciativa ha derivado en el aumento de modelos y tipos de UAV en uso a día de hoy por parte de las Fuerzas Armadas [48]. Así mismo, la necesidad de capacitar a la Fuerza de sistemas de vigilancia no tripulados de largo alcance ha derivado en la adquisición de equipos ampliamente capacitados y tecnológicamente avanzados que ayudan a poner a España en la vanguardia del sector.

Los medios empleados por el Ejército de Tierra (ET), el Ejército del Aire y del Espacio (EA), la Infantería de Marina de la Armada y la Unidad Militar de Emergencias son los siguientes:

- PD-100 *Black Hornet*. Aparato *Micro* empleado tanto por la Legión como por el Regimiento de Inteligencia nº1 y las diferentes unidades de operaciones especiales (incluida la Fuerza de Guerra Naval Especial de la Armada), se trata del equipo aéreo de menor tamaño en uso en las FFAA. Con una longitud de 16 cm y pesando menos de 33 gramos que albergan en su interior tres cámaras para visión tanto diurna como nocturna, este dron se despliega y recupera desde el agarre de su controlador como se puede ver en la Figura 2-17. Su autonomía es de hasta 25 minutos con un alcance de 2 km y su condición acústica es extremadamente silenciosa para evitar su detección en ambientes comprometidos [49].



**Figura 2-17 Operador con un *Black Hornet* [50]**

- *Huginn X1*. Adquiridos inicialmente por la Armada para los estoles de la Fuerza de Guerra Naval Especial (FGNE) y para los infantes de marina del Tercio de Armada en 2016, en 2018 lo adquirió también el ET para sus unidades de operaciones especiales, el Mando de Operaciones especiales (MOE) [51]. Cuadricóptero de apenas 940 gramos, la primordial ventaja de ese dron *Micro* es la capacidad de desplegarlo en apenas 30 segundos, permitiendo una elevada capacidad de respuesta a las unidades ante estímulos externos desconocidos de su entorno. Siendo altamente portátil por el operador, su alcance es de 2 km y su velocidad máxima es de 21,6 km/h con una permanencia en vuelo de 25 minutos [52].
- *Phantom II*. Este micro UAV de cuatro rotores es de la misma familia que el *Phantom 4* empleado por la DGT y su comercialización es civil. Su peso al despegue es de 1,3 kg y tiene un alcance máximo de 1 km. Se han empleado desde 2018 por parte de las dotaciones de la base antártica del Ejército de Tierra *Gabriel de Castilla*, permitiendo por parte de estos para realizar levantamientos topográficos o establecer rutas de evacuación de emergencia [53]. También la Unidad Militar de Emergencias (UME) emplea 5 de estos equipos que han sido usados en multitud de situaciones críticas humanitarias como los terremotos de Nepal de 2015 [54].
- *Wasp AE RQ12A*. Empleado tanto por el Escuadrón de Zapadores Paracaidista del Ejército del Aire (su unidad de operaciones especiales) como por el Tercio de Armada (TEAR) de Infantería de Marina de la Armada [55]. Este dron tiene un peso al igual que el *Phantom II* de 1,3 kg, pero su alcance es mayor, con 5 km de margen de maniobra. Tiene una velocidad máxima de 83 km/h y su techo es de 91 metros con 50 minutos de permanencia en vuelo. Siendo lanzado manualmente, este dron puede realizar aterrizajes en multitud de superficies, incluyendo la capacidad de hacer amerizajes [56].
- *Geodrone*. La UME adquirió en 2017 este vehículo aéreo no tripulado cuya finalidad para la cual se diseñó es el cartografiado y levantamiento topográfico de extensas áreas de forma autónoma. Su origen civil aporta características interesantes a la UME como el de la capacidad de modelado 3D de relieve y edificaciones, proporcionado por los sensores del dron que puede dar perspectivas no exploradas en situaciones de crisis [57]. Su velocidad máxima es de 80 km/h, tiene 45 minutos de autonomía y su peso es de 2,8 kg con un alcance

de 45 km. Para evitar el uso de lanzaderas aparatosas en caso de operar en entornos con poca superficie disponible, el despegue de la aeronave se lleva a cabo desde un trípode. A parte de la Unidad Militar de Emergencias, para levantamientos geográficos también están experimentando con el *Geodrone* el Centro Geográfico del Ejército de Tierra y el Centro Cartográfico y Fotográfico del Ejército del Aire y del Espacio.

- *Raven RQ11B*. Desarrollado por la misma empresa que el *Wasp AE*, este fue de los primeros RPAS que desplegó España en una zona de operaciones al hacerlo en Afganistán en 2008. Tras una primera adquisición en 2007 del modelo RQ11A, en el año 2015 se renovó y actualizó la flotilla al 11B, que se puede ver en la Figura 2-18, y se aumentó a las cerca de 35 unidades que operan el Ejército de Tierra y el Ejército del Aire y del Espacio. Su adquisición se plantea dentro de la imperante necesidad de proporcionar a los infantes desplegados una capacidad integrada de obtención de inteligencia y apoyo visual aéreo. De ahí que las características más reseñables de este sistema sean su práctica portabilidad y despliegue. No obstante, desde su primera integración a las FFAA, han realizado multitud de operaciones de temáticas y fines variados como la operación *Centinela Gallego* de vigilancia de incendios forestales en Galicia [58].



Figura 2-18 Preparación de *Raven* para su despliegue [53]

Este UAV está diseñado para ser lanzado a mano ya que su peso es de apenas 2,2 kg y tiene una envergadura de 1,4 m, pudiendo aguantar en vuelo hasta 90 minutos con un techo de 152 metros, alcance de 10 km y velocidades máximas de hasta 81 km/h [59]. Sus sensores modernizados permiten una captación 360° de video en tiempo real tanto de día como de noche, mediante cámaras infrarrojas, así como la capacidad de operar de relé entre diferentes aeronaves.

- *Seeker*. La Brigada de Infantería de Marina del Tercio de Armada, junto con la Brigada Almogávares VI de Paracaidistas del Ejército de Tierra, son las dos únicas unidades de las Fuerzas Armadas dotadas con el *Seeker*. Este UAV es un equipo de fabricación y diseño nacional que con sus 90 minutos de permanencia en vuelo y su alcance de 15 km dota a las unidades expedicionarias de una gran capacidad de conocimiento del entorno de operaciones. Estando capacitado para su integración en centros de control OTAN, sus sensores integrados permiten la obtención de información visual y térmica en tiempo real [60]. De ala fija, este dron entra dentro de la categoría *Mini* con un peso de 3,5 kg y son las secciones de inteligencia de las unidades las que los emplean.
- *Condor*. De seis rotores y perteneciente a la categoría *Mini*, este UAV de 5 kg tiene un alcance considerable de 18 km con una permanencia en vuelo de 70 minutos y un importante techo de 16.000 pies [61]. De fabricación española, es empleado por unidades del Ejército

del Aire y del Espacio y por los batallones de la UME que lo utilizan en sus labores de rescate y mitigación de los efectos causados por desastres naturales [53]. También pueden ser empleados para llevar a cabo tareas de Vigilancia y reconocimiento tanto en ámbitos terrestres como marítimos y para el transporte de soportes vitales en operaciones de salvamento.

- *Tucán*. También un aparato encuadrado en la categoría *Mini*, el peso de este dron de ala fija no supera los 5 kg y su envergadura es de 2,74 m como se puede apreciar en la Figura 2-19. Con un techo de 8.200 pies, menor que el *Condor*, sí que supera a este en alcance al poder volar hasta a 25 km de distancia y con una autonomía también superior de 90 minutos. Su velocidad máxima es de 100 km/h y la de crucero que permite su considerable autonomía es de 65 km/h [62].



**Figura 2-19 Tucán sobre su plataforma de despegue [63]**

Adquirido en 2017 por su versatilidad y su control desde una estación portátil robusta, resistente y duradera, su empleo en las FFAA lo llevan a cabo principalmente el Grupo de Artillería de Información y Localización (GAIL) del Regimiento de Artillería de Campaña N°.63 del Ejército de Tierra para operaciones de reconocimiento y adquisición de objetivos, así como el Centro Nacional de Adiestramiento *San Gregorio* para tareas de vigilancia y control perimetral y del medioambiente. En el Ejército del Aire y del Espacio, su uso se da por parte de la Escuela de Sistemas Aéreos no Tripulados de Maticán para llevar a cabo la formación de los controladores de drones de Clase I [53].

- *Alpha 800*. Dentro del marco del proyecto *Rapaz*, la Dirección General de Armamento y Material adquirió en 2018 dos *Alpha 800* para el Ejército de Tierra [64]. Estos drones convencionales de un único rotor pesan 14 kg y están propulsados por un motor de combustión de gasolina que les proporciona una autonomía de vuelo de 3 horas gracias al tamaño de su tanque de combustible. Su alcance máximo es de 30 km y dispone de un sistema de pilotaje automático por GPS [65].
- *Fulmar*. Ya de categoría *Small*, el peso al despegue es de 20 kg para este dron de ala fija, de 3 m de envergadura, reflejado en la Figura 2-20.



**Figura 2-20 Dron Fulmar de THALES España [66]**

Con una autonomía de 8 horas, puede operar hasta a 80 km de distancia a una velocidad media de 100 km/h, alcanzando una altitud de 1.300 pies. En caso de amerizaje de emergencia, tiene una flotabilidad asegurada de 30 minutos y tiene una baja firma sonora y radárica [66]. Desplegado también en la escuela de Matacán por el Ejército del Aire, este sistema también lo opera el Ejército de Tierra y la Infantería de Marina para misiones de reconocimiento, inteligencia y vigilancia.

- *Alba*. Un blanco aéreo ligero, esta aeronave no tripulada se emplea por el Ejército de Tierra para realizar prácticas de tiro al cañón o del misil *Mistral*, para cuyas municiones está especialmente adaptado. Con 2,2 metros de envergadura y 25kg de peso al despegue, puede alcanzar los 180 km/h y mantenerse en vuelo durante 60 minutos [67]. Este VANT sirve en la Unidad de Blancos Aéreos del Cuartel General del Mando de Artillería Antiaérea y presta servicios en apoyo al adiestramiento de multitud de unidades que lo requieren [68].
- *Orbiter 3*. En 2018, el Ejército de Tierra adquirió dos sistemas *Orbiter 3* compuestos, cada cual, por 3 aeronaves, para ser un total de 6, y que han llegado a operar en operaciones como *Inherent Resolve* en Irak. A partir de 2022, se comenzó la actualización de estos equipos para dotarles de mejores cámaras y capacidades de vigilancia, así como mejoras en los sistemas de control y capacidad de vuelo. Con una envergadura de 4,4 metros y un peso de 30 kg, su alcance gracias a las actualizaciones efectuadas del sistema de enlace es de un máximo de 150 km a una velocidad máxima de hasta 130 km/h [69].
- *Atlantic*. Un VANT de ala fija, el *Atlantic* también pertenece a la categoría *Small* ya que su peso es de 55 kg y su envergadura de 3,8 metros. Tiene un techo de 12.000 pies y su alcance es de 100 km con una permanencia en vuelo de hasta 8 horas a velocidad crucero de 120 km/h pudiendo dar hasta 170 km/h [70]. Dron de producción nacional por la misma empresa que desarrolló el *Tucán*, la plataforma está fabricada en materiales compuestos de gran resistencia y ligereza [71]. Este equipo que se puede ver en la Figura 2-21 se emplea también al igual que el *Tucán* en el GAIL del Ejército de Tierra, donde destaca por la simplicidad de su uso [53].



**Figura 2-21 *Atlantic* en el GAIL [71]**

- *SIVA*. El Sistema Integrado de Vigilancia Aérea (SIVA) es un sistema de desarrollo nacional por parte del Instituto Nacional de Técnicas Aeroespaciales (INTA) que se integró en las FFAA en 2006. La adquisición de estas aeronaves se efectuó debido a la presencia española en Afganistán, donde no contaban las tropas nacionales con apoyo de RPAS de Clase II. No obstante, al no haberse testado este sistema en entornos hostiles y de alta demanda operativa, al final el ET se decantó por desplegar el *Searcher* que se estudiará a continuación [72]. De 300 kg al despegue, este VANT tiene un alcance de 150 km, una velocidad máxima de 190 km/h, y un techo operativo de 13.000 pies, pudiéndose mantener en vuelo durante 7 horas [73]. Actualmente, presta servicio en el Ejército de Tierra en el GAIL, así como en la Escuela de Sistemas Aéreos no Tripulados de Matacán del EA, si bien los sistemas operados

en esta última unidad no son equipos del Ejército del Aire, sino del INTA, cedidos desde 2012 a la escuela por un convenio de colaboración para el adiestramiento de pilotos. Compuesto por la aeronave tripulada remotamente y la estación de control, el sistema SIVA tiene la capacidad de ser desplegado en multitud de escenarios. Esta aptitud es posible gracias a la movilidad de la estación de control y a la capacidad de despegue y aterrizaje automático de la aeronave, la cual integra así mismo un tren de aterrizaje adaptado para poder operar en pistas de tierra.

- *Searcher MK.III*. Fabricado en Israel, este dron de ala fija se adquirió de urgencia en su versión MK.II-J a finales de 2007 para, como se ha mentado anteriormente, su despliegue en apoyo a las tropas del ET que entonces se encontraban en Afganistán, en la provincia de Badghis. A lo largo de este despliegue, que se prolongó por parte de estas aeronaves hasta 2014, se accidentan 2 de las 4 unidades iniciales, si bien en 2009 se comprarían cuatro plataformas aéreas más [74] y se irían actualizando las plataformas ya desplegadas. De este periodo operacional se obtuvo la demostración de las capacidades del sistema, caracterizándose su baja firma sonora y radárica así como su gran alcance y autonomía, proporcionando, tanto a los puestos de mando y control desplegados como a los de territorio nacional, imágenes en tiempo real de gran calidad y nitidez [75]. El peso máximo de estos RPAS que se puede ver en la Figura 2-22 es de 427 kg, con una envergadura de 8,55 m y pudiendo volar durante 12 horas hasta una distancia de 250 km. Su techo es de 19.000 pies y su velocidad máxima es de 315 km/h [76].

La denominación del Ejército de Tierra de este sistema es PASI, Plataforma Aérea Sensorizada de Inteligencia, y se encuadra dentro de la Clase II de categorías de drones, siendo su empleo el de recolección de inteligencia y apoyo a las tropas desplegadas mayoritariamente a nivel táctico.



**Figura 2-22 Searcher MK.III [53]**

A día de redacción, se están llevando a cabo varios programas de desarrollo para implementar la sustitución de este veterano sistema. Entre los candidatos planteados, destaca el SIRTAP (Sistema Remotamente Tripulado de Altas Prestaciones). Este proyecto, de desarrollo conjunto entre España y Colombia, comenzó en el año 2015 y la intención es que comience a sustituir a los *Searcher* a partir de 2026. El compromiso inicial es que el Ministerio de Defensa adquiera 9 sistemas compuestos por 3 aeronaves cada uno, esto es, un total de 27 drones. De mayor porte que el VANT al que sustituirá, su peso máximo al despegue será de 750 kg y tendrá una importante envergadura de 12 metros, llegando a ser el primer RPAS Clase III MALE de fabricación nacional [77].

- *Predator B*. La plataforma aérea tripulada a distancia de mayor porte de cualquiera de las instituciones del Estado, el *Predator B* (cuya nomenclatura técnica es MQ-9 *Reaper*) con designación del Ejército del Aire EA NR05, fue integrado en las Fuerzas Armadas en el año 2019 con la entrega de las 2 primeras unidades de 4 que se compraron por adjudicación a *General Atomics Aeronautical Systems* [78]. Prestando servicio en el 233 Escuadrón del Ala 23 en Talavera la Real, este RPAS tipo MALE que se puede ver en la Figura 2-23 tiene una envergadura de 20 metros y un peso máximo de 4,7 toneladas, lo que hace que su lanzamiento y recuperación deban de realizarse desde una pista de aterrizaje, limitando su capacidad de despliegue sobre medios navales. Sus características estructurales combinadas con los 925 caballos de vapor generados por su turbohélice permiten a esta aeronave volar durante 25 horas con una velocidad máxima de 405 km/h, teniendo un alcance de 3.000 km y un techo de altitud de hasta 50.000 pies.



**Figura 2-23 *Predator B* del Ejército del Aire y del Espacio [78]**

Los sensores de esta aeronave permiten la recabación de inteligencia y la vigilancia y reconocimiento de objetivos con gran precisión y facilidad. Estos se tratan de un conjunto electroóptico e infrarrojo multiespectral y direccional de gran alcance y detalle, de un designador de blancos (MTS-B) y de un radar multimodal (*Lynx APY-8*) con GMTI (*Ground Moving Target Indicator*) que permite la detección de blancos móviles [79].

Si bien la adquisición de este RPAS se ejecutó con el planteamiento inicial de la obtención de una herramienta de operaciones de inteligencia, sin la integración de software de combate ni la capacidad de portar y emplear armamento, en la fecha de redacción de este TFG se están llevando a cabo estudios de implementación a la aeronave de sistemas de armas y soportes subalares para el porte de armamento y su empleo en posibles misiones de ataque. De igual manera, se está estudiando la posibilidad de un próximo despliegue de estas unidades en Yibuti, donde el Ejército del Aire y del Espacio tiene un destacamento de apoyo a la Operación Atalanta de lucha contra la piratería en la que se integran recurrentemente unidades de la Armada [80].

### 2.4.7 Cuerpo General de la Armada

La Armada es la rama naval de las fuerzas armadas españolas y su cometido es proteger y defender a España, y velar por sus intereses, en y desde la mar. Para ello cuenta con una amplia dotación de personal formado y experimentado y con buques, unidades, centros y demás medios de gran modernidad tecnológica y eficiencia y efectividad en los cometidos para los que han sido concebidos.

Para acometer sus principales deberes institucionales, la armada cuenta con la Flota, donde encuadra sus unidades de acción. Dentro de la Flota se encuentra la Flotilla de Aeronaves (FLOAN) que, responsable de la aviación naval, consiste en escuadrillas que se conforman en función de los diferentes tipos de aeronaves de la flotilla. Con escuadrillas de tanto unidades de ala fija como de ala rotatoria, hasta febrero de 2023, cuando se ha creado la duodécima escuadrilla para los nuevos helicópteros H-135 P3H adquiridos por la Armada [81], la última escuadrilla en conformarse había sido en 2014 [82]; la de unidades aéreas tripuladas de forma remota, la 11ª Escuadrilla de Aeronaves [83], cuyo escudo se puede ver en la Figura 2-24.



Figura 2-24 Escudo de la 11ª Escuadrilla de Aeronaves [83]

La escuadrilla, que ha operado en múltiples escenarios desde su creación, desempeña cometidos de vigilancia marítima, búsqueda y rescate, recolección de inteligencia mediante identificación, reconocimiento y vigilancia de objetivos de interés (misiones denominadas de ISR, *Intelligence, Surveillance and Reconnaissance*) y puede proporcionar apoyo a unidades de operaciones especiales, así como a entidades civiles en detección de incendios, control de fronteras, etc. Así mismo, también tiene como misión apoyar al adiestramiento de las unidades navales en el aspecto de tiro antiaéreo. Sus unidades aéreas no tripuladas se encuadran en la misma categorización OTAN que la tratada anteriormente para el Ejército de Tierra y el Ejército del Aire y del Espacio. Para acometer con sus deberes, la escuadrilla cuenta con los siguientes sistemas:

- *Scan Eagle*. Contando con 10 sistemas operativos, este es el equipo aéreo principal de la escuadrilla, que se puede ver en la Figura 2-25, y que se encarga del grueso de los cometidos asignados a esta unidad de drones. Su diseño se enfoca en la obtención de imágenes de gran calidad en tiempo real y con un amplio periodo de permanencia sobre su objetivo, evitando su detección. Para ello, cuenta con una autonomía de 24 horas de vuelo con un techo de cerca de 20.000 pies. Su alcance es de hasta 100 km (54 NM) en los que es capaz de operar con gran agilidad gracias a su velocidad de crucero de 111 km/h (60 kts) llegando a un máximo de 148 km/h.



**Figura 2-25 Scan Eagle de la Armada [84]**

Para la obtención de video cuenta con cámaras de alta resolución tanto diurnas como nocturnas y térmicas, así como sensores electroópticos e infrarrojos.

Con la adquisición de este dron, la Armada ha buscado obtener una capacidad de polivalencia al centrarse en un equipo desplegable en alta mar mediante su capacidad de embarque en buques de menor porte que aquellos donde se embarcan de forma tradicional aeronaves de ala fija. Para ello, el sistema dispone de una estación de control, desplegable bien en tierra o a bordo de un buque, donde se lleva a cabo el pilotaje de la aeronave, así como el análisis de los datos e imágenes obtenidas. Así mismo, por otro lado, esta aeronave no requiere de una pista de despegue ni de aterrizaje, sino que se basta como se puede ver en la Figura 2-26 de una plataforma de lanzamiento para proporcionarle la velocidad necesaria para su despegue, y de un equipo recuperador que “engancha” a la aeronave en su maniobra de aterrizaje.



**Figura 2-26 Scan Eagle en plataforma de lanzamiento [83]**

- **SCRAB II.** Este es el equipo de la escuadrilla que se encarga de apoyar a los buques y unidades de la flota en su adiestramiento de tiro antiaéreo y lanzamiento de misiles [83]. Se trata de un VANT que cumple con las funciones de blanco aéreo de entrenamiento bien para unidades de artillerías navales como en tierra.

Si bien su alcance y su techo operativo son los del *Scan Eagle*, de 100 km (54 NM) y 20.000 pies, y su envergadura es similar, la ficha técnica de este dron con notables prestaciones de vuelo no se asemeja en muchos otros factores a su compañero de escuadrilla. Aventaja a este con la velocidad, siendo capaz de alcanzar los 430 km/h (230 kts) que, si bien siguen distando de la velocidad de la mayoría de misiles antibuque en el mercado actualmente, su

acercamiento a estas es considerable. Su autonomía no obstante se limita a 60 minutos y los sensores y sistemas con los que cuenta se centran en el espectro de las comunicaciones para permitir tan largas distancias de radioenlace fiable que propician el largo alcance de este blanco aéreo [85].

- *Chukar III*. De la empresa estadounidense *Northrop Grumman*, este vehículo tripulado remotamente es, al igual que el *Scrab*, un blanco aéreo. No obstante, este entró en servicio en la Armada en el año 1994, 11 años antes que el *Scrab*, que lo hizo en 2005. Aun en servicio en la Armada, este dron es propulsado por un turborreactor que le permite alcanzar los 900 km/h con un techo de vuelo de 40.000 pies, pudiendo mantener el vuelo durante 1 hora. Debido al cierre de producción de este blanco por parte de su fábrica estadounidense, está siendo retirado paulatinamente de la Armada por la carencia de componentes de repuesto y difícil sostenimiento [86].
- *M5D-Airfox*. El propósito inicial de este dron de la Armada nada tenía que ver con el ámbito de la milicia. Ideado bajo la nomenclatura de *Tunadrone* por la empresa gallega *Marine Instruments* como una herramienta civil para la detección de bancos de atunes para pescadores de altura, se ha visto convertido en una nueva capacidad de la Armada tras la reciente adquisición de esta de 3 unidades para su empleo en labores de vigilancia marítima y de detección de minas.



Figura 2-27 M5D Airfox en su lanzadera [87]

Tiene hasta 10 horas de autonomía alcanzando una velocidad máxima de 84 km/h (45 kts) con un alcance de 33 km (18 NM) [87]. Sus menores dimensiones y la sencillez de su estación de control habilitan el embarque de esta aeronave tripulada remotamente en buques de poca envergadura como un cazaminas. Así mismo, la metodología de aterrizaje y despegue que emplea sigue los mismos principios que emplea el *Scan Eagle*, pero de montaje, empleo y emplazamiento más sencillos como se puede ver en la Figura 2-27.

## 3 DESARROLLO DEL TFG

Habiendo estudiado los diferentes aspectos del aprendizaje automático y la situación actual de los vehículos aéreos no tripulados, en este capítulo se desarrollará un sistema de localización y seguimiento de objetos flotantes en la mar, a través de visión por computador, desde una perspectiva aérea móvil. Para ello, se expondrán los recursos empleados en la concepción de este sistema, así como su empleo, y se plasmarán sobre este apartado, y de forma explicativa, la metodología y las distintas fases acaecidas para la creación del mismo.

### 3.1 Medios empleados

Para el desarrollo del modelo de aprendizaje se han empleado distintas herramientas seleccionadas en base a diferentes factores:

- Funcionalidad: que las herramientas cumplen con sus funciones deseadas es el factor principal que se ha tenido en cuenta en la selección.
- Soporte técnico: se ha comprobado que se pueden emplear con los medios disponibles.
- Integración: se ha estudiado que ciertas herramientas que así se requerían sean capaces de integrarse las unas con las otras para facilitar la transferencia de datos.
- Costo: se ha evaluado el coste de las diferentes opciones.
- Facilidad de empleo: lo intuitivo de la interfaz de cada herramienta es un factor que se ha tenido en cuenta para evitar que se requiera de mucho tiempo para aprender a utilizarla.
- Seguridad: se ha tenido en cuenta que las diferentes herramientas cumplan con protocolos de seguridad necesarios para proteger los datos y la información manejada.

En base a estos criterios, se han seleccionado y empleado los medios que se tratarán a continuación.

#### 3.1.1 Anaconda

*Anaconda* es una distribución de los lenguajes de programación *Python* y *R* que los instala y permite la instalación también de multitud de paquetes, aportando la capacidad de crear, cargar y guardar entornos diferenciados entre los que poder trabajar con facilidad sin verse estos afectados los unos por los otros [88].

Para el desarrollo de este Trabajo de Fin de Grado, se ha empleado la versión limitada de *Anaconda*, denominada *Miniconda*. Esta versión únicamente incluye, por defecto, los paquetes básicos: `python`, `conda`, `pip` y algún otro paquete que puede resultar práctico como `zlib`.

Durante el desarrollo de este TFG se van a crear dos entornos de trabajo diferentes. Uno para dar soporte al *software* de etiquetado (*Label Studio*) y otro para el soporte del sistema de entrenamiento e inferencia basada en modelos del algoritmo YOLOv8.

A continuación, se resume el conjunto de utilidades que han sido empleadas para el desarrollo de este TFG.

### 3.1.2 *Label Studio*

*Label Studio* es una herramienta de etiquetado de datos para el aprendizaje automático y la creación de conjuntos organizados y estructurados de estos. Esta plataforma de empleo en línea permite a los usuarios etiquetar, clasificar y verificar datos en una interfaz intuitiva, estando diseñada para ser fácil de usar y permitiendo crear etiquetas personalizadas, asignar tareas a diferentes usuarios y verificar la precisión de las etiquetas [89]. *Label Studio* también integra una API (del inglés *Application Programming Interface*) que conlleva una comunicación sencilla con diferentes plataformas y servicios de software, lo que permite integrar fácilmente los datos etiquetados en sus propios flujos de trabajo de aprendizaje automático. Dentro de los múltiples tipos de datos que se pueden estructurar en esta aplicación, está la posibilidad de etiquetar videos y fotos, que será en lo que se focalizará este TFG.

### 3.1.3 *Ultralytics HUB*

*Ultralytics HUB* es una plataforma web para la gestión de datos y el entrenamiento de modelos de aprendizaje profundo basados en el algoritmo YOLOv5 (en el momento de creación de este TFG). Esta es una plataforma de gran versatilidad que permite el desarrollo de modelos de aprendizaje, aportando la capacidad de crearlos y entrenarlos para reconocer e identificar patrones a partir de los datos introducidos. Así mismo, se puede llevar a cabo la validación misma del modelo en la herramienta, donde se permite analizar la efectividad y los resultados del modelo.

Este medio es idóneo para el desarrollo del modelo de detección de objetos flotantes en la mar, ya que permite importar de forma directa los datos desarrollados mediante *Label Studio* y trabajar a partir de ellos. También es un aspecto positivo el hecho de que se pueda enseñar al sistema sin necesidad de crear y ejecutar líneas de código, ya que la propia herramienta se encarga de aplicar el algoritmo de aprendizaje ya desarrollado que se emplea para el entrenamiento del modelo. Mediante una API, al igual que *Label Studio*, se puede hacer uso de aplicaciones como *Google Colab* para cargar con el peso computacional del entrenamiento del modelo.

### 3.1.4 *Google Colab*

*Google Colab* o *Google Colaboratory* es un servicio web proporcionado por *Google* cuyo fin es la investigación, centrado en el desarrollo del campo del aprendizaje automático. Esta herramienta permite a los usuarios escribir y ejecutar código en un entorno de *Jupyter Notebook* en la nube en lenguajes como *Python* y proporcionando acceso a recursos de hardware de gran potencia computacional, incluyendo GPU y TPU. Será la herramienta que se empleará en este trabajo para soportar las cargas más pesadas de computación.

No obstante, el acceso libre a esta capacidad computacional está limitado. Por ello, a lo largo del trabajo se planteará la adquisición de un pase que permita mayor capacidad de procesamiento en *Colab*.

### 3.1.5 *Google Drive*

*Google Drive* es una herramienta que ofrece un servicio de almacenamiento en la nube gestionado por *Google*. Esta aplicación permite a los usuarios almacenar, sincronizar y compartir archivos y carpetas en línea desde cualquier dispositivo con acceso a internet.

Para este trabajo, se empleará para el almacenamiento de datos y directorios. Así mismo, aquí se encuentran depositados una serie de videos que servirán de ayuda visual durante el desarrollo del TFG, y cuyo acceso se facilitará en los apartados pertinentes.

### 3.1.6 Codium

*Codium* es una herramienta libre de edición y ejecución de código. Es compatible con múltiples lenguajes de programación, incluido *Python* que será el que se empleará en este TFG. Integra características como el autocompletado de código o el resaltado de sintaxis, con multitud de extensiones que aportan funcionalidad a la aplicación. En este trabajo, se empleará esta herramienta para la ejecución de los *scripts* anexados. Estos son textos de código que tienen como fin, entre otras cosas, entrenar, aplicar y analizar el modelo de aprendizaje a desarrollar.

### 3.1.7 Coral USB Accelerator

*Coral USB Accelerator* es una plataforma desarrollada por *Coral AI* de *Google* que basa su empleo en el despliegue de modelos de aprendizaje automático en dispositivos, como cámaras y sensores. Este sistema permite a los desarrolladores la integración y empleo de sus modelos de aprendizaje en estos equipos y su ejecución y funcionamiento en tiempo real, pudiendo reaccionar sin la necesidad de una conexión a Internet. Esta herramienta es de gran utilidad y practicidad, ya que permite la implementación de sistemas de *Machine Learning* para aplicaciones como la visión por computadora en multitud de equipos, como por ejemplo un dron, mediante una interfaz USB 3.0, compatible con puertos USB 2.0. *Coral* permite, así mismo, flexibilidad y ergonomía en su implementación, ya que aun siendo una herramienta de gran capacidad, el tamaño del equipo es significativamente mínimo [90].

## 3.2 Desarrollo de sistema de identificación de objetos desde una posición fija

Para desarrollar el sistema de localización y seguimiento de objetos flotantes en la mar a través de visión por computadora desde una perspectiva aérea móvil, lo primero que se ha realizado es el desarrollo de un modelo independiente de aprendizaje automático de mayor simplicidad. Esta fase inicial, al igual que el sistema final que se construirá más adelante, basa su funcionamiento en el algoritmo de aprendizaje YOLO, el cual se ha estudiado previamente. La función de este primer estudio es la familiarización con la metodología a aplicar en la fase final del trabajo, al igual que el estudio de su funcionamiento, mediante el desarrollo de un sistema simplificado. Focalizada en la detección de buques civiles en la mar, esta etapa busca la construcción de un modelo menos complejo que el que se desarrollará más adelante, pero que sirva para la comprensión de la teoría del funcionamiento de los sistemas de aprendizaje profundo, así como su aplicación práctica.

El fin práctico generalizado de este sistema es poder identificar que objetos son los que aparecen en una fotografía, frente al fin que busca la segunda fase práctica del trabajo, que es identificar y seguir a los objetos que aparecen en un video. Para alcanzar este objetivo inicial, se ha entrenado al modelo mediante la alimentación de una fuente de datos constituida por imágenes a las que se le han añadido información complementaria. Estos datos adicionales consisten en los generados al identificar y clasificar de forma manual los diferentes elementos de relevancia que aparecen en cada fotografía. Al llevar a cabo este proceso, a las imágenes se les dota de una información añadida que será sobre la cual se basará el aprendizaje del modelo para llegar a ser capaz, de forma autónoma, de identificar y clasificar los distintos objetos presentes en cada fotograma.

El sistema trabajará sobre esta base de datos, analizando la información y construyendo patrones a partir de ella. De este modo, al procesar todas las imágenes aportadas con sus distintos etiquetados, el modelo alcanzará un nivel de entrenamiento tal que, al aportarle una nueva imagen, será capaz de clasificar sus componentes con un grado de certeza variable. La variabilidad de este nivel de confianza depende de multitud de factores, pero entre ellos, en esta fase cobran especial importancia la cantidad y lo surtido de la fuente de datos, y la imagen que se le presente para procesar una vez ya entrenado el sistema.

### 3.2.1 Obtención de datos

El primer escalón de esta fase ha sido la obtención de una base sobre la cual se alimentará el modelo de aprendizaje. El factor de mayor importancia que se ha tenido en cuenta a la hora de plantear este problema inicial es el de los datos a etiquetar. Con el objetivo de simplificar este primer acercamiento a la creación de modelos de visión por computador, se ha escogido un problema más sencillo de tratar frente al modelo desarrollado en los apartados ulteriores que tendrá que gestionar aspectos más problemáticos, como el movimiento inherente del dron que captura los vídeos. Por ello, se ha buscado que los datos cumplan con una serie de propiedades:

- Tipología: los datos empleados son exclusivamente fotografías. El empleo de videos conllevaría una dificultad añadida debido a su mayor complejidad de procesamiento y etiquetado.
- Perspectiva: la perspectiva de las fotografías es fija y constante durante toda la trama de datos. Esto ayuda al entrenamiento del sistema a la hora de reconocer e identificar patrones, por lo que una variación en la perspectiva puede confundir al aprendizaje. Este factor ha limitado el abanico de posibles fuentes de datos a las fotografías obtenidas en tierra ya que la capacidad de mantenerse en una misma posición, durante un periodo de tiempo suficiente, en medios navales o aéreos es exigua.
- Calidad: Tanto la calidad técnica de las fotografías como la calidad del contenido de estas es favorable al aprendizaje del sistema. Cuanta más calidad técnica tienen las imágenes, mayor facilidad hay para diferenciar los objetos en estas. Así mismo, la calidad del contenido de estas es de principal consideración; se busca que haya una gran cantidad y variedad de buques en los fotogramas para poder enseñar al sistema a diferenciar adecuadamente los distintos tipos de embarcaciones. También que las condiciones de iluminación y claridad de estas sean propicias para la identificación, pero así mismo variadas para ejercitar al sistema en diferentes escenarios meteorológicos.

Estos criterios son los que han conllevado la elección del empleo de cámaras web portuarias para la obtención de los datos, que tras el análisis de diferentes bases de datos de cámaras de distintas localidades se ha decidido la utilización de las fotografías tomadas por la cámara web del puerto pontevedrés de Cangas de Morrazo. A través de la página web de MeteoGalicia [91] se han descargado las fotografías del tráfico marítimo observado durante dos días, y de entre el banco de imágenes obtenido, se han seleccionado las que se consideraban más idóneas para realizar el entrenamiento del modelo. Como se puede ver en la Figura 3-1, estas cumplen con las características buscadas, debido a que su gran y constante afluencia de barcos genera una variedad de objetos a localizar que facilita el proceso de etiquetado y aprendizaje posterior.

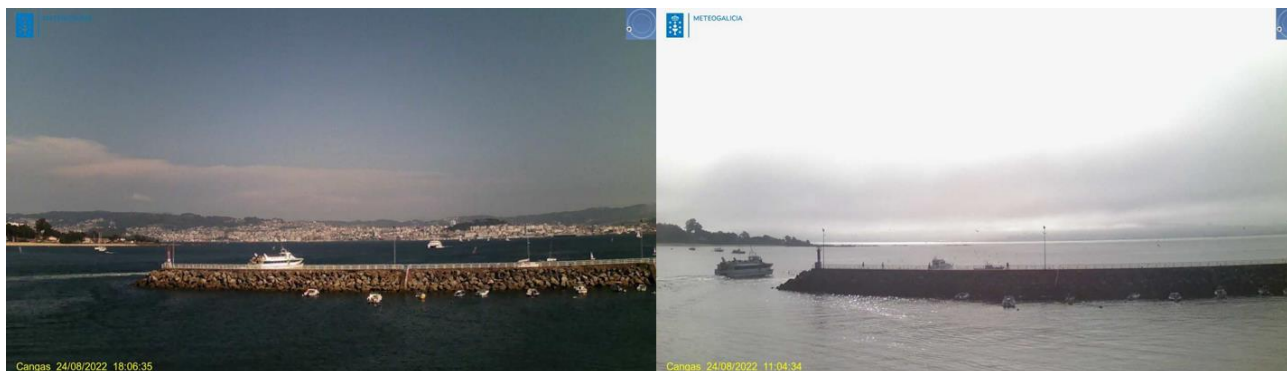


**Figura 3-1 Imágenes empleadas del puerto de Cangas de Morrazo [91]**

Debido a su gran y constante afluencia de barcos la variedad de objetos a analizar es satisfactoria. Tanto embarcaciones deportivas de un gran abanico de tamaños, como buques de pasajeros de diferentes portes y barcos pesqueros, constituyen el tráfico de entrada y salida de la dársena de Cangas de Morrazo,

así como de la navegación por las inmediaciones del puerto. También a la flota local se le añaden los buques de gran envergadura que recalán en el puerto de Vigo, generalmente mercantes y grandes cruceros de pasajeros, que se muestran distantes en un número de fotografías en la orilla de la ría opuesta a Cangas de Morrazo. A esto se le ha de sumar la calidad técnica de las fotografías, que ofrece suficiente precisión como para diferenciar los distintos tipos de embarcaciones en la práctica totalidad de las situaciones, y la variedad de las condiciones de iluminación, que obliga al modelo a identificar no solo patrones de colores sino también de formas y siluetas. Con todo, esta cámara portuaria fija cuyo principal fin es la monitorización y el acceso público al estado de la mar y la situación meteorológica en el puerto de Cangas de Morrazo, reúne las condiciones buscadas para su empleo como fuente de datos para esta fase práctica.

### 3.2.2 Etiquetado

Una vez con los datos a emplear ya seleccionados y obtenidos, se ha pasado a su etiquetado mediante la herramienta *Label Studio* ejecutada desde *Miniconda*. El etiquetado es la acción de identificar, clasificar, e indicar dónde están, los diferentes elementos de importancia en una imagen. *Label Studio* permite hacer esto mediante la creación y superposición de recuadros, de tamaño, orientación y color variable, sobre los objetos a procesar. Para el etiquetado, lo primero que se ha llevado a cabo es la denominación y creación de las etiquetas (*labels*), las cuales se han establecido en función de los distintos objetos flotantes que aparecen en la mar en las fotografías analizadas. Se han diferenciado los distintos tipos de embarcaciones y de objetos presentes en las imágenes disponibles, como por ejemplo los nadadores. Se ha denominado *boat* a las embarcaciones de menor porte, *ship* a los barcos de porte medio, y *vessel* a los buques de mayor porte y envergadura. También se ha tenido en cuenta el texto que aparece en la esquina inferior izquierda de todas las imágenes y que indica la hora de cada una, siendo diferente en todas ellas. Esto se ha hecho con el fin de enseñar al sistema a leer la hora de las fotografías.

*Label Studio* permite la introducción del código que define las etiquetas a emplear, así como las características de cada una. A continuación, se muestra el código empleado para el etiquetado de objetos en este sistema:

```
<View>
  <Image name="image" value="$image"/>
  <Number name="hour" toName="image"/>
  <RectangleLabels name="label" toName="image">
    <Label value="image_info" background="yellow"/>
    <Label value="fish_boat" background="green"/>
    <Label value="speed_boat" background="red"/>
    <Label value="sailing_boat" background="blue"/>
    <Label value="fish_ship" background="olive"/>
    <Label value="passenger_ship" background="purple"/>
    <Label value="sailing_ship" background="cyan"/>
    <Label value="passenger_vessel" background="white"/>
    <Label value="merchant_vessel" background="brown"/>
    <Label value="jet_ski" background="black"/>
    <Label value="surf" background="black"/>
    <Label value="person" background="black"/>
  </RectangleLabels>
</View>
```

Con las etiquetas definidas, lo siguiente que se ha realizado es importar las imágenes a la aplicación web. Mediante una función específica, la herramienta permite que se suban todos los archivos en su conjunto de forma directa, evitando tener que preprocesarlos o subirlos uno a uno.

Ya con todo establecido para el etiquetado, este se lleva a cabo. Todas las imágenes se analizan detenidamente de forma personal para discernir que objetos flotantes se encuentran en ellas. Se identifica y se indica cada tipo de embarcación mediante recuadros de diferentes colores como se puede ver en la Figura 3-2, y cuyo significado se ha indicado previamente en el código de etiquetado. Se busca el máximo ajuste del recuadro al barco para evitar que, al abarcar la selección un área que no corresponde al barco, se pierda calidad de identificación en el proceso de entrenamiento. Así mismo, se requiere que la clasificación del objeto sea de la mayor precisión, evitando clasificar erróneamente los distintos tipos de barcos, para no confundir al sistema y crearle una contradicción y conflicto de fuentes de aprendizaje, lo que puede llevar a un descenso significativo de la futura fiabilidad de clasificación de objetos del sistema.

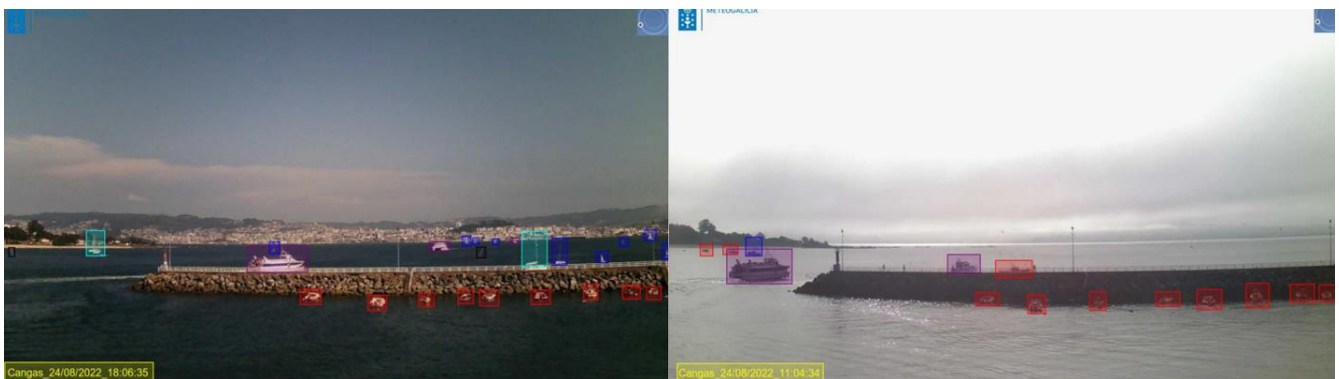


**Figura 3-2 Imágenes etiquetadas de Cangas [elaboración propia]**

Para esta fase, se ha llevado a cabo el etiquetado de 150 imágenes del puerto de Cangas. Por otro lado, el Alférez de Fragata D. Jesús Moreno Fernández, en el contexto de su TFG, titulado “Aplicación de técnicas de aprendizaje profundo a la localización y seguimiento de objetos en entornos marítimos empleando cámaras de video portátiles”, ha realizado el etiquetado de otras 150 fotografías que se han conjuntado en la etapa de entrenamiento para ampliar y mejorar el proceso de aprendizaje del sistema.

### 3.2.3 Entrenamiento

Una vez llevado a cabo el proceso de etiquetado, se dispone de 300 imágenes procesadas a partir de las cuales se ha generado una base de datos (*dataset*) de tamaño adecuado para realizar un primer entrenamiento. Tras el etiquetado, se han exportado los datos a la herramienta de *Ultralytics HUB*. Esta permite entrenar modelos de aprendizaje profundo basados en el algoritmo YOLOv5 y aporta resultados gráficos donde se aprecia la evolución del entrenamiento. Para importar los datos, *Label Studio* permite exportar el *dataset* en formato YOLO, un formato que genera para cada imagen un documento TXT en el que se indica que diferentes objetos hay en la imagen, sus coordenadas y sus dimensiones en alto y ancho.

Sin embargo, el formato empleado por *Label Studio* no es directamente aplicable, ya que no distingue entre las partes de entrenamiento, validación y testeo que suele emplearse en los procesos de entrenamiento y validación automática de resultados. Es por ello, que se ha empleado un *script* (Anexo IV: Script de organización de imágenes) en código *Python* que se encarga de generar esta estructura de datos a partir de las etiquetas generadas en el programa *Label Studio*, es decir las divide en: las empleadas para entrenar al sistema (*Train*), las que usa para validar los resultados (*Validation*) y las que nunca emplea y que servirán para comprobar los resultados en caso de requerirlo (*Test*).

Con la base de datos ya en *Ultralytics*, se establecen los parámetros para el entrenamiento del modelo. Las proporciones asignadas a cada clase de imágenes son, por lo general, del 60% para *Train*,

del 20% para *Validation* e igualmente del 20% para *Test*, como se puede ver en la Figura 3-3. Por otro lado, se determinan las *epochs* o épocas a ejecutar. Cada una representa el número de pasadas a todo el *dataset* que realizará el modelo en su entrenamiento, procesando y actualizando sus mediciones en cada pasada y mejorando así su precisión y rendimiento. En este entrenamiento se han establecido 100 *epochs*; de forma que cuanto mayor sea su número, mayor será el tiempo de procesado y la capacidad computacional requerida. Así mismo, cuanto menor sea, menor será la precisión. Por tanto, hay que encontrar un número que alcance un equilibrio entre ambos factores.

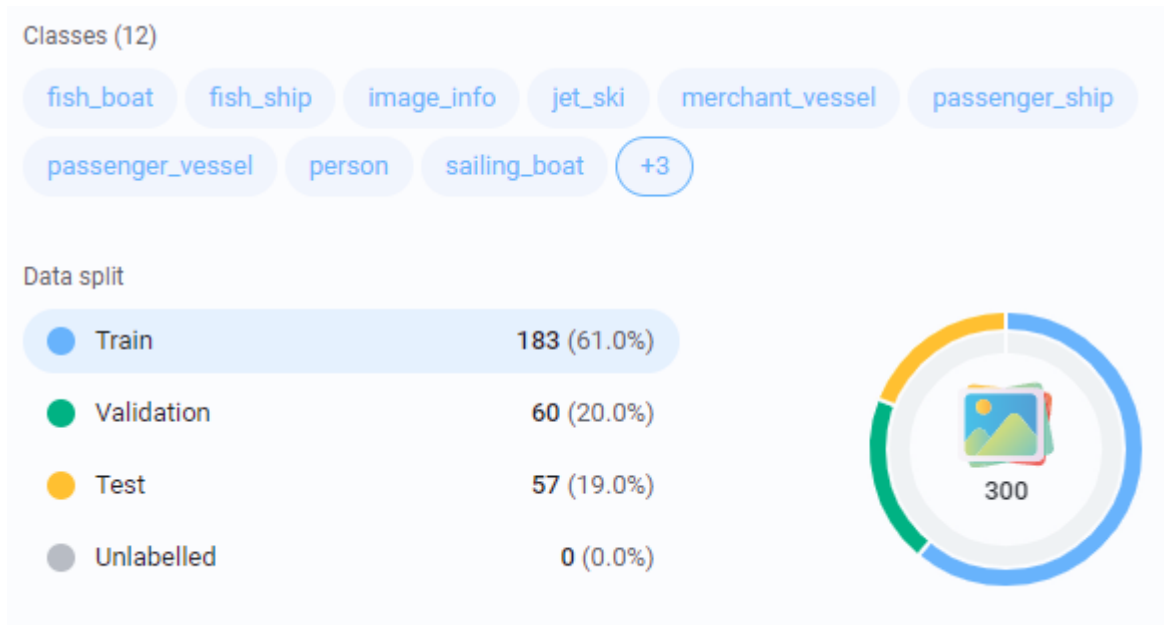


**Figura 3-3 Distribución de imágenes del *dataset* en *Ultralytics* [fuente: *Ultralytics*]**

Una vez configurados los aspectos relevantes en *Ultralytics* se hace uso de *Google Colab*, mediante una API key (clave de API), para dar soporte al entrenamiento. Esto se hace ya que, sin su empleo, el entrenamiento se haría en local, es decir, se emplearían los medios del equipo sobre el que se esté trabajando con *Ultralytics*. En el caso de este primer modelo, aunque se dispone de suficiente capacidad de computación como para entrenarlo en local, se busca la familiarización con los procedimientos que se emplearán en el modelo a desarrollar en los siguientes apartados. Si bien los primeros entrenamientos del siguiente modelo se realizarán en local, el último entrenamiento, que requerirá de una gran potencia de procesamiento, se realizará en *Colab*, por ello se emplea en este método de trabajo.

Tras esto, se ejecuta en *Colab* el entrenamiento del modelo con el algoritmo YOLOv5s. A la finalización de esta tarea, se guardan los resultados en *Ultralytics* para su análisis, para lo cual, se aplica este a una de las imágenes de *Test* reservadas para el análisis de la precisión del modelo y que no han tenido parte en su entrenamiento. Como se puede ver en la Figura 3-4, el sistema es capaz de identificar un gran número de embarcaciones. No obstante, la fiabilidad que tiene el modelo de que cada identidad sea la correcta es significativamente baja, demostrando que el sistema no es lo suficientemente fiable. Así, a alguna de las lanchas rápidas (*speed\_boat*) apenas alcanza a reconocerla con un 26% de fiabilidad, y al rótulo de información de la imagen, presente por igual en todas las fotografías que han nutrido al modelo en su entrenamiento, tan solo acierta a darle un 88%.

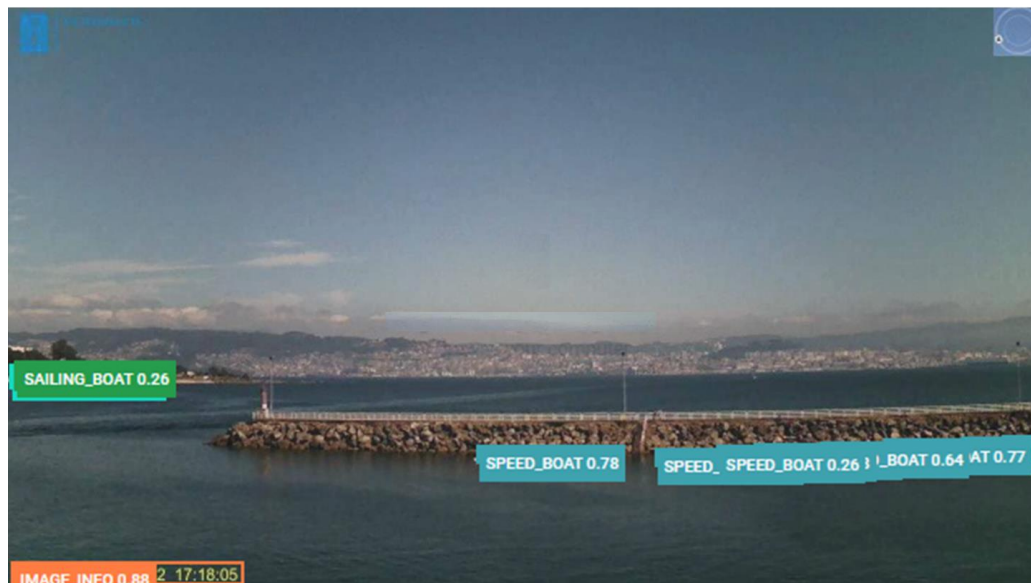


Figura 3-4 Puerto de Cangas analizado por el modelo [elaboración propia]

Por otro lado, *Ultralytics* nos proporciona una serie de gráficas que representan la evolución del modelo a lo largo de su entrenamiento. La primera gráfica, la Figura 3-5, representa las métricas empleadas para analizar el desempeño del modelo durante el entrenamiento. Aquí se representan los siguientes valores y su comportamiento a lo largo de las 100 *epochs* en que se ha ejecutado el entrenamiento:

- **mAP 0,5.** El valor mAP (*Mean Average Precision*) representa la media de las precisiones obtenidas en diferentes umbrales de IoU (del inglés *Intersection over Union* o, en castellano, Intersección sobre Unión) para cada clase de objeto en el conjunto de datos. El IoU es una medida que cuantifica la similitud entre la caja delimitadora predicha por el modelo y la caja real de la imagen. Este valor se calcula dividiendo el área de intersección entre las cajas delimitadoras por el área de unión entre las mismas, y devuelve un valor entre 0 y 1 que indica el grado de solapamiento entre ellas. Un IoU de 1 significa que las cajas delimitadoras coinciden por completo, mientras que un IoU de 0 indica que no hay solapamiento. Por tanto, cuanto mayor sea el IoU, mejor coincidencia habrá entre la estimación del modelo y la realidad. En este caso, el valor mAP 0,5 indica el promedio de las precisiones medias calculadas para umbrales de IoU del 50% para cada clase de objeto detectado.
- **mAP 0,5:0,95.** Aquí se toma en cuenta el IoU desde 0,5 hasta 0,95, en incrementos de 0,05. Por lo tanto, este valor indica la precisión media del modelo, entre las predicciones y la realidad, en diferentes umbrales de solapamiento.
- **Precision.** El valor *Precision* indica la proporción de detecciones positivas que son verdaderas. Esto es, la relación de verdaderos positivos con el número total de objetos detectados por el modelo. Mide, por tanto, la precisión del modelo en detectar únicamente los objetos de interés y evitar falsos positivos. Se calcula como el número de verdaderos positivos dividido entre la suma de verdaderos y falsos positivos.
- **Recall.** Este mide la capacidad de un modelo para identificar todos los verdaderos positivos en un conjunto de datos. Es decir, cuántos de los objetos que realmente existen en la imagen el modelo ha sido capaz de detectar y clasificar correctamente como positivos. Se calcula dividiendo el número de verdaderos positivos por la suma de verdaderos positivos y falsos negativos.

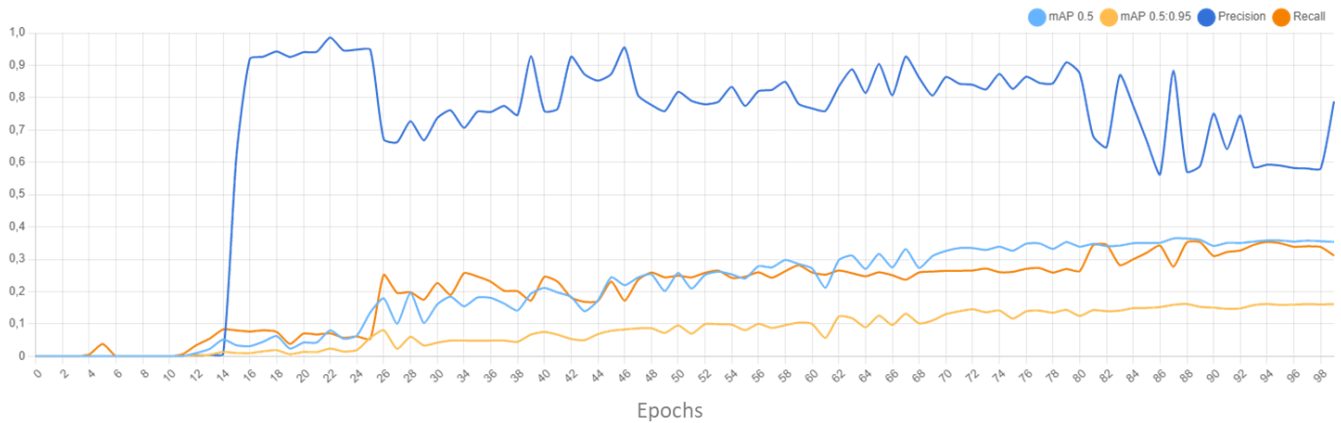


Figura 3-5 Métrica del modelo [elaboración propia]

Analizando la Figura 3-5, se puede observar como ambos valores de mAP se ven incrementados paulatinamente a medida que el modelo sigue con su entrenamiento, procesando en cada *epoch* todo el *dataset* y aprendiendo a reconocer los diferentes patrones. Se puede deducir que, si prolongado el entrenamiento lo suficiente, se podrían alcanzar unos valores de mAP cercanos a la unidad, lo que conllevaría un buen desempeño del modelo. También se aprecia cómo, siguiendo la tendencia de los mAP, va aumentando el valor del *Recall*. Esto significa que, en las primeras pasadas, el modelo pasaba por alto la gran mayoría de los objetos relevantes de las imágenes. A medida que se prolonga el proceso de aprendizaje, va aumentando su capacidad de detectar los diferentes objetos en las fotografías. Por otra parte, destacan los altos valores que alcanza el parámetro de *Precision*, que, si bien tiene muchos picos, se mantiene en un rango de valores aceptable. Esto significa que, de los elementos que el sistema sí que ha sido capaz de detectar, pocos de ellos no eran elementos relevantes. No obstante, se aprecia una sutil tendencia a la baja. Esta se debe a que, según aumenta el número de objetos detectados debido a la mejoría del *Recall*, le resulta más complicado al modelo diferenciarlos entre relevantes e irrelevantes. Con un número de *epochs* lo suficientemente elevado y una base de datos que lo permita, tras alcanzar un equilibrio entre *Precision* y *Recall*, procederían ambos parámetros a adquirir una tendencia de aumento.

La siguiente gráfica se puede apreciar en la Figura 3-6 y trata el parámetro de *Box Loss*. Este representa la diferencia entre las *bounding boxes* generadas por el modelo, y las cajas delimitadoras reales. A diferencia de los valores IoU, *Box Loss* no estudia la superposición de las cajas, sino la diferencia entre sus coordenadas. En este caso, la gráfica muestra cómo la diferencia, entre la ubicación de la caja predicha y ubicación real, disminuye a medida que el modelo se entrena. Una disminución en *Box Loss* indica que el modelo está aprendiendo a predecir con mayor precisión la ubicación de los objetos.

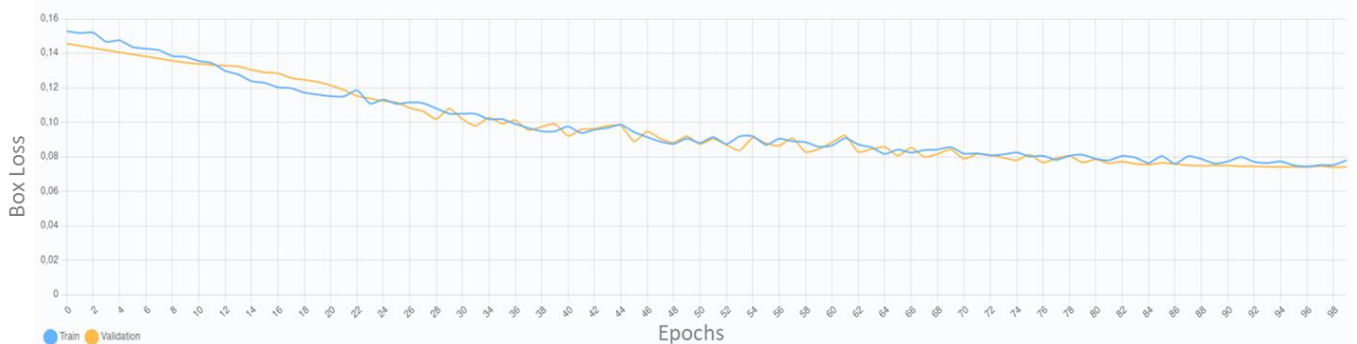


Figura 3-6 Box Loss [elaboración propia]

La Figura 3-7 es la gráfica de *Class Loss*, y muestra la evolución del error del modelo en la clasificación de los objetos durante el entrenamiento. Se puede apreciar cómo el modelo está aprendiendo a clasificar correctamente los diferentes elementos detectados en las imágenes. Un valor alto de *Class Loss* indica que el modelo está teniendo dificultades en la clasificación de los objetos, mientras que un valor bajo representa una capacidad de clasificación más desarrollada.

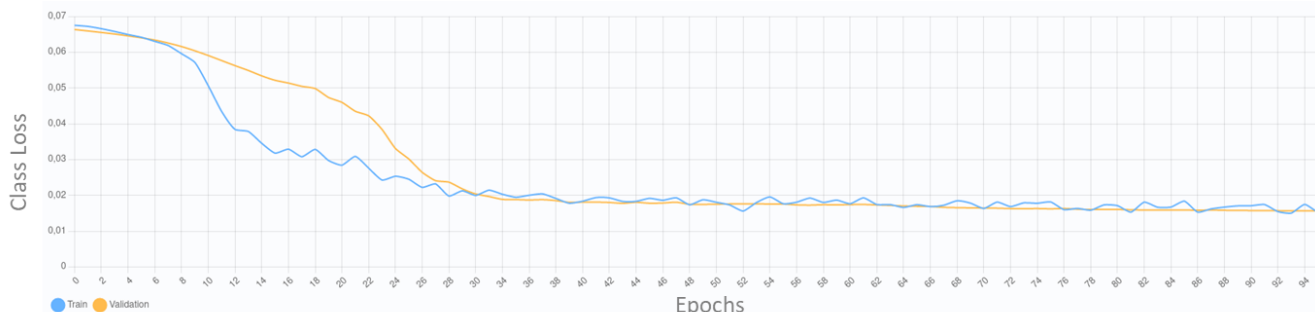


**Figura 3-7 Class Loss [elaboración propia]**

La última gráfica que genera *Ultralytics* es la de *Object Loss* o pérdida de objetos. Esta, que se puede ver en la Figura 3-8, es una representación de la diferencia entre las etiquetas de los objetos en una imagen y las predicciones del modelo para la misma imagen. La disminución en el valor de pérdida indica que el modelo está mejorando su capacidad para detectar objetos en las imágenes. En el caso de que la pérdida aumentase, esto podría conllevar que el modelo se estuviese sobreajustando a los datos de entrenamiento, lo que significaría que en lugar de aprender patrones generales estaría memorizando los datos.

Además de la tendencia general, que en este caso es negativa, es importante prestar atención a los picos en la gráfica, que indican que el modelo está teniendo dificultades para detectar objetos específicos o en ciertas condiciones. A lo largo del entrenamiento, si bien no de forma muy pronunciada, se ve cómo se va reduciendo la frecuencia en que aparecen los picos, por lo que se puede deducir que cada vez le cuesta algo menos identificar los objetos complicados de detectar. No obstante, siguen estando presentes en toda la línea temporal del entrenamiento, por lo que el modelo no ha sido capaz de resolver esa problemática.

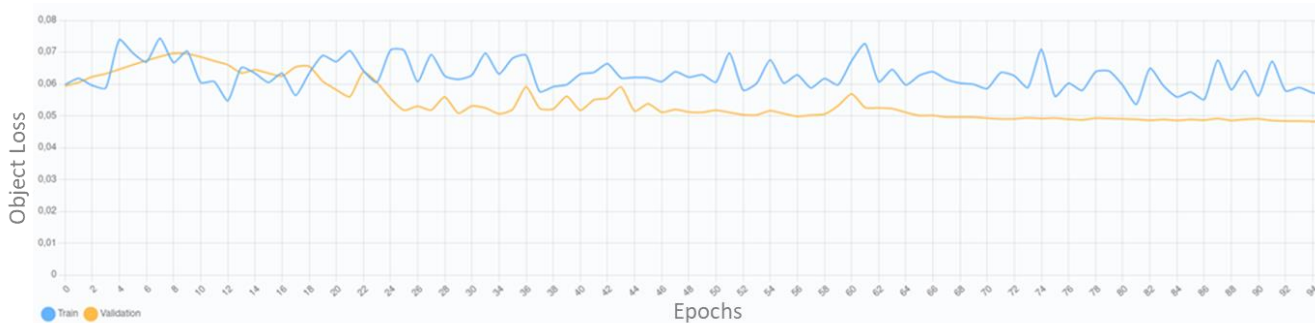


**Figura 3-8 Object Loss [elaboración propia]**

Al igual que con las otras gráficas de pérdida, se busca minimizar el valor del *Object Loss* para obtener un modelo de detección e identificación de objetos preciso y robusto. Para alcanzar una mejora considerable del sistema, se debería ampliar la base de datos con imágenes nuevas de diferentes días, que aportasen entornos de iluminación distintos y más variados.

### **3.3 Desarrollo de sistema de identificación de objetos desde una perspectiva móvil aérea**

El modelo de análisis de fotografías de una cámara fija desde una posición en tierra ha servido para introducir y elaborar el desarrollo y el funcionamiento de un sistema de aprendizaje profundo. Este sistema inicial, si bien complejo en su esencia, ha tenido por precepto el evitar complicaciones excesivas en su entrenamiento, como por ejemplo la variación en la perspectiva de las imágenes. La prioridad del desarrollo ha sido el correcto funcionamiento del modelo frente a la capacidad de su aplicación en escenarios de naturaleza variada y cambiante. No obstante, una vez habiendo alcanzado un nivel de elaboración satisfactorio de un sistema de aprendizaje profundo, se busca en esta fase la construcción de un modelo que, en lugar de rehusar, busque aplicar todas esas complicaciones para poder ser empleado en situaciones dispares. Partiendo de esta premisa, en este apartado se busca la localización y seguimiento de objetos flotantes en la mar desde una perspectiva aérea móvil.

Al contrario que en la fase anterior, donde los buques analizados han sido de índole civil, en esta parte práctica se focalizará la aplicación del sistema a la clasificación de embarcaciones militares, si bien también se entrenará al modelo para identificar barcos de recreo, de pasaje, mercantes y pesqueros. Así mismo, gracias a la ventaja táctica que supone un medio aéreo como un dron de pequeño porte, al poder posicionarse cercano a otra unidad militar evitando su detección, se busca en este apartado el aprovechamiento de esta situación; se persigue enseñar al sistema de aprendizaje a identificar y discernir los distintos equipos que porta cada buque militar de forma visible. Aunque limitada esta capacidad debido al pequeño tamaño de algunos equipos, se busca la clasificación de aquellos cuyo tamaño les permite ser fácilmente reconocibles y cuya presencia puede resultar relevante, proporcionando su identificación inteligencia naval de importancia.

Además, en contraposición al modelo anterior, este sistema no busca exclusivamente la identificación de los distintos elementos de relevancia en una fotografía. Este tiene por objeto el poder analizar imágenes de video y, por lo tanto, ser capaz de identificar un objeto de forma constante a lo largo de toda la secuencia, es decir, realizar un seguimiento del buque y sus equipos durante todo el video. De esta forma, este nuevo sistema aumenta su capacidad de aplicación.

En este apartado, se proporcionan enlaces web para la visualización de algunos videos como ayuda al trabajo. Debido a la naturaleza privada de este TFG, y la consideración de dichos videos como parte integral del mismo, no se contempla la publicación de estos videos en otros medios como *Youtube* o *Vimeo*, así mismo, tampoco se busca su difusión fuera del ámbito de este trabajo. En caso de que los videos se mostrasen borrosos o con una calidad deficiente, es recomendable descargar los archivos para poder visualizarlos con la calidad adecuada.

#### **3.3.1 Obtención de datos**

Como se planteó en la introducción de este TFG, la capacidad de aportar a buques una visión aérea de su entorno es una ventaja de gran peso para el desempeño de sus tareas y navegaciones. Por esto, en este nuevo sistema de identificación, una característica de importancia es que las imágenes a analizar han de ser tomadas desde un medio aéreo cuya posición no sea constante a lo largo de cada secuencia analizada; la aeronave que toma las imágenes se ha de ir moviendo mientras las capta para reflejar con mayor exactitud un hipotético empleo práctico del sistema.

Habiendo efectuado una primera prueba de aprendizaje reflejada en los apartados precedentes, se ha buscado un conjunto de datos que, si bien sí cumplan con los requisitos de calidad contemplados en la fase anterior, han de diferir en los demás aspectos tenidos en cuenta para seleccionar la fuente de datos. La perspectiva, como ya se ha comentado, ha de ser diferente con el fin de integrar la ventaja aérea que se busca en este trabajo. Así mismo, se ha buscado imágenes que se centran en buques de marinas de guerra y que, al mismo tiempo, permiten divisar de forma clara y sin ambigüedades los distintos equipamientos y plataformas de relevancia que les aportan capacidades navales de importancia, como

por ejemplo un cañón. Respecto a la tipología, se han buscado fuentes de video y no fotografías, ya que al haber comprobado el correcto funcionamiento con las fotografías en el apartado anterior se ha de pasar al siguiente escalón que son los videos y que es también el tipo de datos que aporta un dron convencional que pueda ser empleado con el fin que persigue este Trabajo de Fin de Grado.

Tras un proceso de búsqueda y adquisición de videos de diversas procedencias que se ajustan en mayor o menor medida a las características definidas, se ha adquirido un banco de videos de la parada naval que tuvo lugar en 2017 en Marín con motivo de la celebración del 300 Aniversario de la Real Compañía de Guardiamarinas. Estos videos han sido grabados desde multitud de ángulos aéreos, con distintas inclinaciones y perspectivas como se puede apreciar en la Figura 3-9, por un helicóptero AB-212 de la 3ª Escuadrilla de Aeronaves de la Armada. Habiendo obtenido una base de 90 GB de videos, se ha llevado a cabo un proceso de selección inicial para la generación de un banco variado de imágenes que no excedan los 100 MB de tamaño. Esta reducción del contenido se ha hecho para llevar a cabo un primer entrenamiento que no suponga una gran carga de computación y que sea fácilmente ampliable para entrenamientos posteriores del sistema. Dentro de este limitado *dataset* se ha buscado contenido que refleje la mayor variedad de unidades posible que tomaron parte en la demostración naval. No obstante, para conseguir un gran nivel de detalle de clasificación de los distintos equipos que portan las diferentes unidades, estas se han analizado de una forma más individualizada y desde una perspectiva cercana a la unidad.

Así, el primer etiquetado se ha realizado sobre 4 videos independientes, cada uno enfocado en una clase de buque distinta: el primero graba al buque portaaeronaves L-61, el segundo a un submarino, el siguiente a una fragata clase F-100 y el último al buque de aprovisionamiento A-14, siendo estos los principales tipos de medios navales que conforman las armadas actuales. Aun así, en estos primeros videos sigue habiendo presencia de embarcaciones de otros portes y clases, como corbetas, veleros o pesqueros, si bien, estos ocupan un segundo plano permitiendo así un mejor y más cercano encuadre a los equipos y siluetas de los buques a identificar.



**Figura 3-9 Diferentes ángulos y perspectivas en un video del *dataset* [fuente: Armada]**

Tras intentar exportar los videos escogidos a *Label Studio*, han surgido ciertos problemas que se han tenido que corregir. El primer problema resulta del formato en que se tienen los videos: estos están en formato “.MOV” y *Label Studio* solo tiene soporte para procesar en formato “.mp4”. Tras una simple operación de conversión de formato mediante el comando “ffmpeg”, se importan los videos a *Label Studio* y se hace una prueba de etiquetado. El siguiente problema surge al exportar estos videos fuera de la herramienta de etiquetado. Esta solo es capaz de exportar etiquetas, pero no de extraer las imágenes que son necesarias para el entrenamiento del modelo. Tras un análisis de las distintas rutas de acción, se decide que, habiendo ya operado con fotografías y constatado su buen funcionamiento en su empleo en el proceso de desarrollo de un modelo de aprendizaje, se extraigan de los videos fotogramas sobre los que se trabajará; se creará sobre fotografías el *dataset* con el que se entrenará al modelo, en lugar de sobre videos. El objetivo sigue siendo que el modelo sea capaz de analizar videos, si bien la forma en que se genere el etiquetado será la misma que la empleada en el caso de las imágenes. Respecto al problema que surge debido a esta solución, que es la dificultad de seguimiento de los objetos entre imágenes (más fácilmente ejecutable a lo largo de un video), se aplica para solventar esta situación el

algoritmo de *tracking*, integrado en YOLOv8, *BoT-SORT*, pudiéndose emplear también el algoritmo *ByteTrack*, como se estudió en el apartado de explicación del algoritmo YOLO.

Para extraer las imágenes de los videos, se ejecuta el *script* en *Python* del Anexo V: Script de extracción de imágenes.

### 3.3.2 Etiquetado

Una vez se han obtenido las imágenes de los videos, se siguen los mismos pasos que los ejecutados para el etiquetado del modelo anterior. Se han exportado a *Label Studio* las imágenes y se ha procedido a la creación y designación de las etiquetas. En este caso, como se busca la detección de un rango mayor de objetos, se ha diversificado la tipología de etiquetas creadas.

Al igual que el modelo de detección en tierra, se ha contemplado una gran variedad de clases de embarcaciones. No obstante, siendo los principales actores de los fotogramas buques militares, se ha establecido independientemente de su tamaño una clara distinción entre los tipos de embarcaciones; la denominación se ha ceñido, únicamente, a la designación funcional de los barcos, la cual en la mayoría de los casos está correlacionada con su porte. Por otro lado, se han añadido los diferentes equipos y elementos de relevancia a identificar en los barcos de la Armada.

Unas de las principales etiquetas que se han redactado han sido las de los elementos de los buques de mayor tamaño: las aeronaves. Presentes especialmente en el portaaviones, en él se pueden diferenciar dos tipos distintos, las de ala fija o aviones, y las de ala rotatoria o helicópteros; ambos tipos aportan capacidades diferentes y de gran importancia táctica. Estos elementos se encuentran en el entorno marino, si no en el aire, a bordo de un buque, estibados en el hangar de vuelo o en la cubierta de vuelo. El primero de estos sitios está a resguardo, pero el segundo se encuentra abierto al exterior, permitiendo a las cámaras de inteligencia como las que pueden montarse en un dron captar imágenes de las aeronaves. Por ello se ha buscado que el modelo aprenda a identificar tanto a helicópteros como a aviones en pista, al igual que a localizar las cubiertas y plataformas de vuelo en los buques, añadiendo una *label* para estos elementos. Así mismo, se han creado etiquetas para todo el armamento apreciable a media distancia, obviando aquello que requiere de una gran cercanía para su detección, que suelen ser las armas de menor calibre y relevancia táctica. En las imágenes analizadas se pueden apreciar 3 clases de lanzaderas de misiles: el lanzador vertical MK-41 de las fragatas clase F-100, del que solo se aprecia una plataforma rectangular sobre el castillo del buque, como se puede ver en la Figura 3-10, la lanzadera MK-13 de las fragatas clase F-80 situada también en el castillo, y las canastas de los misiles RGM-84 *Harpoon*, que en las F-80 se disparan desde el MK-13 pero en las F-100 lo hacen desde sus propias canastas fuera del MK-41.



Figura 3-10 MK-41 de una F-100 durante el lanzamiento de un ESSM [92]

Por su similitud estructural, que se puede apreciar en la Figura 3-11, tanto los *canisters* (canastas) del *Harpoon* en las F-100, como el MK-13 de las F-80, que en la imagen aparece armado con un misil *Harpoon*, se clasificarán mediante la etiqueta “*Missile\_Launcher*”, mientras que el MK-41 se clasificará con la etiqueta “*Vertical\_Launcher*”. “*Gun*” servirá para etiquetar a los cañones.

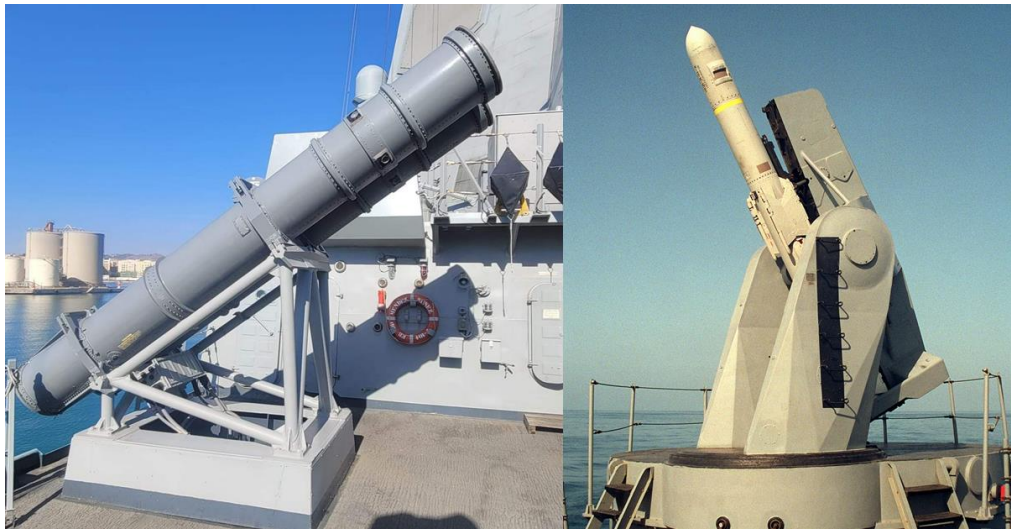


**Figura 3-11 Pareja de *Canisters* [93] frente a un lanzador MK-13 [94]**

El resto de elementos a etiquetar son las antenas de comunicaciones por satélite, cuya silueta suele ser siempre la misma ya que van protegidas por un radomo, y los distintos radares. Si bien hay multitud de tipologías de radares en los buques de guerra, para el entrenamiento de este modelo se ha limitado a tres grandes grupos generales: radar de navegación, radar aéreo y radar de control de tiro. Con todos los elementos tenidos en cuenta, se ha creado el siguiente código de *Label Studio* para el etiquetado:

```
<View>
  <Image name="image" value="$image"/>
  <Header value="RectangleLabels"/>
  <RectangleLabels name="label" toName="image">
    <Label value="Air_Search_Radar" background="rgba(218, 1, 238, 1)"/>
    <Label value="Aircraft_Carrier" background="rgba(0, 255, 0, 1)"/>
    <Label value="Corvette" background="rgba(1, 146, 243, 1)"/>
    <Label value="Fire_Control_Radar" background="rgba(241, 135, 0, 1)"/>
    <Label value="Fixed_Wing_Aircraft" background="rgba(72, 124, 73, 1)"/>
    <Label value="Flight_Deck" background="rgba(187, 1, 37, 1)"/>
    <Label value="Frigate" background="rgba(166, 250, 86, 1)"/>
    <Label value="Gun" background="rgba(178, 135, 216, 1)"/>
    <Label value="Missile_Launcher" background="rgba(0, 252, 159, 1)"/>
    <Label value="Navigational_Radar" background="rgba(116, 239, 254, 1)"/>
    <Label value="Replenishment_Ship" background="rgba(99, 56, 252, 1)"/>
    <Label value="Rotatory_Wing_Aircraft" background="rgba(252, 183, 127, 1)"/>
    <Label value="Satellite_Comms_Antenna" background="rgba(137, 176, 1, 1)"/>
    <Label value="Speed_Boat" background="rgba(95, 192, 153, 1)"/>
    <Label value="Submarine" background="rgba(1, 53, 87, 1)"/>
    <Label value="Vertical_Launcher" background="rgba(148, 86, 0, 1)"/>
  </RectangleLabels>
</View>
```

Los valores “rgba” que aparecen después de cada elemento identifican un color y tonalidad diferente para cada objeto. Estos colores han sido seleccionados de forma aleatoria por *Label Studio* ya que al haber un mayor número de etiquetas que en el modelo anterior, se ha requerido una mayor paleta para diferenciar los elementos en este etiquetado. Así mismo, esta herramienta permite que, si durante la fase de etiquetado de cada imagen, se reconoce un nuevo objeto no contemplado en el código de etiquetas inicial, se pueda modificar o introducir una nueva etiqueta en cualquier momento.

Una vez definidos los parámetros de etiquetado, se ha analizado y etiquetado cada imagen, identificando e indicando los diferentes elementos con la mayor precisión posible. Esto se puede ver al comparar las fotografías mostradas en la Figura 3-12, donde se aprecia una imagen sin etiquetar frente a la misma una vez etiquetada.

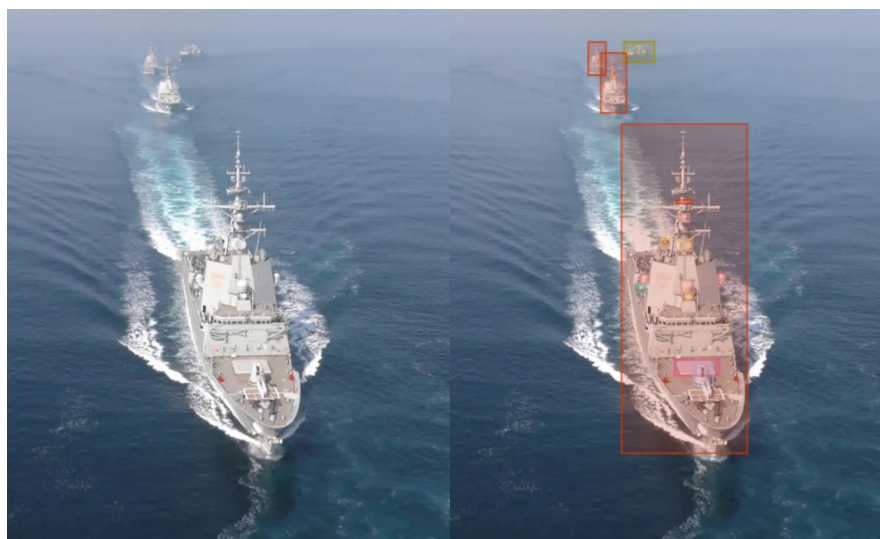


**Figura 3-12 Plano frontal de una F-100 antes y después del etiquetado [elaboración propia]**

Después de este etiquetado inicial, se exportan las imágenes a un ordenador con mayor capacidad computacional para que sobre ese se gestione el entrenamiento del sistema. A raíz de esa exportación surge una problemática menor: algunas de las etiquetas se encuentran desplazadas de sus objetos. Tras un análisis de las posibles causas, se determina que la razón ha sido la ausencia, por parte de *Label Studio*, de la capacidad de exportar en formato YOLO etiquetas cuyo eje de ordenadas no es perpendicular al plano horizontal de la imagen. En este proceso de etiquetado se han encuadrado objetos buscando ajustarse al máximo a sus límites, surgiendo en algunos casos como el visto en la Figura 3-13, la necesidad de rotar el eje de la caja delimitadora con el fin de abarcar menos espacio baldío.



**Figura 3-13 Imagen con etiquetas rotadas [elaboración propia]**

No habiendo surgido esta situación en el etiquetado del modelo de identificación desde una perspectiva fija, en este apartado ha supuesto la revisión del *dataset* generado y la edición y corrección de las imágenes afectadas, como se puede ver en la Figura 3-14, que es la corrección de la figura precedente.



**Figura 3-14 Imagen con etiquetas corregidas [elaboración propia]**

### 3.3.3 Entrenamiento

Una vez el *dataset* ha sido corregido, y cumple con los parámetros idóneos para el aprendizaje del modelo, se lleva a cabo el entrenamiento. Para este sistema, se emplea la última versión del algoritmo con el que se está trabajando, que es en el momento de redacción YOLOv8, operando inicialmente en su variante *nano*. Este algoritmo, sin embargo, al ser tan reciente su desarrollo, todavía no está integrado en la herramienta de *Ultralytics HUB*. Por ello, su aplicación se llevará a cabo mediante la ejecución de *scripts* a través de la herramienta *Codium*. Al ser el *dataset* de un tamaño moderado, 240 imágenes, no es necesario el empleo de *Google Colab*; en local se dispone de suficiente capacidad computacional como para ejecutar el entrenamiento, si bien este requerirá de un periodo considerable de tiempo de procesado.

El *script* empleado es el del Anexo VI: Script de entrenamiento del modelo, donde se parte de un modelo base y se le entrena con el *dataset* recién creado. Este modelo base es a partir del cual se construyen todos los sistemas de detección basados en el algoritmo YOLOv8. Es un sistema básico que ha sido generado mediante un preentrenamiento con un *dataset* genérico de procesamiento de imágenes, denominado *dataset COCO (Common Objects in Context)*. El empleo de este modelo como cimiento para el desarrollo del modelo generado en este TFG permite una velocidad de entrenamiento y capacidad de aprendizaje mayor. Empero, este preentrenamiento no constituye una base práctica para la detección de objetos flotantes en la mar, únicamente potencia la capacidad de procesado del modelo en sus primeras fases de detección de patrones simples.

El *script*, así mismo, ejecuta el algoritmo *BoT-SORT*, pudiéndose emplear también el algoritmo *ByteTrack* que genera los mismos resultados; es decir, permiten realizar el *tracking* de todos los objetos localizados en los videos. Así, el objetivo es que el modelo sea capaz de operar de forma que no confunda entre sí elementos parecidos que, debido a su cinemática relativa durante el video, puedan acabar ocluyéndose, o pudiesen resultar difíciles de seguir y mantener diferenciados.

Se ejecuta por tanto el *script* en *Codium*, y comienza el entrenamiento del modelo. Se fijan un total de 1000 *epochs*, pero, no obstante, se determina que, si tras 50 *epochs* consecutivas, el modelo no consigue ninguna mejoría, se finalice el entrenamiento, independientemente de las iteraciones realizadas. Este evento finalmente acaece al cabo de, únicamente, 72 *epochs*.

### 3.3.4 Aplicación del modelo

Tras un primer entrenamiento, se aplica el modelo de aprendizaje al video de las fragatas F-100 y del buque portaaviones L-61, sobre cuyos fotogramas se entrenó. Al ser estos videos la base de su entrenamiento, el modelo dispone de mayor facilidad para la identificación de los distintos elementos. Con todo esto, el resultado generado es mayoritariamente bueno, si bien se observan ciertas carencias y taras que se deben corregir. Para aplicar el modelo, se ejecuta el *script* del Anexo VII: Script de ejecución YOLOv8n.

El principal problema que se percibe en el análisis de los videos es que, según se aleja la cámara de ciertos objetos pequeños, o varía su perspectiva, el modelo es incapaz de mantenerlos identificados. Esto sucede aun cuando se permite ver con relativa facilidad los objetos en pantalla.

Un ejemplo de esto se puede ver en la Figura 3-15, donde se presenta, en fotogramas, una secuencia analizada del video de la F-100. Aquí se puede apreciar cómo, a medida que la cámara va rotando para dar frente a la fragata, se va perdiendo la detección algunos elementos. De igual manera, va descendiendo paulatinamente la fiabilidad en la detección de otros, como se puede ver indicado por el número a la derecha de cada etiqueta. Si bien en algunos casos, como en la cubierta de vuelo (*Flight\_Deck*), esto se debe a la oclusión de ese elemento, en otros se debe al simple cese por parte del modelo de su detección. Esto se puede apreciar con el radar de control de tiro (*Fire\_Control\_Radar*) y con la lancha estibada en el costado de la fragata (*Speed\_Boat*). Así mismo, en fotogramas anteriores se ve que el modelo identifica sin problema otros objetos que en la Figura 3-15 ya no aparecen detectados, como son otro controlador de tiro y otra antena de comunicaciones. El video analizado se puede ver completo en el siguiente enlace:

[“https://drive.google.com/file/d/1NVY-8XO6enPV48WJxOuu9gAtNQya4eqf/view?usp=share\\_link”](https://drive.google.com/file/d/1NVY-8XO6enPV48WJxOuu9gAtNQya4eqf/view?usp=share_link)



Figura 3-15 Aplicación inicial del modelo [elaboración propia]

Otro aspecto a analizar es la extensión de las etiquetas. Como se puede ver en la Figura 3-15, el nivel de detalle de información que se le ha dado a cada etiqueta ha resultado contraproducente. La longitud de la nomenclatura genera una falta de claridad en la representación del análisis, que dificulta su estudio. Así mismo, en la Figura 3-16, se puede ver como en la popa del buque, donde hay un número elevado de etiquetas debido a la presencia de múltiples aeronaves, se hace imposible discernir una etiqueta de otra, llegando alguna a estar completamente ocultada por las demás. Una posible solución a esta inconveniencia es la edición de las designaciones, acortando los nombres de cada etiqueta o mediante el empleo de siglas o acrónimos.

Empero, debido a que esta problemática tiene una naturaleza más estética y superficial que netamente funcional, ya que no afecta al desempeño del modelo, sino únicamente a su análisis, su resolución no es prioritaria en el proceso de desarrollo. No obstante, resulta como tarea pospuesta el solventar esta inconveniencia.



Figura 3-16 Portaaeronaves analizado por el modelo [elaboración propia]

Respecto a los aspectos positivos de esta primera aplicación del modelo, hay que destacar su correcta detección y clasificación, en algunas secciones de los videos, de la práctica totalidad de los elementos de relevancia presentes, como se puede ver tanto en la Figura 3-16 como en la Figura 3-17. Se aprecia la capacidad de diferenciar los distintos tipos de buques y sus correspondientes equipos, destacando los radares de control de tiro, de pequeño tamaño y modelos distintos entre sí, que tienen estructuras diferentes y conllevan una dificultad añadida para el sistema de detección. El video del que se ha extraído la Figura 3-16 se encuentra en el siguiente enlace:

[https://drive.google.com/file/d/1LosDuahGLRxSoWAq0A0464eoiTHO4jtg/view?usp=share\\_link](https://drive.google.com/file/d/1LosDuahGLRxSoWAq0A0464eoiTHO4jtg/view?usp=share_link)



Figura 3-17 Fragata analizada por el modelo [elaboración propia]

### 3.4 Mejora del modelo

Si bien hay aspectos favorables y menos favorables en la primera aplicación del modelo, en este apartado se analizará la aplicación del mismo sobre videos que, a diferencia del apartado anterior, no se han empleado en su entrenamiento. Estos, como se detallará a continuación, generan resultados que distan del favorable rendimiento del apartado anterior; se hace patente la necesidad de una ampliación del *dataset* y una mejora del modelo. Para ello, se plantea un procedimiento de ejecución basado en el empleo de *scripts*. En este apartado se lleva a cabo una mejora progresiva del sistema que culmina con un entrenamiento final que genera el modelo definitivo de detección de objetos flotantes en la mar.

### 3.4.1 Ampliación del dataset

La primera etapa en este procedimiento es la ampliación de la base de datos, para la cual se necesita repetir los procesos de obtención de fotogramas y de etiquetado vistos anteriormente. No obstante, debido a que ya se tiene un modelo entrenado aplicable a los videos del 300 Aniversario de la Real Compañía de Guardiamarinas, se hará uso de este para agilizar el proceso de etiquetado.

Para ello, se comienza por escoger los videos de los que se saquen los fotogramas a integrar en la base de datos. Con estos, se busca una perspectiva diferente de los buques que la proporcionada por los videos empleados en su primer entrenamiento. El primer video escogido es el de un BAM, un Buque de Acción Marítima. Estas embarcaciones, que se etiquetan en este trabajo como corvetas debido a su porte y funciones, no aparecen apenas en los videos procesados anteriormente; únicamente hacen presencia de fondo, mostrando su silueta en un plano lejano. Por ello, el modelo no es capaz de analizarlas correctamente, sin diferenciar bien de qué tipo de embarcación se trata ni distinguiendo los equipos que porta. Esto hace que sea una buena fuente de aprendizaje para el sistema.

Igualmente, se han analizado y procesado un gran número de nuevos videos diferentes, que aportan perspectivas visuales de las unidades distintas a las ya obtenidas. Debido a la larga duración de algunos de estos videos y para facilitar su procesamiento, requiriendo menor capacidad computacional, se fraccionan estos en segmentos de video de 10 segundos. Para ello, se ejecuta el siguiente comando en la terminal de *Miniconda*:

```
ffmpeg -i 11280039.MOV -c copy -map 0 -segment_time 00:10 -f segment -  
reset_timestamps 1 11280039_%03d.mov
```

Donde el video segmentado es el 11280039.MOV y cada 10 segundos de video se generará un segmento. El nombre de salida de este llevará el prefijo “11280039\_” seguido de un número secuencial de tres dígitos y la extensión “.mov”. El primer segmento de este video se llamará 11280039\_000.mov, el segundo 11280039\_001.mov, y así hasta que finalice la duración del video. Para segmentar otros videos, basta con cambiar el nombre del video y el prefijo en este código por el del nuevo video.

Una vez se han fragmentado los videos, se procede a extraer de los segmentos los fotogramas. Para ello, al igual que en el primer etiquetado para este *dataset*, se empleará el *script* del Anexo V: Script de extracción de imágenes. No obstante, la diferencia con el procedimiento anterior es que, en esta ocasión, antes de extraer las fotografías del video, se aplicará el modelo entrenado para detectar los objetos presentes en el video. El mismo *script* permite ejecutar esa acción. De esta forma, tras aplicarlo, se extraen las imágenes ya etiquetadas por el propio modelo, lo que hace que en vez de tener que etiquetarlas por completo, como se ha hecho anteriormente, únicamente halla que corregir los errores cometidos por el modelo. Además, esta metodología permite ver donde, y desde que perspectiva, le cuesta más al modelo identificar los objetos y cuáles de ellos no es capaz de detectar. En la Figura 3-18 se puede ver una de las imágenes analizada por el modelo procedentes del video de un BAM. Como se puede observar, este no ha sido capaz de identificar correctamente los elementos de relevancia en la fotografía. El sistema ha confundido el tipo de embarcación, corveta, con el de una fragata, que es similar, pero de mayor tamaño, y tampoco ha clasificado correctamente las antenas de comunicaciones satélite, llegando a confundir una con un radar de control de tiro. Así mismo, elementos como el armamento o el radar de navegación no los ha sido capaz de detectar. Sí que ha clasificado correctamente al submarino y a la cubierta de vuelo, elementos muy presentes en el *dataset* inicial sobre el que se ha entrenado el modelo. Esto indica que, con este tipo de embarcación y desde esa perspectiva, el sistema todavía no ha aprendido a detectar los patrones de muchos de los diferentes objetos. Al arreglar el etiquetado de esa imagen, como se puede ver en la Figura 3-19, y complementándolo con otras imágenes similares, se da al *dataset* la información que necesita el modelo para subsanar la carencia.



**Figura 3-18 BAM analizado por el modelo [elaboración propia]**

Para ajustar el etiquetado se emplea nuevamente la herramienta *Label Studio*. Para ello, se importan los fotogramas estudiados por el modelo junto con las etiquetas que este ha generado. Estas etiquetas las genera el mismo *script* que extrae las imágenes, en formato JSON, que permite su importación y lectura por parte de *Label Studio*. Una vez en la herramienta, integradas con el resto de imágenes ya etiquetadas y que conforman el *dataset* sobre el que se ha entrenado el modelo, se corrigen y añaden todas las etiquetas que deben aparecer en cada imagen, como se ve en la Figura 3-19.



**Figura 3-19 BAM con etiquetas corregidas [elaboración propia]**

### 3.4.2 Actualizaciones del modelo

Según se va aumentando la cantidad de imágenes etiquetadas, se lleva a cabo una serie de entrenamientos intermedios antes de realizar un entrenamiento final que genere el modelo definitivo. Estas actualizaciones del modelo se van ejecutando para apreciar el desarrollo del proceso de detección del sistema.

Tras el proceso detallado en el apartado anterior, una vez procesadas un número de imágenes nuevas, se integran a la base de datos descargando todas las imágenes de *Label Studio* y ejecutando a continuación el *script* del Anexo IV: Script de organización de imágenes. Con el nuevo *dataset* se reentrena al modelo y se genera uno nuevo. En la Figura 3-20 se puede ver un fotograma del video del BAM (procesado e integrado en el nuevo *dataset*) analizado por el modelo actualizado. Se aprecia, al comparar la imagen con la Figura 3-18, la significativa mejoría del modelo a la hora de identificar los elementos presentes. No obstante aún se observan carencias, como la no identificación del radar de navegación o del controlador de tiro; en la Figura 3-19 se muestra como se ha buscado enseñar al modelo a identificar esos objetos al etiquetarlos en las imágenes corregidas. Se puede acceder al video de la Figura 3-20 a través de este enlace:

[“https://drive.google.com/file/d/1NNDZjoyRUKLxQNhCeC7OfI9onwMipLsX/view?usp=share\\_link”](https://drive.google.com/file/d/1NNDZjoyRUKLxQNhCeC7OfI9onwMipLsX/view?usp=share_link)

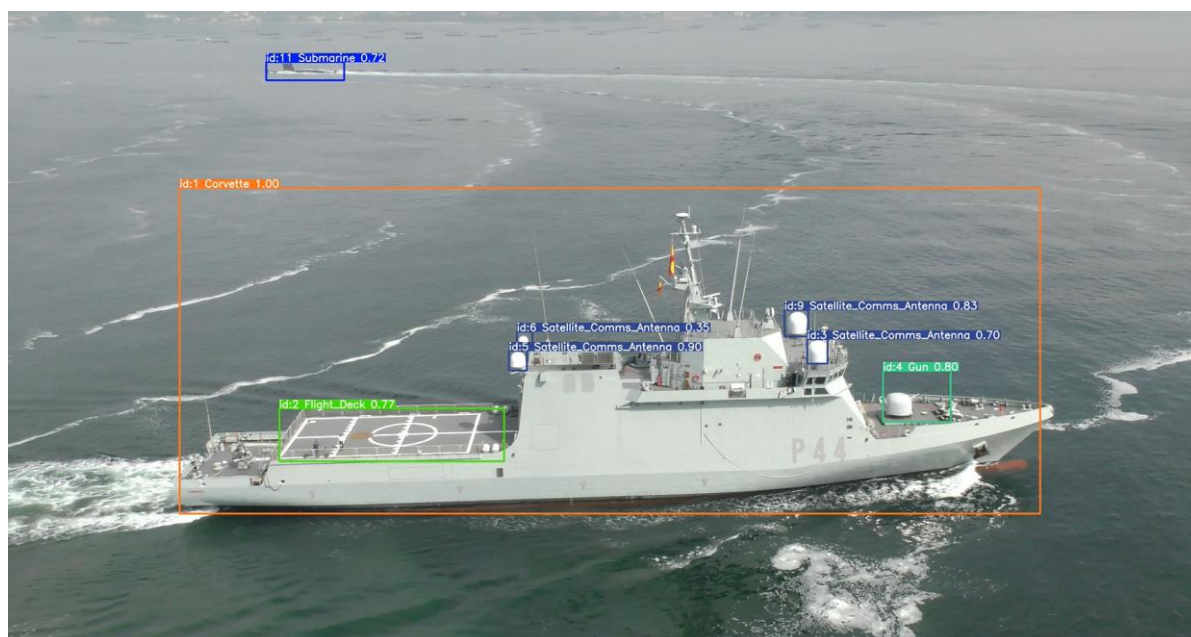


**Figura 3-20 BAM analizado con el modelo reentrenado [elaboración propia]**

A su vez, este nuevo modelo se ejecuta con videos no analizados anteriormente (mediante el *script* del Anexo VII: Script de ejecución YOLOv8n) y se observan sus taras. En caso de que el análisis de los videos no haya sido satisfactorio, es decir, que no se detecten bien los objetos, los nuevos videos se procesan para pasarlos a fotos ya etiquetadas por el modelo. Por último, estas nuevas fotografías ya analizadas son corregidas a través de *Label Studio*, siguiendo el procedimiento del apartado anterior, y se integran de nuevo al *dataset*, que se vuelve a emplear para realizar un nuevo entrenamiento sobre el modelo más reciente.

Este proceso se ha ejecutado un total de 5 veces, actualizando y reentrenando al modelo, hasta que se han añadido 330 nuevas imágenes al *dataset*, conformado ahora por 570 fotografías. En la Figura 3-21 se ve un diagrama que representa de forma esquemática y simplificada el proceso ejecutado de desarrollo y mejora del modelo.

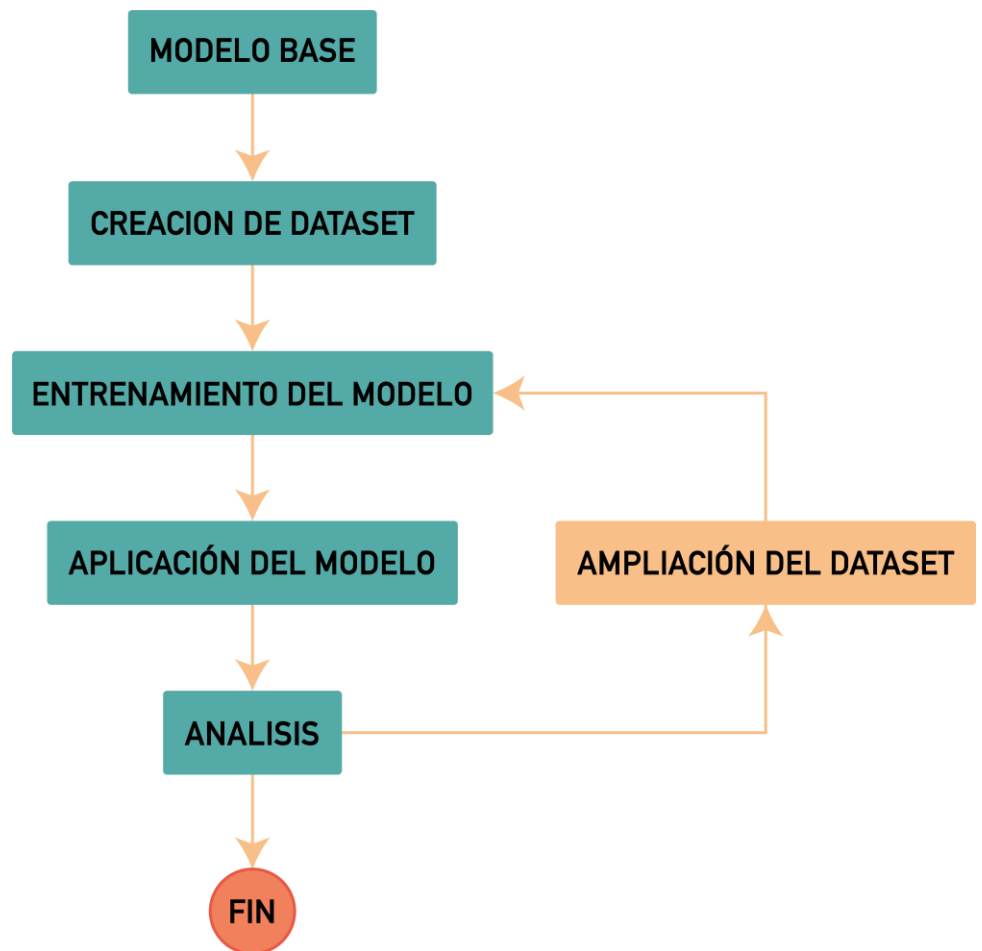


**Figura 3-21 Diagrama del proceso de desarrollo [elaboración propia]**

Se puede apreciar, al igual que en la Figura 3-20, en la Figura 3-22, como tras sus reentrenamientos, el modelo ha sido capaz de aprender patrones que le eran desconocidos anteriormente, o cuya identificación aún le costaba. De esta forma, se ha mejorado significativamente la capacidad del sistema para detectar y *trackear* los objetos de los videos; esto conlleva la validación, como fundamento de aprendizaje para el modelo final, del *dataset* generado. El modelo definitivo se entrena en el siguiente apartado.

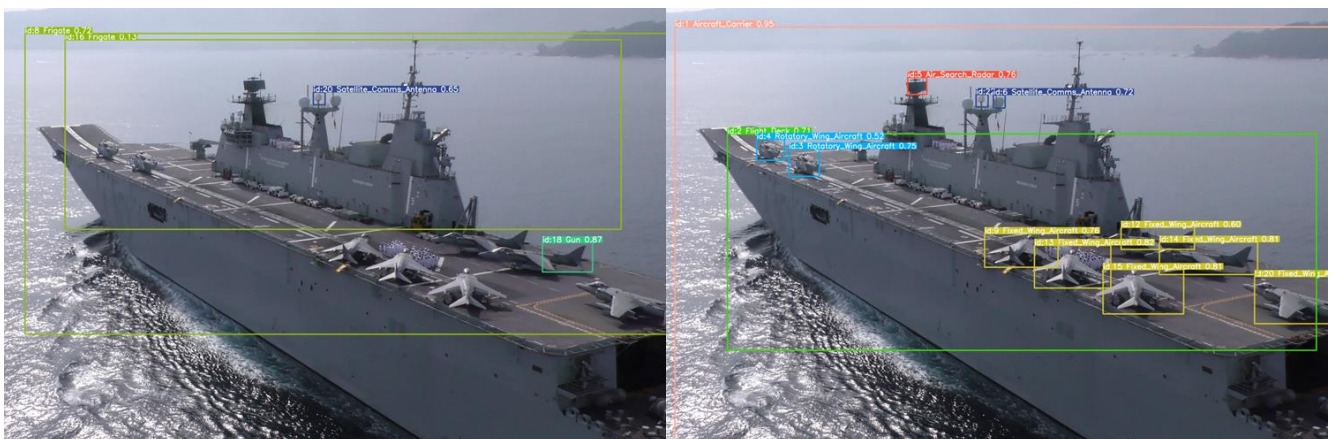


**Figura 3-22 L-61 analizado en distintos estados del entrenamiento [elaboración propia]**

El video sobre el que se ha obtenido la imagen corregida del L-61 se encuentra en este enlace:

“[https://drive.google.com/file/d/10e33cUiUiwnvb2o1eE3UmAXna5F6S4Hq/view?usp=share\\_link](https://drive.google.com/file/d/10e33cUiUiwnvb2o1eE3UmAXna5F6S4Hq/view?usp=share_link)”

### 3.4.3 Entrenamiento final

Habiendo ampliado significativamente el *dataset*, se procede a realizar el entrenamiento que genere el modelo final de aprendizaje. Para ello, en lugar de reentrenar el último modelo actualizado en el apartado anterior, se aplicará esta base de datos al entrenamiento de un modelo base, similar al empleado en el primer entrenamiento del modelo en el apartado 3.3.3, que no ha sido entrenado previamente en el ámbito marino. La única distinción con los modelos anteriores, y la razón por la cual se ha de entrenar de cero, es que este modelo tendrá una estructura *small*, frente a la *nano* del modelo base anterior. Su algoritmo es el YOLOv8s, lo que conlleva una mayor complejidad arquitectónica de su red neuronal. Esta nueva estructura tiene como fin generar mejores resultados y crear una mayor fiabilidad en la detección e identificación de los objetos flotantes en la mar. No obstante, su entrenamiento supone una gran carga computacional, a la que no se puede dar soporte en local. Por ello, se hace uso de *Google Colab* para el entrenamiento, adquiriendo el pase que quita los límites al acceso a GPUs y TPUs de gran potencia, necesarios en esta fase de desarrollo.

Una vez adquirido el pase, se ejecuta el entrenamiento del modelo. Para ello, a diferencia del empleo de *scripts* en *Codium* (local) de los apartados anteriores, se ejecutan una serie de comandos en *Colab*. Estos comandos, que se pueden ver en la Figura 3-23, sirven para establecer los medios para el entrenamiento, y a continuación, ejecutarlo. Indican el directorio, llamado “BARCOS\_ARMADA” y ubicado en *Google Drive*, en el que se encuentra el *dataset* y el modelo base, y al que *Colab* tiene que acceder para operar. El último comando ordena la ejecución del entrenamiento.

```

nvidia-smi

Tue Mar 21 07:12:17 2023

+-----+
| NVIDIA-SMI 525.85.12      Driver Version: 525.85.12      CUDA Version: 12.0      |
+-----+-----+
| GPU   Name           Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|  Memory-Usage | GPU-Util  Compute M. |
|-----+-----+
| 0     Tesla T4             Off          00000000:00:04:0  Off      0%          Default  N/A
| N/A   66C    P0      31W /  70W   0MiB / 15360MiB          0%          N/A
+-----+-----+

+-----+
| Processes:                                                       GPU Memory |
|  GPU   GI    CI          PID    Type   Process name                  Usage   |
| -----+-----+
| No running processes found
+-----+

import os
HOME = os.getcwd()
print(HOME)

/content

# Pip install (recommended)

!pip install ultralytics

from IPython import display
display.clear_output()
!yolo mode=checks

WARNING ⚠️ 'yolo mode=checks' is deprecated. Use 'yolo checks' instead.
Ultralytics YOLOv8.0.55 Python-3.9.16 torch-1.13.1+cud116 CUDA:0 (Tesla T4, 15102MiB)
Setup complete ✅ (2 CPUs, 12.7 GB RAM, 25.4/78.2 GB disk)

from ultralytics import YOLO

from IPython.display import display, Image

from google.colab import drive
drive.mount('/content/drive')
!ls -laF

Mounted at /content/drive
total 20
drwxr-xr-x 1 root root 4096 Mar 21 07:13 ./
drwxr-xr-x 1 root root 4096 Mar 21 07:10 ../
drwxr-xr-x 4 root root 4096 Mar 17 13:39 .config/
drwx----- 5 root root 4096 Mar 21 07:13 drive/

%cd /content/drive/MyDrive/BARCOS_ARMADA/
!ls -laF

/content/drive/MyDrive/BARCOS_ARMADA
total 115912
drwx----- 2 root root 4096 Mar 20 10:47 train/
-rw----- 1 root root 6596 Mar 21 07:12 train-yolov8-barcos-armada.ipynb
drwx----- 2 root root 4096 Mar 20 10:39 YOLO/
-rw----- 1 root root 89570082 Mar 20 19:26 yolov8s_best.pt
-rw----- 1 root root 6534387 Mar 20 10:47 yolov8n.pt
-rw----- 1 root root 22573363 Mar 20 10:47 yolov8s.pt

!yolo task=detect mode=train model=/content/drive/MyDrive/BARCOS_ARMADA/yolov8s_best.pt project=/content/

Ultralytics YOLOv8.0.55 Python-3.9.16 torch-1.13.1+cud116 CUDA:0 (Tesla T4, 15102MiB)
yolo/engine/trainer: task=detect, mode=train, model=/content/drive/MyDrive/BARCOS_ARMADA/yolov8s_best.pt

      from n      params module      arguments
0         -1 1         928 ultralytics.nn.modules.Conv [3, 32, 3, 2]
1         -1 1      18560 ultralytics.nn.modules.Conv [32, 64, 3, 2]
2         -1 1      29056 ultralytics.nn.modules.C2f [64, 64, 1, True]
3         -1 1      73984 ultralytics.nn.modules.Conv [64, 128, 3, 2]

```

Figura 3-23 Comandos en *Colab* [elaboración propia]

El proceso de aprendizaje se desarrolla durante 9 horas, tras las cuales, es capaz de llevar a cabo 224 *epochs*. No obstante, debido a una posible sobrecarga en el empleo de los servidores de *Google Colab*, se pierde la conexión antes de que finalice el entrenamiento. Para remediar esta problemática, se reinicia el entrenamiento empleando como modelo base el modelo ya entrenado hasta las 224 *epochs*. Esto se hace con el fin de agilizar en la mayor medida posible el proceso de aprendizaje, y también, de evitar sobrepasar el límite computacional permitido por el pase adquirido de *Colab*.

Después de 2 horas y 51 *epochs* más, concluye el entrenamiento del sistema y se da por finalizado el proceso de desarrollo del modelo de aprendizaje. Resta únicamente comprobar el buen desempeño del sistema y verificar su posible integración a un dron, que se realizará en el siguiente apartado.

### 3.5 Integración del sistema a la plataforma Coral

Si bien el sistema de identificación elaborado en el apartado anterior se ha desarrollado para su empleo desde cualquier plataforma que aporte una perspectiva aérea, en este TFG se pretende que se pueda emplear desde un vehículo aéreo tripulado remotamente. Para ello, se ha buscado una herramienta que posibilitase su integración a un dron. Mediante la incorporación del modelo al dispositivo *Coral USB Accelerator*, que se puede ver en la Figura 3-24, se puede alcanzar este objetivo. El *Accelerator* de *Coral* ofrece un soporte adecuado para un modelo de aprendizaje profundo y, a su vez, permite su conexión a los sistemas y equipos que incorpora un dron, como su cámara de vídeo. También, gracias a sus reducidas dimensiones, se conforma como la plataforma ideal para su empleo por estas aeronaves tripuladas de forma remota; su tamaño le confiere una gran portabilidad.



**Figura 3-24 Coral USB Accelerator [90]**

No obstante, debido al plazo de tiempo del que se dispone para la realización de este TFG, no se ha conseguido llevar a cabo el proceso de implementación del modelo al dispositivo portátil de *Coral*. Queda este apartado del trabajo, por tanto, en desarrollo y pendiente de su conclusión.

## 4 OBTENCIÓN Y ANÁLISIS DE RESULTADOS

En este capítulo se recogen los resultados y análisis del modelo desarrollado, así como de su aplicación en diferentes videos. Primeramente, se exponen las características alcanzadas por el modelo final, basado en el algoritmo YOLOv8s, y se analizan sus capacidades. A continuación, se someten estas a diferentes pruebas, consistentes en el análisis de videos de orígenes distintos, pero de la misma temática: buques de guerra en navegación. Aquí, se busca comprobar y analizar las aptitudes del sistema en la localización, identificación y seguimiento de los distintos tipos de elementos presentes en el ámbito naval.

### 4.1 Análisis del modelo

Para comenzar el análisis del modelo final, es necesario destacar la gran diferencia apreciada entre la construcción de un sistema de detección desde un punto fijo, con uno de detección desde una perspectiva móvil. Para la introducción a la metodología se construyó satisfactoriamente un modelo de perspectiva fija, surgiendo en este momento diferentes retos como la idoneidad de la perspectiva para poder ser siempre capaz de identificar los contactos, o la poca capacidad de acercamiento a los distintos objetos para facilitar su identificación. Sin embargo, los retos emanados del modelo de perspectiva aérea han sido los opuestos. Las principales problemáticas que han dificultado el buen desempeño del modelo han sido el constante cambio en la perspectiva de la cámara, los tambaleos de la cámara y la aeronave sobre la que estaba montada, que obstruían la continuidad del enfoque de las imágenes, y la constante variación del tamaño de los barcos al acercarse y alejarse la cámara de ellos. Todo esto ha obstaculizado al aprendizaje de los patrones, si bien, el modelo ha podido sobreponerse a estas situaciones, con mayor o menor éxito, gracias a la ampliación del *dataset*, que ha aportado información de gran utilidad para gestionar con éxito estas circunstancias.

Por otro lado, también es necesario mencionar el buen resultado que ha supuesto adaptarse a la última versión del algoritmo YOLO. Debido al poco recorrido que ha tenido hasta el momento de redacción, emplear otras versiones, como YOLOv5 sobre la que se basa el primer modelo construido en este TFG, habría supuesto una mayor comodidad y seguridad para el trabajo. No obstante, ya que se persigue potenciar y aplicar la innovación tecnológica, la única opción consecuente con este propósito del TFG era emplear YOLOv8, pese al riesgo que su tan reciente desarrollo suponía. Esta última versión ha conllevado aplicar metodologías distintas, pero ha posibilitado generar un resultado plenamente satisfactorio y acorde a la actualidad del ámbito del aprendizaje profundo, como se detallará a continuación.

Durante el entrenamiento final, donde se ha empleado el algoritmo YOLOv8s, se han generado una serie de gráficas en las que se ven reflejados parámetros de relevancia respecto al proceso de aprendizaje del nuevo modelo. La más importante de estas gráficas es la que enfrenta los parámetros de *Precision* y

*Recall*. Como se estudió en el apartado 3.2.3, el valor *Precision* indica la proporción de detecciones positivas que son verdaderas, es decir, la cantidad de elementos detectados que realmente son objetos. *Recall*, por otro lado, mide la capacidad para identificar todos los verdaderos positivos en un conjunto de datos; indica cuántos de los objetos que realmente existen en la imagen el modelo ha sido capaz de detectar. La gráfica de *Precision-Recall*, que se muestra en la Figura 4-1, muestra la relación entre estas dos métricas. Un buen modelo debe tener tanto una alta *Precision*, como un alto *Recall*, ya que esto significaría que es capaz de identificar la mayoría de los verdaderos positivos mientras minimiza los falsos positivos.

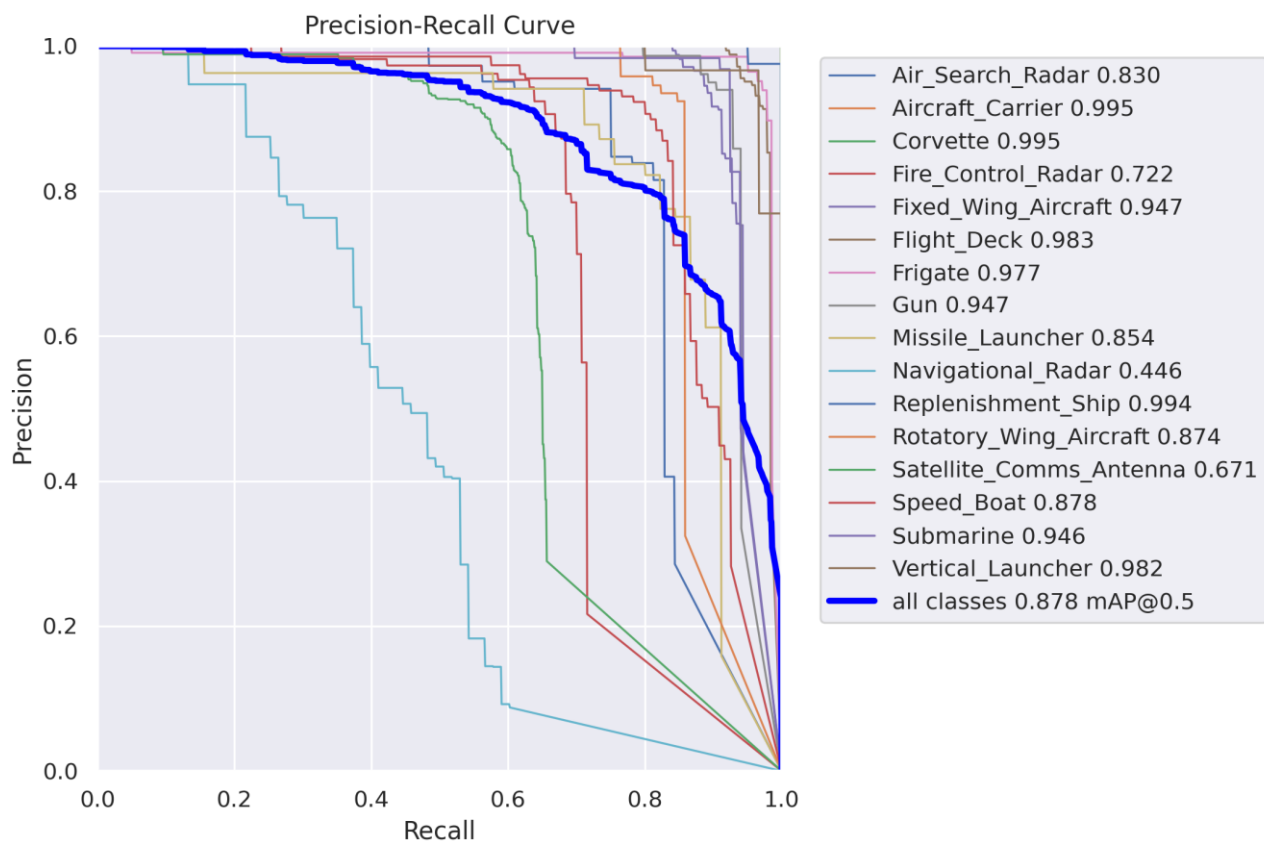


**Figura 4-1 Precision-Recall [elaboración propia]**

En la gráfica de la Figura 4-1, para un modelo ideal, la línea azul, que indica la media de todas las clases (*all classes*), bordearía los límites superior y derecho de la gráfica durante todo su recorrido, hasta juntarse en la esquina superior derecha, e indicando, así, una *Precision* y un *Recall* perfectos para todas las clases de elementos a detectar. Se puede observar como ese no es el caso en los resultados obtenidos con este modelo. Para analizar estos resultados, se tiene en cuenta el área debajo de la curva generada por esa línea azul. Esta representa la integral de la curva y proporciona una media general del rendimiento del sistema. En este caso, el valor resultante es de 0,8, lo que es un buen resultado, ya que cuanto más se acerque a 1, mejor será el rendimiento, mientras que, si el valor es cercano a 0,5, el modelo puede estar realizando predicciones aleatorias. Un valor atípico es el obtenido por el análisis del radar de navegación (*Navigational\_Radar*). Este es el elemento de menor tamaño que se ha etiquetado en las imágenes del *dataset*, y el cual tampoco ha tenido presencia en la mayoría de las imágenes ni se le ha focalizado en ninguna perspectiva analizada. Estas razones han debido de causar una carencia, en el modelo, de información suficiente para la correcta detección e identificación de este equipo.

Otra gráfica de gran valor informativo es la que se puede ver en la Figura 4-2, la matriz de confusión. Esta representación gráfica indica la tasa de aciertos y errores del modelo en cada clase, y facilita así identificar las clases en las que el modelo comete más errores. Como se puede observar, se miden las

predicciones realizadas por el modelo para cada clase, frente a los objetos reales presentes en las imágenes. Cuanto más se aproxime el valor resultante de cada clase a la unidad, mejor habrá sido su detección por parte del modelo. Analizando la gráfica, se puede ver como las predicciones realizadas por parte del modelo sobre la presencia de un portaaviones (*Aircraft\_Carrier*), se han corroborado en todo momento con la situación real de un portaaviones, si bien, en un mínimo número de casos (1%), la etiqueta ha sido demasiado grande y ha abarcado más espacio que el estrictamente necesario para delimitar el buque.

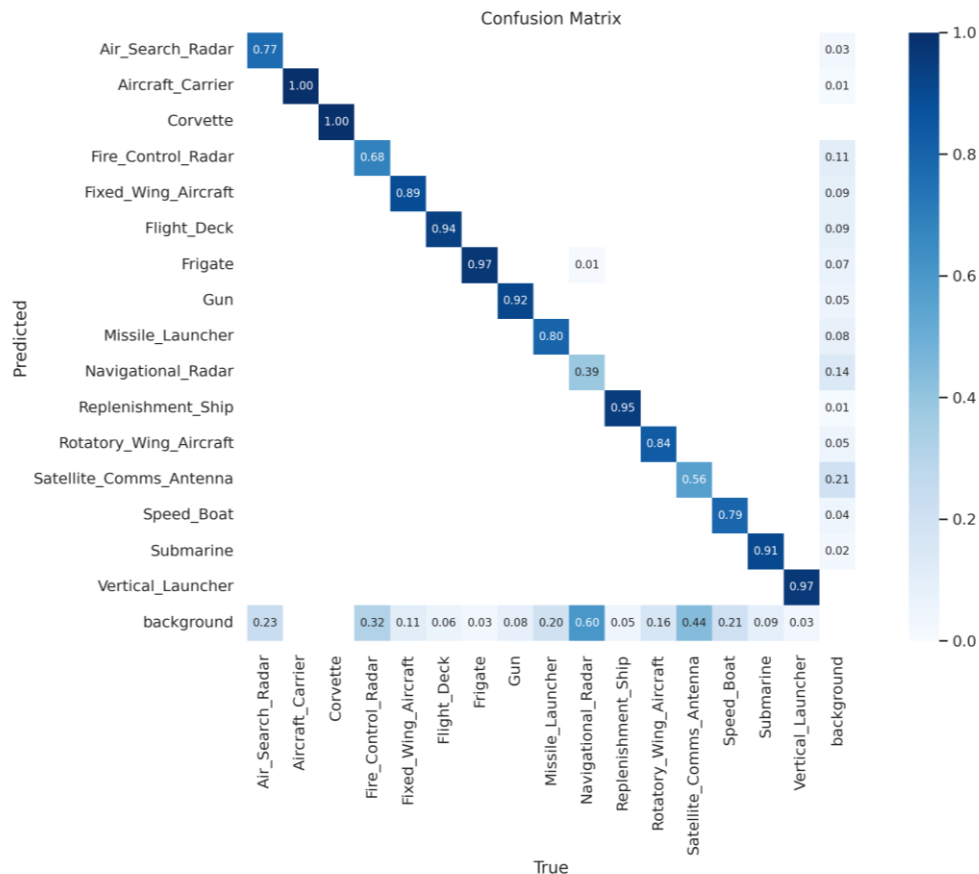


**Figura 4-2 Matriz de confusión [elaboración propia]**

Un valor que sigue la línea del análisis efectuado sobre la gráfica de la Figura 4-1, es el del radar de navegación. Se ve como tan solo en un 39% de las ocasiones es capaz, el modelo, de detectarlo correctamente, y como en un 60% lo pierde con el fondo de la imagen. Ese 1% restante, que resalta en medio de la matriz de confusión, se da cuando el modelo incluye el radar dentro de la etiqueta de su buque, en este caso de una fragata. Esta es una de las complicaciones que surgen de desarrollar un modelo que busque detectar objetos dentro de otros objetos, los cuales debe de detectar también. Otra observación reseñable sobre la matriz es que la fila de “background”, en el eje *Predicted*, indica la proporción de elementos presentes en la imagen que no han sido detectados por el modelo, mientras que en el eje *True*, indica las veces que los recuadros de elementos identificados han sido demasiados grandes, incluyendo parte del fondo, o bien que se ha indicado la presencia de objetos donde no había ninguno.

En esta siguiente imagen, la Figura 4-3, se indica el número de elementos presentes en las imágenes procesadas durante el entrenamiento del modelo. Debido a su presencia en la práctica totalidad de las embarcaciones analizadas, y a que en cada buque había una media de 3 antenas, resalta el gran número de antenas satélite procesadas, superando significativamente al resto de objetos analizados. Así mismo,

en la imagen se indican las ubicaciones y tamaños de los elementos, destacando la eminente situación central de la mayoría de los objetos estudiados.

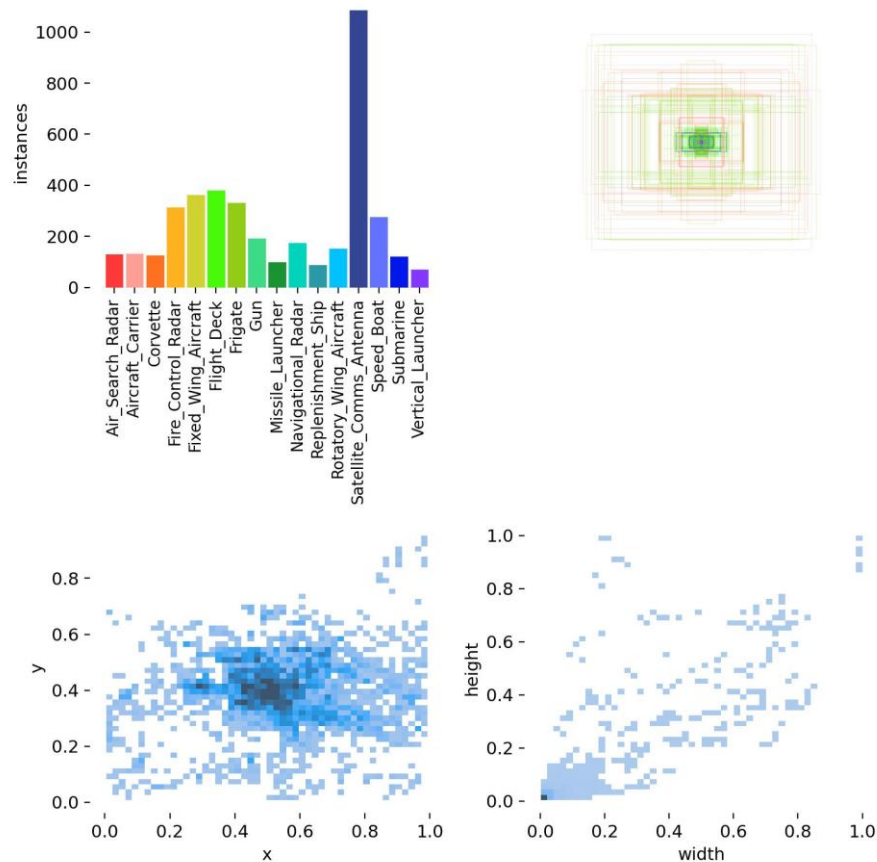


**Figura 4-3 Características de los objetos analizados [elaboración propia]**

Se puede observar, así mismo, el amplio abanico de tamaños que tienen los elementos analizados. Sin embargo, debido al análisis de los equipos adjuntos a los buques, la gran mayoría de los objetos han tenido un tamaño reducido; ya que se buscaba analizar el buque en su conjunto, a una escala adecuada, los equipos sufrieron como consecuencia el estar a la misma escala, y, por tanto, con un tamaño significativamente menor.

A continuación, se muestra, en la Figura 4-4, una serie de imágenes extraídas del proceso de entrenamiento, donde el modelo identificaba los distintos tipos de buque y sus diferentes equipos.

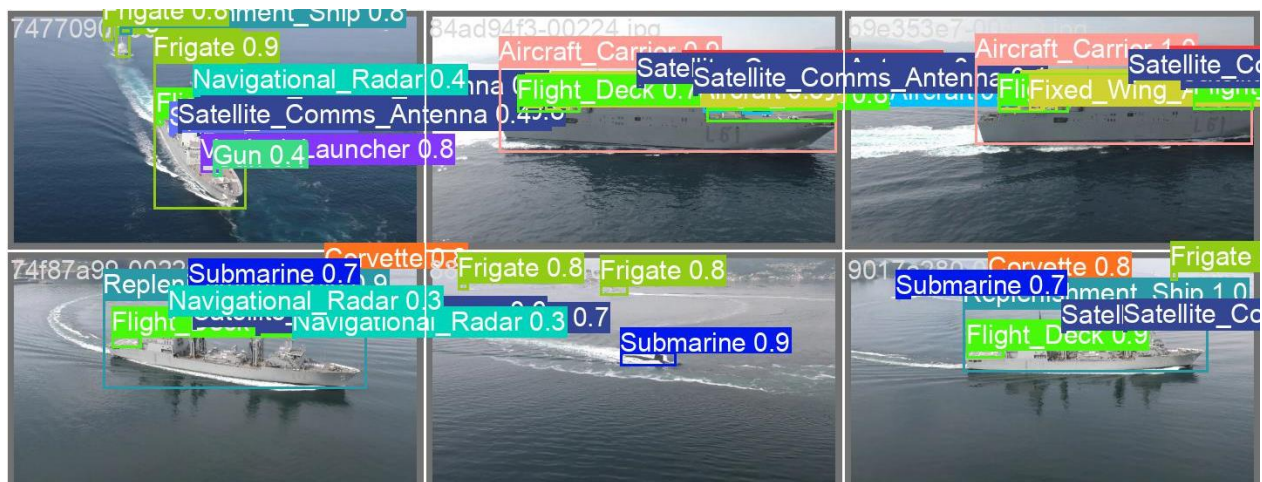


**Figura 4-4 Elementos identificados durante el entrenamiento [elaboración propia]**

Tras este análisis de los datos obtenidos sobre el modelo, se concluye que es un sistema capaz de desenvolverse de forma plenamente satisfactoriamente en el ámbito sobre el que ha sido entrenado. A pesar de que diste de tener un rendimiento perfecto, los resultados generados son acordes a los buscados.

## 4.2 Aplicación y análisis de resultados

Para apreciar de forma práctica el desempeño del modelo, y analizar su buen funcionamiento de forma empírica, una vez estudiadas sus capacidades se ponen estas a prueba mediante la aplicación del modelo para el procesamiento de una serie de diferentes videos. De estos, se puede obtener el nivel de detalle al que el modelo es capaz de trabajar y la capacidad de detectar, clasificar y *trackear* correctamente los distintos objetos.

A continuación, se analizan videos de dos procedencias distintas. Primeramente, se procesan aquellos sobre los que se ha entrenado al modelo, esperando unos resultados de gran precisión. Seguidamente, se aplicará el modelo a material del mismo origen, pero cuyas imágenes no han tenido parte alguna en el aprendizaje del modelo; aquí se buscan resultados satisfactorios, más alguna deficiencia es de esperar. Y, por último, se analizan dos videos que tratan la misma temática, pero cuyos elementos, si bien similares a los estudiados, el modelo nunca los ha visto antes. Los resultados que ofrezca este análisis no se espera que sean de gran calidad, ya que el sistema no ha sido entrenado en ese escenario.

### 4.2.1 Video parada naval

El primer video en analizarse es uno de aquellos cuyos fotogramas han conformado parte del *dataset* sobre el que ha aprendido el modelo. El video analizado muestra al portaaviones L-61 realizando una pasada un BAM que se encuentra estático y en un primer plano, ocluyendo ligeramente al portaaviones a medida que este avanza. En la Figura 4-5 se puede ver un instante del video, al que se puede acceder siguiendo este enlace:

[“https://drive.google.com/file/d/17rUmPqZFxtawSzFC5N3mqCV3UewESfw/view?usp=share\\_link”](https://drive.google.com/file/d/17rUmPqZFxtawSzFC5N3mqCV3UewESfw/view?usp=share_link)

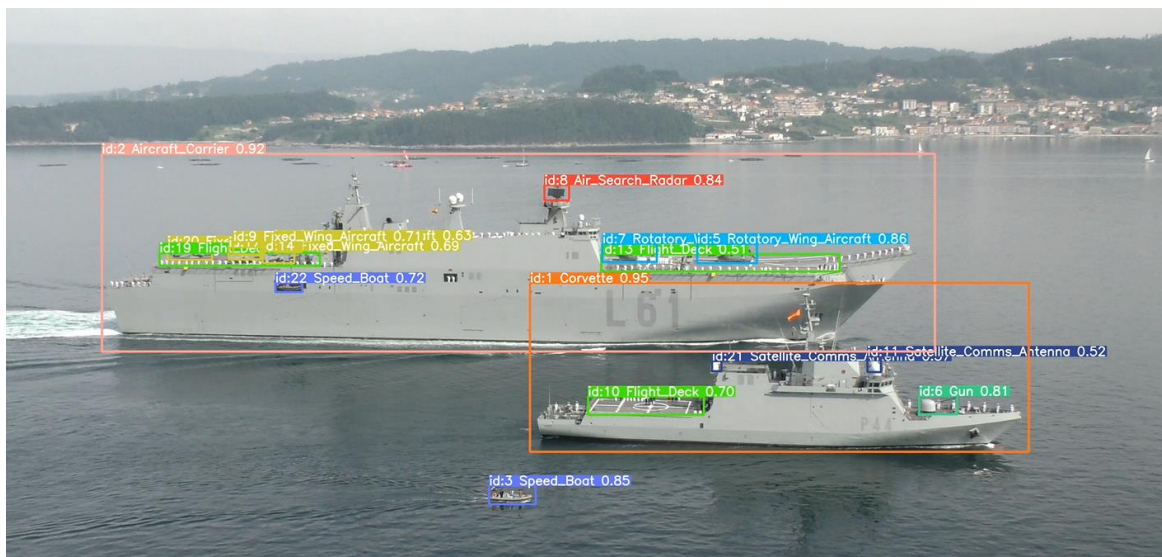


Figura 4-5 Pasada del L-61 [elaboración propia]

Se aprecia cómo hay algunos elementos que no se han podido identificar, como todas las antenas de comunicaciones por satélite, o el radar de dirección de tiro del BAM, que en algunos momentos sí que llega a detectar el modelo. A lo largo del video se van perdiendo también otros elementos que luego se recuperan, como el radar aéreo del L-61. No obstante, el resultado general es más que satisfactorio. El principal matiz a resaltar es la capacidad de delimitar ambos buques adecuadamente, aun al interferirse

mutuamente las cajas delimitadoras. Así mismo, destaca como se identifica una lancha motora (*Speed\_Boat*), tanto navegando, como estivada a bordo del portaaeronaves.

El siguiente video es también una grabación de la parada naval del 300 aniversario de la creación de la Real Compañía de Guardiamarinas, en la Ría de Pontevedra. Sin embargo, a diferencia del anterior, las imágenes de este video no han formado parte del *dataset* de entrenamiento, por lo que perspectivas de ciertos buques le son desconocidas al modelo. Aun así, al ser este el entorno sobre el que ha aprendido el sistema, los resultados son buenos. Para visualizar el video, se puede seguir este enlace web:

“[https://drive.google.com/file/d/1u0cLYv-b4eBAKnZO1hsKlF3hqugq9J7A/view?usp=share\\_link](https://drive.google.com/file/d/1u0cLYv-b4eBAKnZO1hsKlF3hqugq9J7A/view?usp=share_link)”

Como se puede ver en la Figura 4-6, el encuadre es similar al del video anterior, aunque ligeramente más alejado. Esta similitud facilita al modelo identificar y hacer *tracking* al BAM y a algunos de sus elementos, así como a la lancha. No obstante, la presencia de la fragata en la zona del BAM confunde al modelo, que no termina de discernir con claridad su límite superior, integrando en la etiqueta de “*Corvette*” secciones de la fragata.

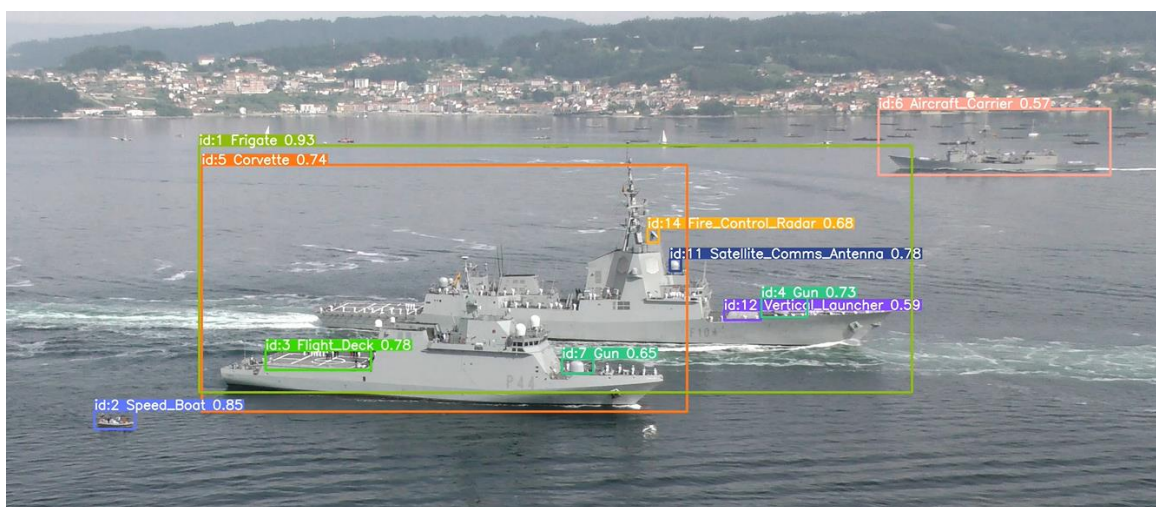


**Figura 4-6 Pasada de una F-100 [elaboración propia]**

El modelo sí que es capaz de identificar varios de los equipos de la fragata, destacando el radar de control de tiro y el lanzador vertical, equipos de difícil identificación por sus tamaños y geometrías. Sin embargo, aquellos elementos que se encuentran en la zona de convergencia con el BAM, como su lanzador de misiles o su cubierta de vuelo, no se han podido detectar. Al modelo le ha confundido claramente que ambos buques estén tan cerca, no terminando de ser capaz de dirimir donde empieza uno y donde acaba el otro. Esto se puede comprender al ver la etiqueta de “*Frigate*”, que comprende la casi totalidad del BAM. Por otro lado, el segundo tipo de fragata presente en la imagen (de la clase F-80), no se ha conseguido clasificar correctamente, asignándole una etiqueta de portaaviones al, posiblemente, no ser capaz de filtrar la presencia de las bateas de la silueta de la fragata.

Para poder apreciar la mejoría que ha supuesto que el entrenamiento del modelo final se realizase sobre el algoritmo YOLOv8s (*small*), en lugar del YOLOv8n (*nano*) que se ha empleado en todos los entrenamientos anteriores, en la Figura 4-7 se muestra el mismo fotograma que en la Figura 4-6, pero analizado con el último modelo entrenado basado en YOLOv8n. Los dos modelos cuentan con el mismo *dataset*, conformado por 570 imágenes iguales para ambos. Si bien la clasificación general es similar en ambos casos, se hace palpable la mayor capacidad de análisis y el mejor rendimiento del modelo basado en la arquitectura *small*.

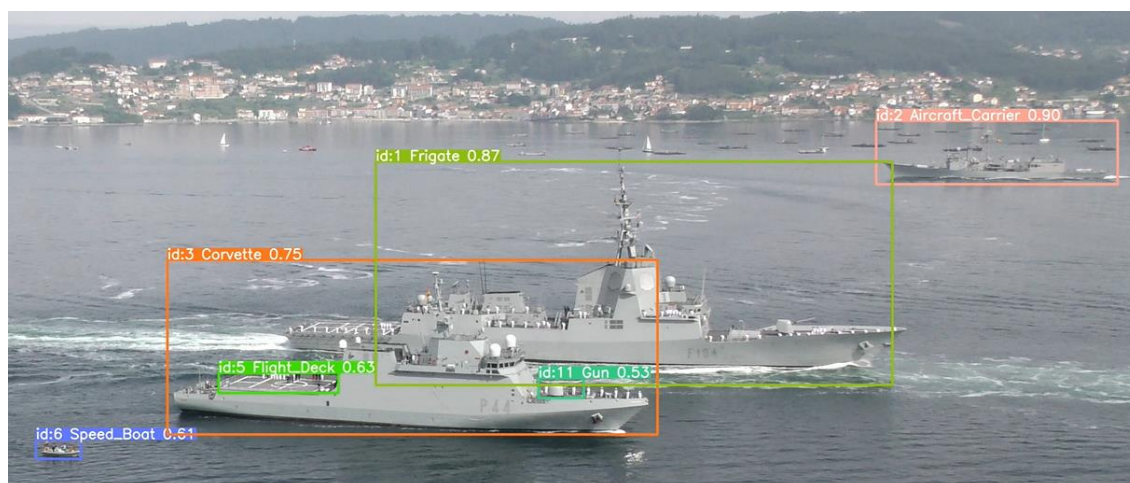


Figura 4-7 Análisis de pasada con YOLOv8n [elaboración propia]

#### 4.2.2 Video externo

Una vez se ha analizado el desempeño del modelo en el ambiente en el que se le ha entrenado, se procede a poner a prueba sus capacidades de detección, identificación, y seguimiento de buques y de sus equipos en un escenario que nunca ha analizado previamente ni del que ha aprendido.

Para ello, se emplean dos videos con diferentes perspectivas aéreas de las embarcaciones, una lejana y otra algo más cercana. Los videos seleccionados son imágenes de los ejercicios RIMPAC (*Rim of the Pacific*) de 2016. En estos ejercicios, organizados por la Armada de los Estados Unidos y que se desarrollan de forma bianual en las aguas próximas a las Islas Hawái, se llevan a cabo formaciones navales, con hasta más de 40 buques y submarinos, de 13 marinas de guerra distintas, lo que propicia una fuente idónea para comprobar las capacidades desarrolladas por el modelo.

No obstante, no se esperan resultados similares a los obtenidos en el apartado anterior; no se ha entrenado al modelo con estas clases de buques ni con sus equipos anejos, por lo que desconoce sus patrones. Llevar a cabo una ampliación del *dataset*, para incluir todas las tipologías y clases de buques, conllevaría un incremento exponencial de la capacidad computacional necesaria para aplicar el modelo, lo que resultaría en una problemática, de compleja resolución, respecto a la integración del modelo en una plataforma portátil de pequeñas dimensiones, factor cardinal de este trabajo. Aun teniendo todo esto en cuenta, al sí conocer algunos patrones similares, se busca que el modelo pueda, por lo menos, detectar algún buque.

En el primero de los dos videos aparecen hasta 11 buques. Aquí, como se puede ver en la Figura 4-8, el modelo es capaz de detectar cinco de esos buques. El punto de visión de la cámara grabadora es significativamente elevado, perspectiva que, si bien está presente, no se ha incluido con demasiada extensión en el *dataset*. El proceso de identificación le es más complicado que el de detección, pues en alguno de los barcos oscila entre asignaciones distintas, como se puede observar comparando la Figura 4-8 con la Figura 4-9. Sin embargo, una vez ha detectado esos cinco elementos, hace una gran labor manteniéndolos bajo seguimiento a lo largo del video, y no perderlos, o perdiéndolos y recuperándolos instantáneamente, a pesar de la evolución de la perspectiva. Se puede reproducir el video siguiendo este enlace:

[https://drive.google.com/file/d/1yymRsXq09ApZBgShoowH01P2cB1cd8Zc/view?usp=share\\_link](https://drive.google.com/file/d/1yymRsXq09ApZBgShoowH01P2cB1cd8Zc/view?usp=share_link)



**Figura 4-8 Primeros instantes análisis RIMPAC\_1 [elaboración propia]**

El buque portaaviones al que se le asigna la etiqueta nº 3 (id:3), es un buque australiano de la clase “Canberra”, construido por la empresa española Navantia, y que comparte diseño con el buque L-61 incluido en el *dataset*. Esto facilita significativamente su clasificación como “Aircraft\_Carrier” por parte del modelo, llegando incluso a identificar parte de la cubierta de vuelo, como se ve en la Figura 4-9. Los otros tres portaaviones del video, que se van siguiendo aguas en una formación en línea, no tienen patrones que hayan sido estudiados durante el aprendizaje del modelo. Por ello, de esos tres buques solo es capaz de detectar a los dos más cercanos, y su clasificación, aunque acertada, tiene un porcentaje de certeza pequeño, llegando, durante el video, a confundir al más lejano con una fragata.



**Figura 4-9 RIMPAC\_1 [elaboración propia]**

Lo mismo sucede con las fragatas; son varias las que están presentes en el video, pero el modelo tan solo es capaz de detectar a las dos más cercanas, confundiendo a la más alejada de esas con otras clases de buques. Aun así, el resultado obtenido es satisfactorio.

El segundo video analizado de estos ejercicios se encuentra en este enlace:

[“https://drive.google.com/file/d/1cCxQxJ4k6STpNgxXZmD3071ZEdeErpmh/view?usp=share\\_link”](https://drive.google.com/file/d/1cCxQxJ4k6STpNgxXZmD3071ZEdeErpmh/view?usp=share_link)

Este video ofrece una perspectiva algo más cercana a la formación de los buques. Como se puede ver en la Figura 4-10, el modelo es capaz de identificar con relativa seguridad la fragata del primer plano,

clasificando incluso su cañón, aun estando a cierta distancia. Las embarcaciones en segundo plano, sin embargo, le resultan más complicadas de analizar.



Figura 4-10 RIMPAC\_2 [elaboración propia]

Inicialmente, el sistema identifica al portaaviones más cercano correctamente. No obstante, va alternando entre su clasificación correcta (*Aircraft\_Carrier*) y la clasificación de fragata (*Frigate*), hasta que finalmente deja de detectarlo, como se ve en la Figura 4-11. Así mismo, la fragata que hay detrás de ese mismo portaaviones, no la detecta en ningún momento. La otra fragata, que si detecta, no la encuadra adecuadamente en su caja delimitadora (id:5 en la Figura 4-10), y la acaba perdiendo como se ve en los últimos momentos del video, plasmados en la Figura 4-11. Hay una tercera fragata, detrás de esta última, pero se encuentra a una distancia considerable de la cámara.



Figura 4-11 Últimos instantes análisis RIMPAC\_2 [elaboración propia]

También hay que mencionar la presencia de una LCS (*Litoral Combat Ship*) estadounidense, de la clase *Independence*, un tipo de buque desconocido por completo para el modelo. Su equivalente más acertado es el de un BAM español, por lo que la etiqueta adecuada sería la de corveta (*Corvette*). Sin embargo, su porte se asimila más al de una fragata F-80, y su silueta a la de una fragata F-100, por lo que su clasificación como fragata también habría sido aceptable. El modelo, sin embargo, la clasifica como portaaviones, dándole el número 7, que se puede reconocer en la Figura 4-10 (id:7). Esto posiblemente se deba a su gran cubierta de vuelo, que asemeja el perfil del buque con el de un

portaaviones. Durante algunos momentos del video, como el representado en la Figura 4-11, el modelo llega a entrar en detalle, siendo capaz de identificar, en la fragata del primer plano, la cubierta de vuelo y lo que él clasifica como una antena de comunicaciones por satélite, además del cañón de proa mencionado anteriormente. El último buque en el video es el emergente portaaeronaves de la clase Canberra, al cual el sistema vuelve a detectar y clasificar correctamente con facilidad.

Tras el análisis de este último video, se da por finalizado el capítulo de obtención y análisis de resultados.

## 5 CONCLUSIONES Y LÍNEAS FUTURAS

### 5.1 Conclusiones

Las capacidades que presenta el constante desarrollo tecnológico presente en la actualidad, postulan la generación de una cantidad de posibilidades inabarcable. La principal conclusión de este TFG es que estas oportunidades han de ser explotadas en el ámbito naval, donde las aplicaciones de nuevas tecnologías pueden dar lugar al desarrollo de herramientas de gran utilidad. Especialmente en el ámbito bélico, se ha de perseguir el poseer siempre la ventaja tecnológica frente a los adversarios navales.

La inteligencia artificial constituye un campo que ofrece, aún, muchos rumbos de investigación, siendo una fuente de desarrollo de gran valor. Como se ha visto a lo largo de este trabajo, sus aplicaciones pueden constituir ayudas en multitud de ámbitos; tan diferentes como el análisis de radiografías y la seguridad en la navegación. Así mismo, se ha reflejado el cada vez mayor empleo, por parte de todos los sectores de la sociedad, de los drones. Tanto el mundo militar, que abarca el empleo de unidades tan dispares entre sí como que unas pueden ser sujetadas en la palma de la mano, mientras que otras se asemejan en tamaño a aeronaves convencionales, como el mundo civil, donde se emplean en ámbitos que comprenden desde la paquetería hasta la ingeniería, han sabido integrar en sus métodos de trabajo y formas de operar el uso de los drones, apreciando las capacidades que aportan las aeronaves tripuladas de forma remota. Estas capacidades, junto con las posibilidades que ofrece la inteligencia artificial, en este caso a través del aprendizaje profundo, es lo que ha buscado casar y compaginar este TFG para ofrecer una herramienta práctica para el mundo naval.

El principal objetivo de este trabajo se presentó como la construcción de un modelo de aprendizaje automático que posibilitase la localización y el seguimiento de objetos flotantes mediante el uso de drones con cámara. Tras el proceso de desarrollo, se ha conseguido obtener un modelo capaz de llevar a cabo estas tareas, pudiendo ejecutar un buen análisis de la situación naval en que se encuentra un buque, y aportando una herramienta que facilite las labores de los pilotos de drones embarcados, dando por satisfecho el objeto de este TFG. Una observación a remarcar es, que este modelo ha sido entrenado para emplearse en un ámbito específico, el naval, y dentro de este, ha sido focalizado en el mundo militar y las embarcaciones de guerra. Por tanto, se concluye que, en el caso de aplicar este sistema a un entorno distinto a este, los resultados obtenidos podrían ser nulos o poco precisos, necesitando un entrenamiento adicional para enseñar al modelo a analizar ese nuevo contexto.

Para construir este sistema, se han alcanzado los distintos hitos marcados como objetivos a cumplir en el proceso de desarrollo, durante el cual se han evidenciado factores de importancia que han influenciado el resultado. Mediante la creación, estudio previo, de un sistema similar, pero de menores prestaciones al modelo final desarrollado, se ha conseguido alcanzar una comprensión detallada de los fundamentos del aprendizaje profundo y la visión por computador, y se ha obtenido, así mismo, una

adecuada base práctica sobre la que ejecutar la construcción del modelo final. Por otro lado, el conformado de una base de datos capaz de cubrir los atributos necesarios para el correcto aprendizaje del modelo ha sido una parte esencial en el proceso de creación del modelo. Además, ha resultado de gran utilidad el proceso recurrente de reentrenamientos, mediante el cual se ha ido actualizando el modelo, empleándose, cada vez, la versión más reciente de este para analizar diferentes videos, y así, poder descubrir que carencias sufría la base de datos.

Respecto a las problemáticas que han surgido durante el desarrollo de este trabajo, como la incapacidad de realizar el etiquetado directamente sobre los videos, en vez de sobre imágenes, o el error de exportación generado por las etiquetas rotadas, se han podido solventar adecuadamente. Así mismo se ha podido superar el trance surgido durante el entrenamiento final del modelo, al perderse la conexión con los servidores de *Colab*. Por último, la integración del sistema a la plataforma *Coral AI* no se ha podido completar plenamente durante el plazo temporal otorgado para la entrega de este TFG. Se pretende, sin embargo, finalizar esta integración de forma inmediatamente posterior a la redacción de este trabajo.

## 5.2 Líneas futuras

Tras haber desarrollado un modelo satisfactorio que puede detectar y hacer seguimiento de un abanico de embarcaciones y objetos flotantes en la mar, y es capaz de distinguir de qué tipos de buques se trata y los equipos que portan, se presentan una serie de líneas de investigación y desarrollo que pueden contribuir a la mejora del modelo y a potenciar la línea general en la que se ha centrado este TFG. Las líneas futuras que se proponen son las siguientes:

- Ampliación del abanico de objetos flotantes que es capaz de identificar el modelo. Para potenciar su practicidad, se puede completar el *dataset* con datos sobre elementos que no han sido amparados en este TFG. Esta ampliación se puede plantear, por ejemplo, enfocándose en el ámbito de la seguridad para la navegación, con elementos que si no son maniobrados por las embarcaciones pueden causar averías, como palangres, bateas y boyas, u otros objetos que suponen un peligro mayor, como contenedores a la deriva o icebergs. Otro posible encaminamiento es el de la detección de personas en la mar, que podría resultar en una herramienta de gran utilidad para la búsqueda de naufragos.
- Integración del modelo generado en equipos operados por la Armada. Debido a que este modelo ha sido desarrollado con una evidente predisposición a ser empleado por parte de marinas de guerra, una posible línea de trabajo es la búsqueda de su despliegue en los distintos drones que opera la Armada.
- Potenciación de las capacidades del modelo para la detección y el seguimiento de embarcaciones rápidas y embarcaciones semisumergibles, para constituirse como una herramienta de utilidad en la lucha contra el tráfico ilícito de drogas. El narcotráfico es un problema de gran presencia en las zonas costeras españolas. En el sur peninsular es frecuente el avistamiento de lanchas rápidas provenientes de África, cargadas con sustancias ilícitas. En el norte, especialmente en las Rías Bajas, hay presencia de embarcaciones empleadas para la introducción de drogas provenientes de Sudamérica, destacando aquí el uso de semisumergibles, o “narcosubmarinos”, avistados con cada vez mayor frecuencia.
- Creación de un nuevo modelo de aprendizaje, sobre el mismo *dataset*, basado en posibles versiones futuras del algoritmo YOLO. Debido a la constante investigación que se lleva a cabo en torno al aprendizaje profundo y sus algoritmos, existe una alta probabilidad de que, en el futuro próximo, respecto a la redacción de este trabajo, se desarrolle una nueva versión del algoritmo YOLO.

## 6 BIBLIOGRAFÍA

- [1] «Fedón,» vol. Dialogos III, nº 65b, 2007.
- [2] H. C. King, *The history of the telescope*, Dover Publications, 1979.
- [3] J. E. Greivenkamp y D. Steed, *The 300-year quest for binoculars*, Optica Publishing Group, 2010.
- [4] P. Cunningham, M. Cord y S. J. Delany, «Supervised Learning,» de *Machine Learning Techniques for Multimedia*, vol. Supervised Learning, Berlín, Springer, 2008.
- [5] Z. Ghahramani, «Unsupervised Learning,» de *Advanced Lectures on Machine Learning*, Berlín, Springer, 2004.
- [6] L. P. Kaelbling, M. L. Littman y A. W. Moore, «Reinforcement Learning: A Survey,» *Journal of Artificial Intelligence Research*, vol. 4, 1996.
- [7] «Iberdrola,» [En línea]. Available: <https://www.iberdrola.com/innovacion/machine-learning-aprendizaje-automatico>. [Último acceso: 31 1 2023].
- [8] J. McCarthy, M. Minsky, N. Rochester y C. Shannon, *A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence*, 1955.
- [9] «The birthplace of artificial intelligence-Dartmouth Workshop in 1956,» Imedia, 1956. [En línea]. Available: <https://min.news/en/tech/c4f93f274f1cee02c23f04958d1d96e7.html>. [Último acceso: 20 2 2023].
- [10] G. Y. Sandesh, H. Pathak, Ranjan Mishra y Reddy Mishra, «The Understanding of Deep Learning: A Comprehensive Review,» vol. 2021, 2021.
- [11] NVIDIA, «Deep Learning,» 18 2 2015. [En línea]. Available: <https://developer.nvidia.com/deep-learning>. [Último acceso: 1 2 2023].
- [12] v7labs, «The Essential Guide to Neural Network Architectures,» [En línea]. Available: <https://www.v7labs.com/blog/neural-network-architectures-guide>, <https://www.v7labs.com/blog/neural-network-architectures-guide>. [Último acceso: 1 2 2023].

- [13] IBM, «What is Deep Learning? | IBM,» [En línea]. Available: <https://www.ibm.com/topics/deep-learning>. [Último acceso: 1 2 2023].
- [14] K. O'Shea y R. Nash, «An Introduction to Convolutional Neural Networks,» Aberystwyth University, Ceredigion, 2015.
- [15] Z. Li, F. Liu, W. Yang, S. Peng y J. Zhou, «A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects,» *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, nº 12, 2022.
- [16] v7labs, «What Is Computer Vision?,» [En línea]. Available: <https://www.v7labs.com/blog/what-is-computer-vision>, <https://www.v7labs.com/blog/what-is-computer-vision>. [Último acceso: 1 2 2023].
- [17] A. Farhadi y J. Redmon, «You only look once: Unified, real-time object detection,» *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.
- [18] R. Kundu, «YOLO Algorithm for Object Detection Explained,» 3 2 2023. [En línea]. Available: <https://www.v7labs.com/blog/yolo-object-detection>, <https://www.v7labs.com/blog/yolo-object-detection>. [Último acceso: 28 2 2023].
- [19] S. Karagiannakos, «YOLO - You only look once (Single shot detectors),» 13 5 2019. [En línea]. Available: <https://theaisummer.com/YOLO/>. [Último acceso: 28 2 2023].
- [20] J. Redmon, S. Divvala, R. Girshick y A. Farhadi, «You Only Look Once: Unified, Real-Time Object Detection,» Cornell University, Ithaca, NY, 2016.
- [21] J. Redmon y A. Farhadi, «YOLO9000: Better, faster, stronger,» *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
- [22] Ultralytics, «YOLOv8 Docs,» [En línea]. Available: <https://docs.ultralytics.com/>. [Último acceso: 28 2 2023].
- [23] M. Bednarek y M. Łysakowski, «A new YOLO is here - YOLOv8,» Flyps, 11 1 2023. [En línea]. Available: <https://www.flyps.io/blog/a-new-yolo-is-here-yolov8>. [Último acceso: 1 3 2023].
- [24] J. Buckley, *Air Power in the Age of Total War*, Routledge, 1998.
- [25] H. Rousseau y E. Esquis de Louis Figuier, 'Les Merveilles de la ciencia : Aerostats' (Maravillas de la Ciencia: Los globos de aire), París: Jouvet et Cie, 1868.
- [26] A. P. Yuste y M. S. Palma, «Scanning Our past from Madrid: Leonardo Torres Quevedo,» vol. 93, nº 7, 2005.
- [27] Getty Images, «V1 Flying bombs,» [En línea]. Available: <https://www.gettyimages.es/fotos/v-1-flying-bombs>. [Último acceso: 1 2 2023].
- [28] Y. Azoulai, «Unmanned combat vehicles shaping future warfare,» 2011.
- [29] IAI, «Scout Airplane Unmanned Maritime Patrol,» [En línea]. Available: <https://www.iai.co.il/p/scout>. [Último acceso: 23 2 2023].
- [30] M. Palik y M. Nagy, «Brief history of UAV development,» vol. 31, nº enero, 2019.
- [31] Ministerio de Fomento, «Plan estratégico para el desarrollo del sector civil de los drones en España 2018-2021,» Madrid, 2018.

- [32] F. Afghan, A. Razi, J. Chakareski y J. Ashdown, «Wildfire monitoring in remote areas using autonomous unmanned aerial vehicles,» 2019.
- [33] Ayuntamiento de Ciudad Real, «El Ayuntamiento utiliza drones para “mapear” la ampliación del Parque Forestal de La Atalaya,» [En línea]. Available: <https://www.ciudadreal.es/servicios-municipales/movilidad-y-transportes/movilidad-sostenible/caminos-escolares/22-noticias/medio-ambiente/17425-el-ayuntamiento-utiliza-drones-para-%E2%80%9Cmapear%E2%80%9D-la-ampliación-del-parque-forestal-de-la-atala>. [Último acceso: 31 1 2023].
- [34] Actualidad Aeroespacial, «Correos prueba los drones del proyecto Delorean,» 15 12 2022. [En línea]. Available: <https://actualidadaeroespacial.com/correos-prueba-los-drones-del-proyecto-delorean/>. [Último acceso: 31 1 2023].
- [35] Ministerio del Interior, Interviewee, *Tráfico distribuye los 39 drones que vigilarán las carreteras españolas este verano*. [Entrevista]. 6 julio 2021.
- [36] Agencia Tributaria, «Qué es Vigilancia Aduanera - Organización,» [En línea]. Available: <https://sede.agenciatributaria.gob.es/Sede/aduanas/vigilancia-aduanera/que-vigilancia-aduanera.html>. [Último acceso: 1 2 2023].
- [37] Departamento de Aduanas, «Lineas de acción,» Agencia Tributaria, Madrid.
- [38] Agencia Tributaria, «Qué es Vigilancia Aduanera - Medios personales y operativos,» [En línea]. Available: <https://sede.agenciatributaria.gob.es/Sede/aduanas/servicio-vigilancia-aduanera/que-vigilancia-aduanera/medios-personales-operativos.html>. [Último acceso: 7 2 2023].
- [39] Salvamento Marítimo, «87 Unidades marítimas y aéreas,» [En línea]. Available: <http://www.salvamentomaritimo.es/conocenos/nuestros-medios/87-unidades-maritimas-y-aereas>. [Último acceso: 7 2 2023].
- [40] Salvamento Marítimo, «Salvamento Marítimo por primera vez realiza vuelos de vigilancia de contaminación, búsqueda y rescate con un dron de la Agencia Europea de Seguridad Marítima en Huelva,» 6 2 2019. [En línea]. Available: <http://www.salvamentomaritimo.es/sala-de-comunicacion/sala-de-prensa/salvamento-maritimo-por-primera-vez-realiza-vuelos-de-vigilancia-de-contaminacion-busqueda-y-rescate-con-un-dron-de-la-agencia-europea-de-seguridad-maritima-en-huelva>. [Último acceso: 7 2 2023].
- [41] Salvamento Marítimo, «El Gobierno autoriza la compra pública de tecnología innovadora para Salvamento Marítimo por un valor estimado de 19 millones de euros,» 16 3 2021. [En línea]. Available: <http://www.salvamentomaritimo.es/sala-de-comunicacion/sala-de-prensa/el-gobierno-autoriza-la-compra-publica-de-tecnologia-innovadora-para-salvamento-maritimo-por-un-valor-estimado-de-19-millones-de-euros>. [Último acceso: 7 2 2023].
- [42] L. D. Castro, «Interior adquirirá 13 drones para la Policía Nacional por 324.000 euros,» Infodron, 19 3 2023. [En línea]. Available: <https://www.infodron.es/texto-diario/mostrar/4218000/interior-adquirira-13-drones-policia-nacional-324000-euros>. [Último acceso: 23 3 2023].
- [43] Policía, «Unidades Policiales - Medios Aéreos,» [En línea]. Available: [https://www.policia.es/\\_es/tupolicia\\_conocenos\\_eventos\\_detalle.php?tipo=5&subtipo=28#](https://www.policia.es/_es/tupolicia_conocenos_eventos_detalle.php?tipo=5&subtipo=28#). [Último acceso: 2 2 2023].
- [44] C. I. O. Cambeiro, «CBO1 Cambeiro al AF Álvarez sobre los drones de la Guardia Civil,» Unidad Pegaso, Guardia Civil, 2023.

- [45] Curso de Drones, «Usos de los drones en la Guardia Civil ¿Quieres ser piloto de drones?,» [En línea]. Available: <https://cursodedrones.es/usos-de-los-drones-en-la-guardia-civil/>. [Último acceso: 26 2 2023].
- [46] DJI, «DJI - Página oficial,» [En línea]. Available: <https://www.dji.com/es>. [Último acceso: 26 2 2023].
- [47] R. Olthoff, «Counter Unmanned Aircraft Systems (UAS) Threats,» Ministerie Van Defensie, 2021.
- [48] D. G. d. A. y. M. -. MINISDEF, «Iniciativa de I+D,» 1 7 2015. [En línea]. Available: <https://www.tecnologiaeinnovacion.defensa.gob.es/es-es/Contenido/Paginas/detalleiniciativa.aspx?iniciativaID=227>. [Último acceso: 7 2 2023].
- [49] Teledyne FLIR, «BLACK HORNET PRS DATASHEET,» [www.teledynelflir.com](http://www.teledynelflir.com), Kent, 2022.
- [50] Infodefensa, «El MOE y la Brigada Paracaidista reciben la última versión del mini dron Black Hornet,» [En línea]. Available: <https://www.infodefensa.com/texto-diario/mostrar/3471147/moe-brigada-paracaidista-reciben-ultima-version-mini-dron-black-hornet>. [Último acceso: 7 2 2023].
- [51] Infodron, «El MOE del Ejército de Tierra opera los UAV Hugin X1 y Blackhornet,» [En línea]. Available: <https://www.infodron.es/texto-diario/mostrar/3531100/moe-ejercito-tierra-opera-uav-hugin-x1-blackhornet>. [Último acceso: 8 2 2023].
- [52] Sky-Watch, «HUGINN X1 TECHNICAL SPECIFICATIONS,» [www.sky-watch.dk](http://www.sky-watch.dk), Støvring, 2015.
- [53] J. L. Expósito, «Reportaje Volar con los pies en el suelo,» vol. Revista Española de Defensa, nº Abril, 2018.
- [54] J. L. Expósito, «La hora de los Drones,» vol. Revista Española de Defensa, nº Abril, 2015.
- [55] Confidencial Digital, «El Wasp-AE, un dron clave para operaciones de desembarco de la Infantería de Marina,» 27 2 2017. [En línea]. Available: <https://www.elconfidencialdigital.com/articulo/defensa/Wasp-AE-operaciones-desembarco-Infanteria-Marina/20170224142913084537.html>. [Último acceso: 8 2 2023].
- [56] AeroVironment, «Wasp Datasheet,» AeroVironmen, 2023.
- [57] Conyca, «Sistemas Geodrone | Conyca,» [En línea]. Available: <https://www.conyca.es/sistemas-geodrone>. [Último acceso: 8 2 2023].
- [58] Vozpópuli, «Un Ejército de drones,» 4 2 2017. [En línea]. Available: [https://www.vozpopuli.com/actualidad/ejercito-drones\\_0\\_996201538.html](https://www.vozpopuli.com/actualidad/ejercito-drones_0_996201538.html). [Último acceso: 9 2 2023].
- [59] AeroVironment, «Raven\_Datasheet,» avinc, Arlington.
- [60] Defensa.com, «Los primeros RPAS Seeker en unidades de élite en las Fuerzas Armadas españolas,» 14 5 2021. [En línea]. Available: <https://www.defensa.com/espana/primeros-rpas-seeker-unidades-elite-fuerzas-armadas-espanolas>. [Último acceso: 5 2 2023].
- [61] Dronetools, «UAV VTOL CÓNDOR,» [En línea]. Available: <https://dronetools.es/>. [Último acceso: 7 2 2023].
- [62] SCR Sistemas de Control Remoto, «TUCAN, Light compact RPAS,» SCR, San Sebastián de los Reyes, 2023.
- [63] SCR Sistemas de Control Remoto, «SCR,» 24 6 2020. [En línea]. Available: <https://scrdrones.com/scr-recibe-la-primera-autorizacion-en-espana-para-volar-fuera-de-linea-de-vista-con-drones-de-mas-de-2-kg/>. [Último acceso: 8 2 2023].
- [64] Hispaviación, «Alpha Unmanned Systems vende dos helicópteros Alpha 800 UAV a las Fuerzas Armadas a través de la DGAM,» 17 12 2018. [En línea]. Available: <https://www.hispaviacion.es/alpha-unmanned>

- systems-vende-dos-helicopteros-alpha-800-uav-las-fuerzas-armadas-traves-la-dgam/. [Último acceso: 8 2 2023].
- [65] ALPHA Unmanned Systems, «ALPHA 800 DATASHEET,» [www.alphaunmannedsystems.com](http://www.alphaunmannedsystems.com), Madrid.
- [66] THALES, «FULMAR X, Sistema Aéreo no Tripulado (UAS),» [www.thalesgroup.com](http://www.thalesgroup.com), Madrid, 2023.
- [67] SCR Sistemas de Control Remoto, «ALBA Blanco Aéreo Ligero,» SCR, San Sebastián de los Reyes.
- [68] Ministerio de Defensa, «La Unidad de Blancos Aéreos en Canarias,» *Depeartamento de Comunicación del Ejército de Tierra*, nº 004, 2018.
- [69] Aeronautics, «Orbiter 3 STUAS Brochure,» Aeronautics GROUP, Yavne, Israel.
- [70] SCR Sistemas de Control Remoto, «ATLANTIC I, High-performance RPAS,» SCR, San Sebastián de los Reyes, 2023.
- [71] Hispaviación, «La empresa española SRC suministrará sus RPAS Atlantic y Tucán a las Fuerzas Armadas Españolas,» 19 12 2017. [En línea]. Available: <https://www.hispaviacion.es/la-empresa-espanola-src-suministrara-rpas-atlantic-tucan-las-fuerzas-armadas-espanolas/>. [Último acceso: 8 2 2023].
- [72] Aeronaves Militares Españolas, «SIVA (Sistema integrado de vigilancia aérea),» 25 2 2017. [En línea]. Available: <https://aeronavesmilitaresespanolas.com/siva-ejercito-de-tierra/>. [Último acceso: 14 2 2023].
- [73] Ejército de Tierra, «Sistema integrado de vigilancia aérea (SIVA),» [En línea]. Available: [https://ejercito.defensa.gob.es/materiales/vehiculo\\_aereo\\_no\\_tripulado/SIVA.html](https://ejercito.defensa.gob.es/materiales/vehiculo_aereo_no_tripulado/SIVA.html). [Último acceso: 14 2 2023].
- [74] Aeronaves Militares Españolas, «SEARCHER MK.III,» 25 2 2017. [En línea]. Available: <https://aeronavesmilitaresespanolas.com/searcher-mk-iii-ejercito-de-tierra/>. [Último acceso: 8 2 2023].
- [75] Defensa.com, «El Ejército vuela en pruebas el Searcher MK II para obtener la certificación que le permita operarlo en territorio nacional,» 18 9 2015. [En línea]. Available: <https://www.defensa.com/espana/ejercito-vuela-pruebas-searcher-mk-ii-para-obtener-certificacion>. [Último acceso: 7 2 2023].
- [76] Ejército de Tierra, «Plataforma aérea sensorizada de inteligencia (PASI),» [En línea]. Available: [https://ejercito.defensa.gob.es/materiales/vehiculo\\_aereo\\_no\\_tripulado/PASI.html](https://ejercito.defensa.gob.es/materiales/vehiculo_aereo_no_tripulado/PASI.html). [Último acceso: 9 2 2023].
- [77] Infodefensa, «Sirtap, el dron que sustituirá al Searcher de España y al Hermes 450 en Colombia,» [En línea]. Available: <https://www.infodefensa.com/texto-diario/mostrar/4098078/sirtap-dron-sustituira-searcher-espana-hermes-450-colombia>. [Último acceso: 7 2 2023].
- [78] Ejercito del Aire y del Espacio, «Ejército del Aire - Aeronaves - Detalle aeronave,» [En línea]. Available: <https://ejercitodelaire.defensa.gob.es/EA/ejercitodelaire/es/aeronaves/avion/MQ-9-Predator-B/>. [Último acceso: 7 2 2023].
- [79] Aeronaves Militares Españolas, «MQ-9 PREDATOR B,» [En línea]. Available: <https://aeronavesmilitaresespanolas.com/mq-9-predator/>. [Último acceso: 7 2 2023].
- [80] Defensa.com, «Los Predator B del Ejército del Aire serán armados,» 31 1 2023. [En línea]. Available: <https://www.defensa.com/espana/espana-armara-predator-b-ejercito-aire-espacio>. [Último acceso: 7 2 2023].

- [81] Armada, «Creación de la duodécima escuadrilla de la flotilla de aeronaves,» 21 2 2023. [En línea]. Available: [https://armada.defensa.gob.es/ArmadaPortal/page/Portal/ArmadaEspannola/conocenosnoticias/prefLang-es/00noticias--2023--02--NT-021-12-ESCUADRILLA-es?\\_selectedNodeID=5661118&\\_pageAction=selectItem](https://armada.defensa.gob.es/ArmadaPortal/page/Portal/ArmadaEspannola/conocenosnoticias/prefLang-es/00noticias--2023--02--NT-021-12-ESCUADRILLA-es?_selectedNodeID=5661118&_pageAction=selectItem). [Último acceso: 25 2 2023].
- [82] Aeronaves Militares Españolas, «SCAN EAGLE,» [En línea]. Available: <https://aeronavesmilitaresespanolas.com/scan-eagle-armada-espanola/>. [Último acceso: 2 2 2023].
- [83] Armada, «11ª Escuadrilla Aeronaves - Armada - Ministerio de Defensa - Gobierno de España,» [En línea]. Available: <https://armada.defensa.gob.es/ArmadaPortal/page/Portal/ArmadaEspannola/buquesaeronaves/prefLang-es/03flotilla-aeronaves-aeronaves-no-tripuladas>. [Último acceso: 2 2 2023].
- [84] Aeronaves Militares Españolas, «El Scan Eagle en la «Operación Atalanta»,» 15 12 2022. [En línea]. Available: <https://aeronavesmilitaresespanolas.com/2020/12/22/el-scan-eagle-en-la-operacion-atalanta/>. [Último acceso: 2 2 2023].
- [85] SCR Sistemas de Control Remoto, «SCRAB II,» [En línea]. Available: <https://scrdrones.com/producto-item/scrab-ii/>. [Último acceso: 2 2 2023].
- [86] Capitán de Corbeta M. Perales Garat, «La Armada Tiene un Tesoro. Presente y Futuro de la ABTN,» vol. Revista General de Marina, nº Julio, 2014.
- [87] Marine Instruments, «M5D AIRFOX Main Features,» Nigrán, 2022.
- [88] conda.io, «Glossary — conda 23.1.0 documentation,» [En línea]. Available: <https://docs.conda.io/projects/conda/en/stable/glossary.html#miniconda-glossary>. [Último acceso: 4 2 2023].
- [89] Label Studio, «Open Source Data Labeling,» [En línea]. Available: <https://labelstud.io/>. [Último acceso: 4 2 2023].
- [90] Coral, «Coral,» [En línea]. Available: <https://coral.ai/>. [Último acceso: 6 2 2023].
- [91] MeteoGalicia, «Cámaras Web - MeteoGalicia,» [En línea]. Available: [https://www.meteogalicia.gal/observacion/camarasweb/camarasWeb.action?request\\_locale=es](https://www.meteogalicia.gal/observacion/camarasweb/camarasWeb.action?request_locale=es).
- [92] J. Olivera, «La Armada Española se destacó en los ejercicios Escudo Formidable 2017 de la OTAN,» [En línea]. Available: <https://www.zona-militar.com/2017/10/17/la-armada-espanola-se-destaco-los-ejercicios-escudo-formidable-2017-la-otan/>. [Último acceso: 8 3 2023].
- [93] Foro Naval, «Foro Naval – Página 6,» 18 11 2022. [En línea]. Available: <https://foronaval.com/category/naval-2/>. [Último acceso: 8 3 2023].
- [94] Sea Forces, «Mk-13 Guided Missile Launching System GMLS,» [En línea]. Available: <https://www.seaforces.org/wpnsys/SURFACE/Mk-13-missile-launcher.htm>. [Último acceso: 8 3 2023].

## ANEXO I: IMPLICACIONES SOCIALES, ECONÓMICAS, Y AMBIENTALES

La construcción de un modelo de aprendizaje profundo que localiza y hace seguimiento de objetos flotantes, empleando drones con cámara, tiene diversas implicaciones sociales, económicas y ambientales.

Respecto al ámbito social, este proyecto se enfoca en la importancia de la seguridad ciudadana, específicamente la seguridad de las personas en la mar, independientemente de cualquier posible distinción entre ellas. Como se planteó en la introducción, este sistema sirve de herramienta para que los barcos comprendan mejor su entorno marítimo, y así, entre otras cosas, evitar accidentes. También, en las líneas futuras, se plantea el posible fomento de este empleo del modelo, proponiendo un trabajo de focalización en la detección de objetos flotantes que supongan peligros de mayor frecuencia o riesgo para los barcos. Así mismo, se propone el desarrollo de una base de datos que permita detectar a naufragos y poder facilitar sus rescates, pudiéndose también emplear para la localización de balsas salvavidas a la deriva. Otro criterio valorado, sobre la implicación que tiene este TFG respecto a este ámbito, es el planteamiento del cese del empleo de aeronaves tripuladas para misiones de riesgo que busquen la obtención de inteligencia naval. En este trabajo, se desarrolla una alternativa para disminuir la necesidad de arriesgar las vidas de pilotos que deban reconocer objetivos navales; buscando priorizar la seguridad, se está más en consonancia con las perspectivas actuales de la sociedad respecto a la importancia de toda vida.

En cuanto a las implicaciones económicas, el mayor factor de relevancia de este proyecto es la defensa de los drones como herramientas suficientemente capacitadas como para llevar a cabo tareas que se han realizado tradicionalmente por aeronaves convencionales. La sustitución de estas aeronaves, ya sean aviones o helicópteros, por aeronaves tripuladas remotamente, supone una diferencia económica significativa. Los aviones y helicópteros empleados en las tareas de identificación y seguimiento de objetivos conforman plataformas de alto coste de desarrollo, adquisición, mantenimiento y operación. Tanto las marinas militares como las civiles hacen uso de estos medios a día de hoy, invirtiendo grandes cantidades de recursos. Mediante la sustitución de estas aeronaves por drones, sobre los que aplicar el modelo desarrollado en este TFG para la detección, clasificación y seguimiento de objetos en la mar, se pueden conseguir reducciones de gasto exponenciales. Tanto el precio de compra, como el costo de operación de una aeronave no tripulada (generalmente de propulsión eléctrica), suelen situarse muy por debajo que los de sus análogos tripulados. Respecto a los de mayor envergadura y prestaciones, el desarrollo de los drones supuso en un comienzo una gran carga económica, pero a día de hoy, gracias al avance tecnológico alcanzado, no resulta mucho más caro que el de los aviones que presenten capacidades similares.

En cuanto a la concienciación ambiental, este proyecto vuelve a destacar por su búsqueda de la sustitución de aeronaves convencionales por drones. Las primeras emplean combustibles fósiles altamente contaminantes, mientras que los segundos suelen ser de propulsión eléctrica, que, si recargados mediante energías renovables, pueden suponer la práctica eliminación de emisiones de partículas contaminantes para el medioambiente. Así mismo, otro aspecto encuadrable en la conservación del medioambiente, es el posible empleo que se le puede dar al modelo desarrollado de detección de objetos flotantes en la mar. Este sistema, mediante una ligera ampliación del *dataset* y un reentrenamiento, puede constituirse como una herramienta perfectamente capaz de asistir en la localización, para su limpieza, de basuras en mares y océanos; puede emplearse para detectar y hacer seguimiento de plásticos contaminantes, vertidos de petróleo, u otros residuos que puedan dañar el ecosistema marino.



## **ANEXO II: REFLEXIONES ÉTICAS Y SOCIALES**

El trabajo desarrollado plantea varias inquietudes éticas y sociales. Por un lado, el sistema de análisis de video construido en este TFG puede generar preocupaciones sobre la privacidad de las personas que puedan ser captadas por la cámara del dron. En esta misma línea, surge la posibilidad de que se utilice el modelo para fines ilegítimos, como el acoso o la vigilancia sin consentimiento de personas, en este caso mediante el seguimiento de las embarcaciones en las que se encuentran. Un dron con cámara de vídeo es una herramienta de gran utilidad, pero de entre los muchos empleos que se le pueden dar, siempre existe la posibilidad de que alguno de estos no sea bien intencionado. El modelo construido en este trabajo constituye un añadido más a las capacidades de un dron, lo que puede resultar en un aumento de las facilidades que puede tener un avieso individuo para cometer sus fechorías. Estas cuestiones plantean interrogantes sobre el alcance de la legislación existente en este ámbito, de su suficiencia regulatoria, o de la posible idoneidad de su revisión.

Por otro lado, una herramienta de obtención de inteligencia, como se plantea el modelo de este trabajo, constituye un medio que puede resultar responsable indirecto de ataques o atentados. Mediante el modelo creado, se pueden identificar distintos equipos y sistemas de defensa de buques militares, generando esto información de gran valor operacional, y que, tras su análisis, puede resultar en la decisión de efectuar una acción violenta contra la embarcación estudiada, conllevando esto posibles pérdidas de vidas humanas.

Estos factores dan un amplio margen de reflexión. Por ello, tras la ponderación de las implicaciones éticas y sociales que conlleva el sistema creado en este TFG, se concluye que ha de ser una prioridad el trabajar para garantizar que este se utilice de manera responsable y justa. Se ha de evitar que el modelo desarrollado sea empleado por individuos u organizaciones de naturaleza incierta. Así mismo, es importante garantizar que el modelo sea transparente en su funcionamiento y que se respeten las leyes y regulaciones vigentes en cuanto a la recolección, almacenamiento y uso de datos personales. Solo de esta manera se puede asegurar que el uso de tecnologías como la inteligencia artificial y el aprendizaje profundo contribuya al bienestar de la sociedad y no se convierta en una amenaza para los derechos humanos.



## ANEXO III: CATÁLOGO DE SIGLAS

- YOLO: *You Only Look Once* (solo se mira una vez)
- IA: Inteligencia Artificial
- ANN: *Artificial Neural Network* (Red Neuronal Artificial)
- CNN: *Convolutional Neural Network* (Red Neuronal Convolucional)
- VANT: Vehículo Aéreo no Tripulado
- UAV: *Unmanned Aerial Vehicle* (Vehículo Aéreo no Tripulado)
- RPAS: *Remotely Piloted Aerial System* (Sistema Aéreo Pilotado Remotamente)
- IAI: *Israel Aerospace Industries*
- DGT: Dirección General de Tráfico
- SVA: Servicio de Vigilancia Aduanera
- AEAT: Agencia Estatal de Administración Tributaria
- EMSA: *European Maritime Safety Agency* (Agencia Europea de Seguridad Marítima)
- CNP: Cuerpo Nacional de Policía
- UAS: *Unmanned Aerial System* (Sistema Aéreo no Tripulado)
- GAO: Grupo de Apoyo Operativo
- UEI: Unidad Especial de Intervención
- UCO: Unidad Central Operativa
- OTAN: Organización del Tratado del Atlántico Norte
- MALE: *Medium Altitude Long Endurance*
- HALE: *High Altitude Long Endurance*
- FFAA: Fuerzas Armadas
- ET: Ejército de Tierra
- EA: Ejército del Aire y del Espacio
- FGNE: Fuerza de Guerra Naval Especial
- MOE: Mando de Operaciones Especiales
- UME: Unidad Militar de Emergencias
- TEAR: Tercio de Armada
- GAIL: Grupo de Artillería de Información y Localización
- GPS: *Global Positioning System* (Sistema de Posicionamiento Global)
- SIVA: Sistema Integrado de Vigilancia Aérea
- INTA: Instituto Nacional de Técnicas Aeroespaciales
- PASI: Plataforma Aérea Sensorizada de Inteligencia
- SIRTAP: Sistema Remotamente Tripulado de Altas Prestaciones
- GMTI: *Ground Moving Target Indicator* (Indicador de Objetivos Terrestres Móviles)

- FLOAN: Flotilla de Aeronaves
- ISR: *Intelligence, Surveillance and Reconnaissance*
- API: *Application Programming Interface* (Interfaz de Programación de Aplicaciones)
- GPU: *Graphics Processing Unit* (Unidad de Procesamiento Gráfico)
- TPU: *Tensor Processing Unit* (Unidad de Procesamiento Tensorial)
- TXT: *Textfile*
- IoU: *Intersection over Union* (Intersección sobre Unión)
- mAP: *Mean Average Precision* (Valor Medio de Precisión Media)
- ESSM: *Evolved Sea Sparrow Missile*
- COCO: *Common Objects in Context*
- JSON: *JavaScript Object Notation*
- BAM: Buque de Acción Marítima
- USB: *Universal Serial Bus*
- RIMPAC: *Rim of the Pacific*
- LCS: *Litoral Combat Ship*

## ANEXO IV: SCRIPT DE ORGANIZACIÓN DE IMÁGENES

```
# -*- coding: utf-8 -*-
import sys, traceback, time, os, glob, shutil, zipfile
import numpy as np
from configuration import configure

##### TRACK CONFIGURATION #####
EXPORT_PROJECT_NAME = "project.zip" #Nombre del archivo descargado de Label
Studio con todas las imágenes etiquetadas"

#####

if __name__ == '__main__':
    try:
        print('***** EXPORT LABEL STUDIO TO YOLO
        *****')

        # Configure
        configuration = configure()

        # Read images
        images = glob.glob(configuration['images_label_studio_path'] + '*.jpg')
        num_images = len(images)

        if num_images == 0:
            print ('Formating from Label Studio. Empty source images : ' ,
configuration['images_label_studio_path'])
            sys.exit(0)

        print ('Formating from Label Studio. Number of images: ' , num_images)

        # Remove YOLO destiny directory and create new template
        shutil.rmtree(configuration['images_yolo_path'])
        shutil.copytree(configuration['templates_yolo_path'],
configuration['images_yolo_path'])

        # Unzip project zip file
        project_file_path = configuration['export_path'] + EXPORT_PROJECT_NAME
        project_yolo_path = configuration['export_path'] + 'YOLO/'
        os.makedirs(project_yolo_path, exist_ok=True)
        shutil.rmtree(project_yolo_path)

        yolo_labels_path = project_yolo_path + 'labels/'
        #print('project_file_path=',project_file_path)
        #print('yolo_labels_path=',yolo_labels_path)
        #shutil.unpack_archive(project_file_path)
        with zipfile.ZipFile(project_file_path, 'r') as zip_ref:
            zip_ref.extractall(project_yolo_path)
        labels = glob.glob(yolo_labels_path + '*.txt')
        num_labels = len(labels)
        print ('Formating from Label Studio. Number of labels: ' , num_labels)

    print('=====')
    image_counter = 1
    for image_path in images:
        image_name = os.path.basename(image_path)
        label_name = image_name.replace("jpg", "txt")
```

```

label_path = yolo_labels_path + label_name

if not (os.path.isfile(label_path)):
    image_reset_path = configuration['images_reset_path'] + image_name
    print('*****')
    print('LABEL NOT FOUND FOR image_path=', image_path, ' moving to
: ', image_reset_path)
    print('*****')
    shutil.move(image_path, image_reset_path)
    continue

prob = np.random.rand()

if prob < 0.6:
    target_path = 'train/'
elif prob < 0.9:
    target_path = 'valid/'
else:
    target_path = 'test/'

image_target_path = configuration['images_yolo_path'] + target_path +
'images/'
label_target_path = configuration['images_yolo_path'] + target_path +
'labels/'

print('-----', str(image_counter), '-----')
image_counter += 1
print('image_path=', image_path)
#print('label_path=', label_path)

#print('image_target_path=', image_target_path)
#print('label_target_path=', label_target_path)

shutil.copy2(image_path, image_target_path)
shutil.copy2(label_path, label_target_path)

# Remove temporal directory
shutil.rmtree(project_yolo_path)
except KeyboardInterrupt:
    print('KEYBOARD INTERRUPT !!! ', str(traceback.format_exc()))
    sys.exit(0)
except:
    print('MAIN ERROR : ', str(traceback.format_exc()))
    sys.exit(0)

```

## ANEXO V: SCRIPT DE EXTRACCIÓN DE IMÁGENES

```
# -*- coding: utf-8 -*-
import sys, traceback, os, shutil
from ultralytics import YOLO
from configuration import configure
from functions import video_to_images

##### TRACK CONFIGURATION #####

IMPORT_VIDEO_NAME = "video.mov" #"Nombre del video"
IMPORT_IMAGE_NAME = "imagen" #"Prefijo del nombre con que se guardarán las
imágenes"
IMPORT_IMAGE_SIZE = 800
IMPORT_FRAME_START = 1
IMPORT_FRAME_INTERVAL = 20
IMPORT_NUM_IMAGES_MAX = 10
IMPORT_MODEL_NAME = "barcos_armada_yolo8n.pt" #"Nombre del modelo"
IMPORT_SAVE_ANNOTATIONS = True

#####
if __name__ == '__main__':
    try:
        # Configure
        configuration = configure()

        # Create and train model
        video_path = configuration['videos_dataset_path'] + IMPORT_VIDEO_NAME
        print('video_path=', video_path)
        annotation_model_path = configuration['models_path'] +
configuration['model_version'] + "/" + IMPORT_MODEL_NAME
        images_path = configuration['images_import_path'] + IMPORT_IMAGE_NAME +
"/"

        print('images_path=', images_path)
        if (os.path.isfile(images_path)):
            shutil.rmtree(images_path)
        os.makedirs(images_path, exist_ok=True)
        video_to_images(video_path = video_path, images_path = images_path,
tasks_path = configuration['tasks_label_studio_path'],
        image_base_name = IMPORT_IMAGE_NAME, frame_interval =
IMPORT_FRAME_INTERVAL, frame_start = IMPORT_FRAME_START,
        num_images_max = IMPORT_NUM_IMAGES_MAX, save_annotations =
IMPORT_SAVE_ANNOTATIONS, annotation_model_path = annotation_model_path, labels =
configuration['labels'],
        image_size = IMPORT_IMAGE_SIZE)
    except KeyboardInterrupt:
        print('KEYBOARD INTERRUPT !!! ', str(traceback.format_exc()))
        sys.exit(0)
    except:
        print('MAIN ERROR : ', str(traceback.format_exc()))
        sys.exit(0)
```



## ANEXO VI: SCRIPT DE ENTRENAMIENTO DEL MODELO

```
# -*- coding: utf-8 -*-
import sys, traceback
from ultralytics import YOLO
from configuration import configure

##### TRAIN CONFIGURATION #####
TRAIN_MODEL_NAME = "yolo8n.pt" #"Nombre del modelo inicial"
TRAIN_EPOCHS = 1000
TRAIN_RUN_NAME = "BARCOS_ARMADA_YOLO8N" #"Nombre del nuevo modelo"
TRAIN_DEVICE = "0" #"cpu" #
TRAIN_IMAGE_SIZE = 800

#####

if __name__ == '__main__':
    try:
        print('***** TRAIN YOLO8 WITH DATASET *****')

        # Configure
        configuration = configure()

        # Create and train model
        model_path = configuration['models_path'] + configuration['model_version']
+ "/" + TRAIN_MODEL_NAME
        print('model_path=', model_path)
        model = YOLO(model_path)
        yaml_source = configuration['images_yolo_path'] + "data.yaml"
        #print('yaml_source=', yaml_source)
        model.train(data = yaml_source, epochs = TRAIN_EPOCHS, imgsz =
TRAIN_IMAGE_SIZE, device = TRAIN_DEVICE, patience = 50,
        batch = configuration['train_batch_size'], save = True, project =
configuration['path'] + "RUNS/TRAIN/", name = TRAIN_RUN_NAME)

    except KeyboardInterrupt:
        print('KEYBOARD INTERRUPT !!! ', str(traceback.format_exc()))
        sys.exit(0)
    except:
        print('MAIN ERROR : ', str(traceback.format_exc()))
        sys.exit(0)
```



## ANEXO VII: SCRIPT DE EJECUCIÓN YOLOv8N

```
# -*- coding: utf-8 -*-
import sys, traceback
from ultralytics import YOLO
from configuration import configure

##### TRACK CONFIGURATION #####

TRACK_VIDEO_NAME = "video.mov" #"Nombre del video a analizar"
TRACK_SAVE_NAME = "video" #"Nombre del video ya analizado"
TRACK_SAVE_RESULTS = True # Opción de guardar o no el video analizado#
TRACK_MODEL_NAME = "barcos_armada_yolo8n.pt" #"Nombre del modelo"
TRACK_TRACKER_NAME = "bytetrack.yaml" #"Nombre del algoritmo de tracking, puede
ser Bytetrack o BoT-Sort (botsort.yaml)"#
TRACK_IMAGE_SIZE = 800

#####

if __name__ == '__main__':
    try:
        # Configure
        configuration = configure()

        # Create model
        model_path = configuration['models_path'] + configuration['model_version']
+ "/" + TRACK_MODEL_NAME
        #print('model_path=', model_path)
        model = YOLO(model_path)

        # Track video
        track_video_path = configuration['videos_dataset_path'] + TRACK_VIDEO_NAME
        #print('track_video_path=', track_video_path)
        model.track(source = track_video_path, show = True, save =
TRACK_SAVE_RESULTS, device = "cpu", project = configuration['path'] +
"RUNS/TRACK/",
                                name = TRACK_SAVE_NAME, imgsz = TRACK_IMAGE_SIZE,
tracker=TRACK_TRACKER_NAME)
    except KeyboardInterrupt:
        print('KEYBOARD INTERRUPT !!! ', str(traceback.format_exc()))
        sys.exit(0)
    except:
        print('MAIN ERROR : ', str(traceback.format_exc()))
        sys.exit(0)
```