



ULPGC
Universidad de
Las Palmas de
Gran Canaria

eii

ESCUELA DE
INGENIERÍA INFORMÁTICA

Trabajo Fin de Grado

DetECCIÓN DE OBJETOS EN ENTORNOS MARÍTIMOS CAPTURADOS CON UNA CÁMARA DUAL TÉRMICA-ÓPTICA

TITULACIÓN: Grado en Ingeniería Informática

AUTOR: Selene González Curbelo

TUTORIZADO POR:

Nelson Manuel Monzón López y Sergio Ballines Barrera

9 de julio de 2024

Agradecimientos

En primer lugar, quisiera expresar mi agradecimiento a mis tutores Nelson y Sergio por su orientación, paciencia, confianza y dedicación en el ámbito de la investigación y el mundo de la inteligencia artificial. Su apoyo y feedback me han permitido avanzar en este entorno.

A todo el equipo de Qualitas Artificial Intelligence Science (QAISC)¹ por permitirme realizar esta investigación, por su disposición y colaboración, además del ambiente de trabajo acogedor y motivador que han creado.

Quisiera también agradecer a la Plataforma Oceánica de Canarias (PLOCAN)² por el acceso a la cámara dual-térmica, la cual ha sido crucial para la recopilación de datos y entorno de pruebas reales.

Finalmente, a todas las personas que me han apoyado a lo largo de este proceso. A mi familia y amigos, por su constante apoyo y comprensión en los momentos más desafiantes.

¹<https://qaisc.com/>

²<https://plocan.eu/>

Resumen

La seguridad marítima es crucial dentro del contexto de la economía azul, que abarca el uso sostenible de los recursos oceánicos para el crecimiento económico, la mejora de los medios de vida y el empleo, al tiempo que se preserva la salud del ecosistema marino. Un entorno marítimo seguro permite la explotación responsable de recursos marinos, el desarrollo del turismo costero, la pesca sostenible y la energía marina que son sectores clave del sector económico.

Vigilar adecuadamente las rutas marítimas y hacer seguimiento de las embarcaciones mediante métodos de Inteligencia Artificial (IA) es de enorme relevancia para mantener esta seguridad. La IA nos permite analizar grandes volúmenes de datos en tiempo real, detectar patrones sospechosos y predecir posibles amenazas, mejorando así la capacidad de respuesta ante incidentes. Esto no solo protege las vidas humanas y los bienes transportados, sino que también asegura la continuidad del comercio y la explotación sostenible de los recursos marinos, pilares esenciales para el crecimiento de la economía azul.

Este Trabajo Fin de Grado (TFG) analiza la potencialidad de una cámara dual térmico-óptica para mejorar el problema de la vigilancia de rutas marítimas apoyados en las imágenes térmicas que genera. Las cámaras térmicas hoy en día desempeñan un papel crucial en contraste con las cámaras RGB, especialmente en condiciones climáticas adversas, baja visibilidad u oscuridad nocturna. Estos dispositivos pueden superar los desafíos comentados al detectar el calor emitido por los objetos, en vez de depender de la luz visible.

Este proyecto se ha realizado en colaboración con la empresa Qualitas Artificial Intelligence Science (QAISC) proporcionando recursos de I+D y el acceso a su red de cámaras. Agradecemos también la ayuda brindada por la Plataforma Oceánica de Canarias (PLOCAN) otorgando acceso al estudiante a la información y datos de la cámara instalada en el centro comercial El Mirador, en el municipio de Telde, localizado al noreste de Gran Canaria.

Abstract

Maritime security is crucial within the context of the blue economy, which encompasses the sustainable use of ocean resources for economic growth, improved livelihoods, and employment while preserving the health of the marine ecosystem. A secure maritime environment enables the responsible exploitation of marine resources, the development of coastal tourism, sustainable fishing, and marine energy, which are key sectors of the economic sector.

Monitoring maritime routes adequately and tracking vessels using Artificial Intelligence (AI) methods is of immense importance for maintaining this security. AI allows us to analyze large volumes of data in real-time, detect suspicious patterns, and predict potential threats, thereby enhancing the response capability to incidents. This not only protects human lives and transported goods but also ensures the continuity of trade and the sustainable exploitation of marine resources, essential pillars for the growth of the blue economy.

This Final Degree Project (TFG) analyzes the potential of a dual thermal-optical camera to improve the issue of maritime route surveillance supported by the thermal images that it generates. Thermal cameras play a crucial role today in contrast to RGB cameras, especially in adverse weather conditions, low visibility, or nighttime darkness. These devices can overcome the challenges mentioned by detecting the heat emitted by objects, rather than relying on visible light.

This project was carried out in collaboration with Qualitas Artificial Intelligence Science (QAISC), providing R& D resources and access to their camera network. We also appreciate the assistance provided by the Oceanic Platform of the Canary Islands (PLOCAN) for granting the student access to information and data from the camera installed at the El Mirador shopping center in the municipality of Telde, located in the northeast of Gran Canaria.

Índice general

1. Introducción	1
1.1. Contexto General	2
1.2. Motivación	3
1.3. Objetivos	4
1.4. Competencias específicas y aportaciones del trabajo	5
1.4.1. Competencias	5
1.4.2. Aportaciones	6
1.5. Desafíos	6
2. Estado actual	8
2.1. Fundamentos de una Red Neuronal	9
2.1.1. Red Neuronal	9
2.1.2. GAN (Generative Adversarial Network)	9
2.1.3. Traducción de imagen a imagen no emparejada	12
2.2. Tareas de Visión por Computador	12
2.2.1. Clasificación	12
2.2.2. Localización	12
2.2.3. Detección	12
2.2.4. Composición de imágenes	13
2.3. Transfer Learning	15
2.4. Modelos de Detección de Objetos	15
2.4.1. YOLO (You Only Look Once)	15
2.4.2. Faster-RCNN (Region-based Convolutional Neural Networks)	17
2.4.3. CenterNet	18
2.4.4. SSD	19
2.5. Conjunto de datos	19
2.6. Métricas de Evaluación de los modelos	21
2.7. Captura de imágenes	22
2.7.1. Cámara térmica	22
2.7.2. Cámara óptica	23
2.7.3. Radar	23
3. Metodología	24
3.0.1. Metodología Scrum	24

3.0.2.	Metodología Kanban	24
3.0.3.	Metodología Scrumban	25
4.	Recursos utilizados y Presupuesto	26
4.1.	Recursos	26
4.1.1.	Hardware	26
4.1.2.	Software	27
4.1.3.	Librerías	27
4.1.4.	Otras herramientas	28
4.2.	Presupuesto	29
4.2.1.	Coste de equipos físicos	30
4.2.2.	Coste personal	30
4.2.3.	Presupuesto final	30
5.	Desarrollo	31
5.1.	Sprint Zero: Análisis y selección del modelo	31
5.2.	Sprint 1: Captura de Imágenes de la Cámara Térmica	33
5.3.	Sprint 2: Evaluación Dataset COCO	35
5.4.	Sprint 3: Evaluación Inicial y Preparación del Modelo Base	36
5.4.1.	Análisis de Datos Iniciales	36
5.4.2.	Entrenamiento del Modelo Inicial	37
5.5.	Sprint 4: Distribución de Datos	40
5.6.	Sprint 5: Creación y Evaluación de Imágenes Sintéticas	45
5.6.1.	IRSships	45
5.7.	Sprint 6: Entrenamiento de GAN para la Generación de Imágenes	48
5.7.1.	CycleGAN	48
5.8.	Sprint 7: Técnicas de Composición de Imágenes	54
5.8.1.	Blended Diffusion	54
5.8.2.	OpenCV	56
5.8.3.	Libcom	59
5.8.4.	Superposición de Recortes	61
5.9.	Sprint 8: Generación de Datos mediante Inteligencia Artificial	66
5.9.1.	Aplicaciones Online	66
5.9.2.	GLIDE (Guided Language-to-Image Diffusion for Generation and Editing)	69
5.9.3.	Text-guided image-to-image generation	69
5.10.	Sprint 9: Dataset Combinado (Superposición de Recortes y Text-guided image-to-image generation)	74
5.11.	Sprint 10: Análisis Comparativo de Datasets	79
5.12.	Sprint 11: Evaluación de Modelos YOLO	84
5.12.1.	Comparativa Dataset Combinado	84
5.12.2.	Comparativa Dataset Combinado Modificado	87
6.	Conclusiones y trabajo futuro	91
6.1.	Conclusión Datasets	91

6.2. Conclusión Modelo YOLO	91
6.3. Trabajo Futuro	92
6.3.1. Generación de Entornos	92
6.3.2. Integración en el Sistema de la Empresa	92
6.3.3. Mejora de Imágenes	93
6.3.4. Exploración de Nuevas Distribuciones	93
6.3.5. <i>Hard Negative</i>	93

Índice de figuras

1.1.	Imagen del entorno del mirador con paleta de colores <i>Mode Raimbow</i>	3
1.2.	Imagen del entorno del mirador con paleta de colores <i>Mode White Hot</i>	3
1.3.	Imagen óptica y térmica con calima. Imagen obtenida del Mirador	3
1.4.	Vista de la zona del mirador. Imagen obtenida del Mirador	4
2.1.	Estructura básica de una red neuronal. Imagen obtenida de [22]	9
2.2.	Conjunto de datos no emparejados (<i>Unpaired data</i>) utilizados en CycleGAN. La columna X contiene imágenes de barcos de una cámara térmica, mientras que la columna Y contiene imágenes de barcos de una cámara óptica.	10
2.3.	Pérdida de consistencia cíclica (<i>Cycle Consistency Loss</i>) utilizada en CycleGAN. La imagen superior muestra la conversión de una imagen de RGB a térmica, mientras que la imagen inferior muestra la conversión de una imagen térmica a RGB	11
2.4.	Tareas de Visión por Computador	13
2.5.	Proceso de aplicación de Composición guiada por Texto. Primero, se muestra la imagen de fondo, seguida de la imagen de la máscara que se aplicará. Finalmente, se presenta el resultado final obtenido al combinar la imagen de fondo y la máscara mediante la técnica de <i>Blended Diffusion</i> . Imagen generada con [4]	14
2.6.	Ejemplo de composición guiada por texto. De izquierda a derecha: fondo, barco, máscara del barco y resultado de la Composición Guiada por Imágenes. Imagen generada por [36]	14
2.7.	Funcionamiento de YOLO. Imagen obtenida de [38]	16
2.8.	Rendimiento en el conjunto de datos de detección de objetos de <i>MS COCO</i> . Imagen obtenida de la publicación original de YOLOv9 [46]	17
2.9.	Estructura de Faster R-CNN. Imagen obtenida de [39]	18
2.10.	Arquitectura de CenterNet. Imagen obtenida de [11]	18
2.11.	Arquitectura de SSD. Imagen obtenida de [27]	19
2.12.	Ejemplos de imágenes de barcos contenidas en el Dataset generado	20
2.13.	Ejemplos de imágenes descartadas del conjunto de datos. Las dos imágenes de la izquierda fueron descartadas por ser muy similares, mostrando el mismo barco y fondo, lo que no aporta valor adicional al entrenamiento. La imagen de la derecha fue descartada por ser en blanco y negro en lugar de térmica. Imágenes obtenidas de [4].	20
2.14.	Estructura interna de una cámara térmica. Imagen obtenida de [13]	22

4.1. Pantalla de inicio del proyecto, con los distintos datasets a etiquetar en CVAT	28
4.2. Ejemplo de imagen etiquetada utilizando la herramienta CVAT	29
5.1. Resultados de inferencia con el dataset COCO con imágenes térmicas	35
5.2. Ejemplo de imágenes con el mismo barco y entorno, mostrando la similitud de imágenes dentro de un conjunto de datos. Imagen obtenida de [30]	36
5.3. Imagen en blanco y negro de un entorno marítimo. Imagen obtenida de [10] .	37
5.4. Imagen con barcos ocluidos. Imagen obtenida de [9]	37
5.5. Matriz de confusión del entrenamiento del dataset inicial con distribución aleatoria.	38
5.6. Resultados del entrenamiento del dataset inicial con distribución aleatoria. .	39
5.7. Detecciones realizadas por el modelo inicial con distribución aleatoria	40
5.8. Matriz de confusión del entrenamiento del dataset inicial con distribución de histograma	43
5.9. Resultados del entrenamiento del dataset inicial con distribución de histograma	43
5.10. Detecciones realizadas con el dataset inicial con distribución de histograma .	44
5.11. Ejemplo de Barco Ferry Armourique generado con [49]	45
5.12. Ejemplo de Barco Destroyer Akizuki generado con [49]	45
5.13. Máscara iceberg. Imagen obtenida de [49]	46
5.14. Máscara plataforma petrolífera. Imagen obtenida de [49]	46
5.15. Máscara montaña. Imagen obtenida de [49]	46
5.16. Máscara cielo. Imagen obtenida de [49]	46
5.17. Imágenes sintéticas generadas con apariencia térmica del entorno marítimo a una temperatura de seis grados. Imagen generada con IRSships [49]	47
5.18. Imágenes sintéticas generadas con apariencia térmica del océano a una temperatura de diecisiete grados. Imagen generada con IRSships [49]	47
5.19. Imagen sintética generada con apariencia térmica de un barco con temperatura de cuatro grados. Imagen generada con IRSships [49]	47
5.20. Imágenes de barcos generadas mediante CycleGAN [23] [52] de 1280x1280px.	49
5.21. Imágenes de barcos generadas mediante CycleGAN [23] [52] de 256x256px. .	50
5.22. Matriz de confusión del entrenamiento del dataset inicial y de las imágenes generadas con CycleGAN.	51
5.23. Resultados del entrenamiento del conjunto de datos inicial e imágenes generadas con CycleGAN.	52
5.24. Detecciones realizadas con el dataset inicial y CycleGAN.	53
5.25. Proceso de aplicación de <i>Blended Diffusion</i> . Primero, se muestra la imagen de fondo, seguida de la imagen de la máscara que se aplicará. Finalmente, se presenta el resultado final obtenido al combinar la imagen de fondo y la máscara mediante la técnica de <i>Blended Diffusion</i> . Imagen generada con el código publicado en [4].	54
5.26. Imágenes generadas con <i>Blended Diffusion</i> [4]	55
5.27. Barco a color. Imagen generado con [4]	56
5.28. Barco con sombra. Imagen generada con [4]	56
5.29. Resultado de fusionar dos imágenes del mismo tamaño con el algoritmo de Image Blending with OpenCV 3	57

5.30. Composición de imagen usando <code>cv2.copyMakeBorder()</code> y <code>addWeighted</code> . El Fondo muestra la imagen de fondo utilizada, mientras que el Barco muestra el objeto que se superpone. La imagen resultante combina estos elementos usando <code>addWeighted</code> después de aplicar <code>cv2.copyMakeBorder()</code> para replicar el borde del barco	58
5.31. Imagen compuesta haciendo uso de gaussian. Imagen generada con [36]	60
5.32. Imagen compuesta haciendo uso de Poisson. Imagen generada con [36]	61
5.33. Resultados de composición de imágenes de barcos sobre fondos marítimos . .	62
5.34. Matriz de confusión del entrenamiento del dataset inicial y Superposición de Recortes	63
5.35. Resultados del entrenamiento del dataset inicial y Superposición de Recortes	64
5.36. Inferencia del entrenamiento del dataset inicial y Superposición de Recortes .	65
5.37. Resultados de imágenes generadas con ChatGPT.	66
5.38. Resultados de imágenes con Runway ML	67
5.39. Resultados de imágenes generadas con LeonardoAI	67
5.40. Resultados de imágenes generadas con Artbreeder	68
5.41. Resultados de barcos generadas mediante GLIDE [34]	69
5.42. Resultados generados con Text-guided image-to-image generation [12] con distintos valores de strength y guidance scale. De arriba a abajo, se observa: un resultado similar a la imagen original, un resultado a color que no corresponde a una cámara térmica, y un resultado que no se asemeja al prompt proporcionado.	70
5.43. Imágenes de barcos generadas mediante Text-guided image-to-image generation [12]. Las imágenes situadas la izquierda son las originales, y las de la derecha el resultado del proceso	71
5.44. Matriz de confusión del entrenamiento del conjunto de datos inicial y generación de imágenes guiada por texto	72
5.45. Resultados del entrenamiento del conjunto de datos inicial y generación de imágenes guiada por texto	72
5.46. Inferencia del entrenamiento del conjunto de datos inicial y Generación de imagen a imagen guiada por texto	73
5.47. Matriz de confusión del entrenamiento del dataset inicial y Generación de imagen a imagen guiada por texto y Superposición de Recortes	74
5.48. Resultados del entrenamiento del dataset inicial y Text-guided image-to-image generation y Superposición de Recortes	75
5.49. Inferencia del entrenamiento del dataset inicial y Generación de imagen a imagen guiada por texto y Superposición de Recortes	76
5.50. Matriz de confusión del entrenamiento del dataset inicial y Text-guided image-to-image generation y Superposición de Recortes e imágenes añadidas	77
5.51. Resultados del entrenamiento del dataset inicial y la generación de imágenes de texto a imagen y superposición de recortes e imágenes añadidas	78
5.52. Inferencia del entrenamiento del dataset inicial y generación de imágenes guiadas por texto e imagen a imagen y superposición de recortes e imágenes adicionales	79

5.53. Barco detectado en primer plano con las cajas delimitadoras no ajustadas correctamente con el dataset Combinado Modificado	80
5.54. Barcos ocluidos que son detectados como un único barco con el dataset Text-guided image-to-image generation	82
5.55. Barco detectado en la zona del Mirador con situación de calima con el dataset Combinado	82
5.56. Muro detectado como barco zona del Mirador con el dataset Text-guided image-to-image generation	83
5.57. Ejemplos de falsos positivos en el dataset Inicial Histograma	84
5.58. Inferencia del entrenamiento del dataset inicial y Text-guided image-to-image generation y Superposición de Recortes con YOLOv7, YOLOv8 y YOLOv9. La imagen superior contiene calima, en la segunda barcos superpuestos y la tercera es un caso general	86
5.59. Inferencia del entrenamiento del dataset inicial y generación de imagen a imagen guiada por texto, con superposición de recortes modificada usando YOLOv7, YOLOv8 y YOLOv9. La imagen superior muestra calima, en la segunda se observan barcos superpuestos y en la tercera un caso general	89
6.1. Plataforma en el entorno marítimo del Mirador, detectada como barco (FP)	92

Índice de tablas

4.1. Especificaciones técnicas del módulo térmico utilizado en el sistema de detección. Los valores fueron obtenidos de Hikvision [18]	27
4.2. Características técnicas del módulo óptico complementario al módulo térmico. Los valores fueron obtenidos de Hikvision [18]	27
4.3. Se detallan los costos asociados a los componentes de hardware necesarios para el desarrollo del sistema de detección	30
4.4. Costo estimado del personal involucrado en el proyecto	30
4.5. Costo total del proyecto	30
5.1. Análisis de modelos de detección de objetos	33
5.2. Resultados de la Validación de YOLO con Diferentes Datasets	80
5.3. Resultados validación YOLO con el Dataset Combinado	85
5.4. Resultados de la Validación de diferentes YOLO con Dataset Combinado . .	87
5.5. Resultados de la validación de YOLO con el Dataset Combinado Modificado	88
5.6. Resultados de la Validación de diferentes YOLO con Dataset Combinado Modificado	90

Índice de Algoritmos

1.	Proceso para capturar imágenes en la cámara térmica de la zona del Mirador	34
2.	Preprocesamiento y División de Imágenes en Conjuntos de Entrenamiento y Validación mediante Histogramas	42
3.	Image Blending with OpenCV	57

Capítulo 1

Introducción

Este Trabajo de Fin de Grado (TFG) se centra en la implementación de un sistema de detección de barcos mediante una cámara dual térmica, con el objetivo de proporcionar una vigilancia más precisa y eficiente en un entorno marítimo. Cabe mencionar que, a pesar de que la cámara en sí es dual térmica, el proyecto se centra en la parte térmica. Dicha cámara puede detectar embarcaciones de forma eficaz incluso en condiciones de baja visibilidad, como por ejemplo durante la noche o en situaciones meteorológicas adversas. Esta capacidad es esencial en escenarios donde las condiciones pueden cambiar drásticamente y afectar a la seguridad de las operaciones marítimas.

El proyecto se lleva a cabo en colaboración con la empresa *Qualitas Artificial Intelligence Science S.A.* (QAISC) ¹, la *Plataforma Oceánica de Canarias* (PLOCAN) ² y el *Centro de Tecnologías de la Imagen* ³ (CTIM). Quienes proporcionaron los recursos necesarios para capturar imágenes y validar el sistema con datos reales. Esta colaboración ha sido fundamental para garantizar que la tecnología desarrollada no solo tiene una base teórica, sino también aplicable en condiciones reales, permitiendo mejorar la validez y eficacia del modelo propuesto en un entorno real.

El principal desafío que aborda este TFG radica en las limitaciones de los sistemas de detección convencionales, especialmente en situaciones donde la visibilidad se ve comprometida. Los sistemas tradicionales basados en cámaras RGB (*Red, Green, Blue*) pueden fallar en detectar objetos en condiciones de baja visibilidad o niebla densa; por otro lado, mediante el uso de una cámara térmica, el sistema propuesto supera estas limitaciones al detectar el calor emitido por las embarcaciones. Esto no solo aumenta la precisión de la detección, sino que también proporciona una herramienta eficaz para prevenir accidentes marítimos, que a menudo ocurren debido a la falta de visibilidad.

Una particularidad del enfoque adoptado en este proyecto es el uso de redes neuronales debido a la limitación en el acceso a los datos térmicos brutos de la cámara. En vez de trabajar con los datos térmicos directamente, se emplean imágenes que representan estas

¹<https://qaisc.com/>

²<https://plocan.eu/>

³<https://ctim.ulpgc.es/>

capturas térmicas. Esta restricción da como resultado una disminución de la precisión y riqueza de los datos, ya que las representaciones térmicas no contienen toda la información detallada que los datos térmicos crudos pueden proporcionar.

La motivación detrás de este trabajo no se limita a la necesidad de mejorar la seguridad en la zona de PLOCAN, donde las embarcaciones no deben estar ubicadas durante la noche, sino también a desarrollar un modelo generalizable. Este enfoque asegura que el sistema desarrollado tenga un impacto significativo en la seguridad marítima en diferentes partes del mundo, contribuyendo al avance tecnológico en el campo de la detección de objetos marítimos.

En resumen, este TFG no solo persige la innovación en la tecnología de detección de embarcaciones, sino también establecer un nuevo estándar en la seguridad marítima mediante la aplicación de tecnologías avanzadas y la colaboración efectiva entre instituciones académicas y centros de investigación especializados.

1.1. Contexto General

La capacidad de las cámaras térmicas para captar el calor emitido por los objetos favorece la elaboración de imágenes definidas de los cuerpos. Para facilitar la interpretación de las temperaturas captadas, se establecen distintos esquemas de colores. Uno de los más conocidos es la paleta *rainbow*, que destaca por su diversa gama de colores. Este tipo de paleta se utiliza con regularidad para detectar la temperatura en las personas o los animales. La Ilustración 1.1 presenta esta paleta en el entorno del Mirador. Sin embargo, es importante tener en cuenta la diferencia en la variabilidad de las temperaturas de los objetos en el entorno marítimo, que es significativamente menor que en las personas, lo que influye en la elección de la paleta de colores.

En cuanto al entorno marítimo, los esquemas simples se utilizan con mayor frecuencia, como se puede observar en la Ilustración 1.2. Las tonalidades más claras representan una mayor temperatura, mientras que las más oscuras, temperaturas más frías. La elección de la gama de colores es crucial para la comprensión precisa del entorno. Aunque la decisión final recae sobre el usuario, en entornos en constante cambio, como el marítimo, la paleta de tonalidades más simple, como es el caso de *white hot*, tiene sumamente una gran ventaja sobre el esquema *rainbow* en lo que respecta a la cantidad de información innecesaria que proyecta y su incómoda interpretación.

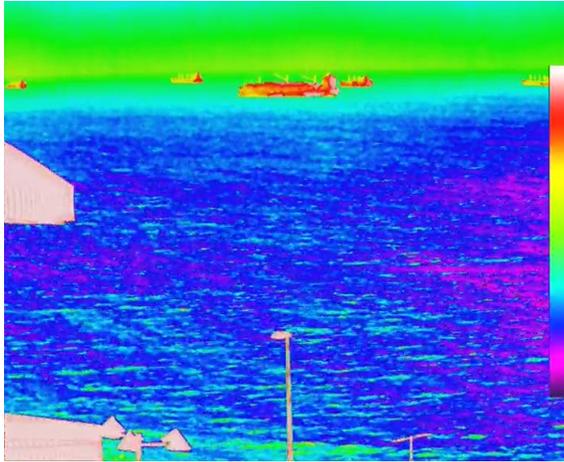


Ilustración 1.1: Imagen del entorno del mirador con paleta de colores *Mode Rainbow*



Ilustración 1.2: Imagen del entorno del mirador con paleta de colores *Mode White Hot*

Hay que tener en consideración que una cámara infrarroja puede operar en condiciones desafiantes para las cámaras ópticas, como la oscuridad total, precipitaciones, neblina o presencia de calima, la cual se muestra en la Ilustración 1.3, presentando una comparativa entre una imagen óptica y térmica afectada por calima, mostrando la visibilidad del barco en ambos casos. Dado que es una representación de la radiación térmica de los objetos, cuanto más lejos están, la señal puede perderse en el camino. Por lo tanto, hay que tener en cuenta la atenuación, donde hay que considerarla en estado de lluvia o niebla, en la cual la dispersión de las gotas de agua disminuirá la intensidad de la señal y afectará la capacidad de detección a distancias mayores.

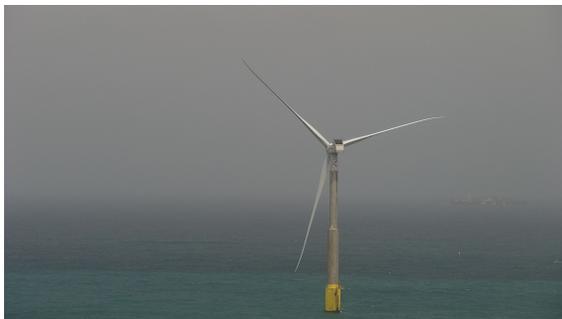


Ilustración 1.3: Imagen óptica y térmica con calima. Imagen obtenida del Mirador

1.2. Motivación

El Trabajo de Fin de Grado (TFG) se centra en el uso de una cámara térmica proporcionada por la Infraestructura Científica y Tecnológica Singular (ICTS) PLOCAN, para localizar

barcos en un ambiente marino, la cual se ubica en la azotea del Centro Comercial El Mirador. La Ilustración 1.4 ofrece una representación visual de la región a monitorizar.

La iniciativa se originó debido a la necesidad de localizar embarcaciones en dicha zona por la noche. Debido a que las instalaciones se encuentran cerradas por la noche, las embarcaciones no deben estar ubicadas en la zona de PLOCAN. Es por esto que se ha desarrollado este TFG. A pesar de esta motivación inicial, se pretende generar un modelo que vaya más allá de esta necesidad particular. De manera que permitirá contar con un entorno real que cumpla con las condiciones requeridas para el desarrollo del proyecto.

Además, se busca superar las limitaciones inherentes de los sistemas de detección convencionales en entornos marítimos desafiantes. Mejorar la precisión y eficiencia en la localización de barcos durante la noche no solo incrementa la seguridad, sino que también contribuirá al progreso tecnológico en este campo específico.

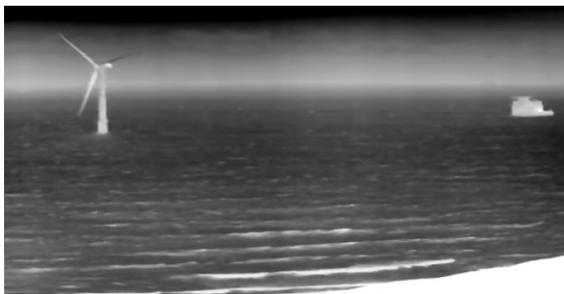


Ilustración 1.4: Vista de la zona del mirador. Imagen obtenida del Mirador

1.3. Objetivos

Mejorar la precisión y eficiencia en la detección de objetos en entornos marítimos es el principal objetivo de este proyecto. Para lograr este objetivo, se han establecido los siguientes objetivos:

1. Realizar pruebas exhaustivas en entornos reales con diferentes condiciones climáticas y visibilidad reducida para evaluar la robustez y la eficacia del sistema propuesto.
2. Profundizar en los conocimientos adquiridos por el alumno en lenguaje Python, utilizando este lenguaje de programación para el desarrollo e implementación del sistema.
3. Adquirir y procesar los datos necesarios para el entrenamiento y la evaluación del modelo de detección de embarcaciones.
4. Desarrollar y entrenar un modelo de detección de embarcaciones utilizando técnicas de aprendizaje profundo, con el objetivo de mejorar la precisión y la capacidad de detección en entornos marítimos.
5. Integrar el modelo desarrollado en un sistema completo de detección de objetos y evaluar su desempeño en diferentes condiciones de iluminación y visibilidad.

6. Documentar adecuadamente el proceso de desarrollo, implementación y evaluación del sistema, proporcionando así una referencia útil para investigadores y profesionales interesados en este campo.

Para ello, se realizaron pruebas exhaustivas en diferentes condiciones, especialmente con calima y visibilidad reducida en la zona del Mirador, lo que permitió evaluar la robustez y efectividad del sistema en un entorno real, aunque no se ha alcanzado el análisis en todas las condiciones climáticas deseadas por la falta de imágenes. Se profundizó en el conocimiento de Python desarrollando este proyecto, utilizando este lenguaje en todo el proceso. Los datos para el entrenamiento y la evaluación del modelo se adquirieron, procesaron y distribuyeron correctamente. El modelo de detección de embarcaciones, desarrollado con técnicas de aprendizaje profundo en Python, ha logrado mejoras significativas en la precisión y la capacidad de detección, aunque aún no alcanza un 100 % de detección. Sin embargo, no se ha logrado integrar el modelo en un sistema completo de detección de objetos dentro de la empresa. En cuanto a la documentación, el proceso de desarrollo, implementación y evaluación se está documentando adecuadamente, proporcionando una referencia útil para investigadores y profesionales interesados en el campo de la detección de objetos en entornos marítimos.

1.4. Competencias específicas y aportaciones del trabajo

En cuanto a los alcances de las competencias específicas⁴ y las aportaciones del trabajo realizado, este proyecto beneficia el ámbito socio-económico, técnico y científico.

1.4.1. Competencias

- **CI6:** Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.
- **CI16:** Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería del software.

Teniendo en cuenta el desarrollo del proyecto, se han analizado otras competencias.

- **G8:** Conocimiento de las materias básicas y tecnologías que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.
- **CI15:** Conocimiento y aplicación de los principios fundamentales y técnicas básicas de los sistemas inteligentes y su aplicación práctica.

⁴<https://www.eii.ulpgc.es/sites/default/files/2022-05/Grado%20en%20Ingenier%C3%ADa%20Inform%C3%A1tica%20-%20memoria%20del%20plan%20de%20estudios%20-%202019.pdf>

1.4.2. Aportaciones

El proyecto mencionado presenta varias contribuciones:

- **Ámbito socio-económico:** Vigilar zonas marítimas de acceso restringido, como áreas protegidas y de zonas exclusivas como PLOCAN. Por ello, es posible garantizar la seguridad y detener accesos no autorizados. Asimismo, evaluar ecosistemas marinos y rastrear el tránsito de barcos en los puertos. Ayudando así a una gestión sostenible, segura y eficaz de los recursos.
- **Ámbito técnico:** El avance en la integración de las secciones de Redes Generativas Adversativas (GAN) con la Inteligencia Artificial (IA) permite crear modelos y conjuntos de datos mucho más complejos, abriendo la posibilidad de nuevas líneas de investigación en áreas anteriormente inexploradas.
- **Ámbito científico:** Este proyecto da lugar a un avance significativo ya que es un tema poco estudiado hasta ahora, y establece las primeras semillas de trabajo para investigaciones futuras en estas disciplinas de control de seguridad marítima y conservación de medio ambiente.

En resumen, la identificación de embarcaciones mediante tecnologías de vanguardia es esencial para los diferentes sectores y forma parte de la protección y conservación responsable del medio ambiente marino.

1.5. Desafíos

El desarrollo de un sistema de detección de embarcaciones utilizando cámaras duales térmicas presenta varios desafíos técnicos y prácticos. A continuación se detallan los principales desafíos abordados en este proyecto y las razones para centrarse en la parte térmica de las imágenes capturadas.

- **Limitaciones de Visibilidad en el Entorno Marítimo:** El entorno marítimo está sujeto a condiciones variables de visibilidad debido a factores como la niebla, la lluvia y la oscuridad nocturna. Las cámaras convencionales RGB son significativamente menos efectivas bajo estas condiciones, lo que puede resultar en una detección inadecuada de embarcaciones.
- **Acceso a Datos Térmicos Brutos:** Un obstáculo considerable es la falta de acceso a los datos térmicos brutos de la cámara dual-térmica. En lugar de trabajar directamente con estos datos, se dispone de imágenes que representan las capturas térmicas, lo cual introduce una capa adicional de procesamiento y puede complicar el análisis.
- **Procesamiento de Imágenes Térmicas:** El procesamiento de imágenes térmicas requiere técnicas especializadas para interpretar las variaciones de calor. Estas técnicas deben ser precisas para diferenciar correctamente entre objetos relevantes, como embarcaciones, y el fondo marítimo.

- **Validación y Pruebas en Condiciones Reales:** Las condiciones en el entorno marítimo pueden ser muy cambiantes y complicadas para verificar un sistema de detección. Es necesario llevar a cabo pruebas en distintas condiciones de clima y luz para garantizar la resistencia y confiabilidad del sistema en todo momento. Al ser solo el mirador el entorno disponible, se hace difícil captar todas las situaciones adversas necesarias para validar.
- **Accesibilidad de la Tecnología y Costo:** Las cámaras térmicas o dual-térmicas y otros equipos avanzados pueden ser costosos, lo que limita la accesibilidad de esta tecnología a organizaciones con recursos limitados.

Después de establecer el contexto y los objetivos del proyecto en la introducción, es esencial definir el enfoque que se empleará para llevar a cabo el desarrollo del sistema. En la siguiente sección, se presenta la metodología adoptada en el transcurso del proyecto.

Capítulo 2

Estado actual

El avance de la tecnología en la última década ha transformado significativamente el campo de la visión por computador, especialmente en la detección de objetos. Este progreso ha sido impulsado por el desarrollo de redes neuronales profundas y técnicas avanzadas de aprendizaje automático que han permitido alcanzar niveles de precisión y eficiencia antes inimaginables. En el contexto de la seguridad marítima, estas innovaciones son cruciales para mejorar la detección y el seguimiento de embarcaciones, especialmente en condiciones de baja visibilidad y en entornos adversos.

La última tecnología en este campo cubre una amplia gama de técnicas, que incluyen desde métodos tradicionales basados en la visión por computador hasta innovadoras aplicaciones de aprendizaje automático y redes neuronales. Las cámaras RGB (*Red, Green, Blue*) han sido ampliamente utilizadas por su capacidad para capturar imágenes detalladas en condiciones de buena iluminación. Sin embargo, su eficacia disminuye drásticamente en entornos de baja luz, como durante la noche o en presencia de niebla densa.

Para superar estas limitaciones, las cámaras térmicas han emergido como una solución prometedora. Estas cámaras son capaces de detectar la radiación infrarroja emitida por los objetos, lo que permite la identificación de embarcaciones independientemente de las condiciones de iluminación. La tecnología térmica ha demostrado ser particularmente efectiva en entornos marítimos, donde las fluctuaciones de visibilidad son comunes.

Además de la tecnología de cámaras, la aplicación de técnicas de aprendizaje profundo ha revolucionado el campo de la detección de objetos. Las redes neuronales convolucionales (CNN) han mostrado un rendimiento superior en la identificación y clasificación de embarcaciones en imágenes, tanto RGB como térmicas. Estos modelos pueden ser entrenados con grandes volúmenes de datos para reconocer patrones complejos y mejorar continuamente su precisión.

2.1. Fundamentos de una Red Neuronal

2.1.1. Red Neuronal

Una red neuronal (NN), como se muestra en la Ilustración 2.1, imita el funcionamiento del cerebro humano, compuesta por muchas neuronas interconectadas que producen secuencias de activaciones. Las neuronas de entrada perciben el entorno mediante sensores, mientras que otras se activan a través de conexiones ponderadas. El aprendizaje consiste en ajustar los pesos para que la NN realice tareas deseadas, como conducir un coche de forma autónoma o análisis de imágenes médicas. Este proceso puede requerir varias etapas computacionales, transformando las activaciones de manera no lineal. [41]

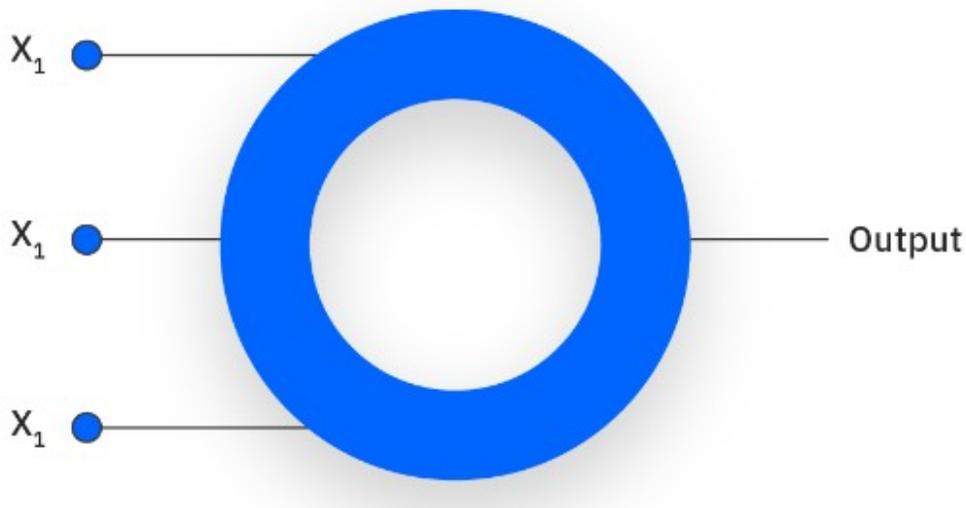


Ilustración 2.1: Estructura básica de una red neuronal. Imagen obtenida de [22]

2.1.2. GAN (Generative Adversarial Network)

Las Redes Generativas Adversariales (GANs) son una arquitectura de aprendizaje profundo diseñada para generar datos nuevos que imitan un conjunto de datos de entrenamiento. Una GAN se conforma de dos redes neuronales: el Generador y el Discriminador. El Generador modifica ciertos atributos para generar datos falsos, mientras que el Discriminador verifica la autenticidad de los datos, determinando si son reales o falsos. Mientras compiten entre sí, ambos modelos mejoran su rendimiento hasta que el Generador produzca datos realistas. Dentro del marco del proyecto, las GAN pueden generar imágenes térmicas realistas para mejorar la detección de objetos de los modelos, mejorando así su precisión y robustez. [15]

Dado que los datos térmicos en bruto de la cámara del mirador y del dataset utilizado en el modelo no están disponibles, no fue posible utilizar ThermalGAN. ThermalGAN ha sido

específicamente desarrollado para generar y procesar imágenes térmicas, lo cual habría sido extremadamente interesante.

Debido a la falta de los datos térmicos requeridos, se optó por el uso de CycleGAN. CycleGAN está diseñado para aprender a convertir una imagen de un dominio a otro sin la necesidad de tener pares de imágenes correspondientes, conocido como *unpaired data* (ver Ilustración 2.2). Esto significa que CycleGAN puede aprender a mapear imágenes de fotos RGB a una representación térmica sin requerir conjuntos de datos donde cada imagen RGB tenga una imagen térmica correspondiente.

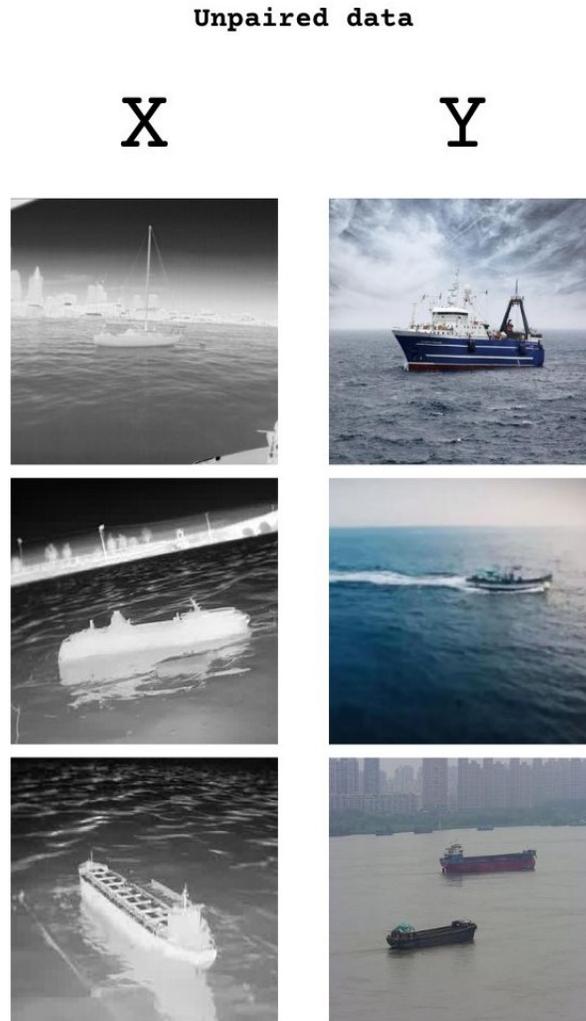


Ilustración 2.2: Conjunto de datos no emparejados (*Unpaired data*) utilizados en CycleGAN. La columna **X** contiene imágenes de barcos de una cámara térmica, mientras que la columna **Y** contiene imágenes de barcos de una cámara óptica.

La arquitectura de CycleGAN incluye dos generadores y dos discriminadores:

- Un generador **G** transforma imágenes del dominio **X** (fotos RGB) al dominio **Y** (re-

presentación térmica).

- El otro generador \mathbf{F} realiza la conversión inversa, transformando imágenes del dominio \mathbf{Y} al dominio \mathbf{X} .
- Los discriminadores \mathbf{D}_Y y \mathbf{D}_X se encargan de distinguir entre imágenes reales del dominio \mathbf{Y} y las generadas por \mathbf{G} , y entre imágenes reales del dominio \mathbf{X} y las generadas por \mathbf{F} , respectivamente.

Una característica clave de CycleGAN es la *Cycle Consistency Loss*, como se muestra en la Ilustración 2.3. Esta pérdida asegura que si una imagen x del dominio \mathbf{X} se transforma al dominio \mathbf{Y} mediante \mathbf{G} y luego se convierte de nuevo al dominio \mathbf{X} mediante \mathbf{F} , la imagen resultante $\mathbf{F}(\mathbf{G}(x))$ debe ser similar a la imagen original x . Lo mismo aplica en la dirección inversa: una imagen y del dominio \mathbf{Y} transformada al dominio \mathbf{X} mediante \mathbf{F} y luego vuelta a convertir al dominio \mathbf{Y} mediante \mathbf{G} debe dar como resultado una imagen $\mathbf{G}(\mathbf{F}(y))$ similar a y . Esta consistencia cíclica es fundamental para garantizar que las características esenciales de las imágenes se mantengan a través de las transformaciones.

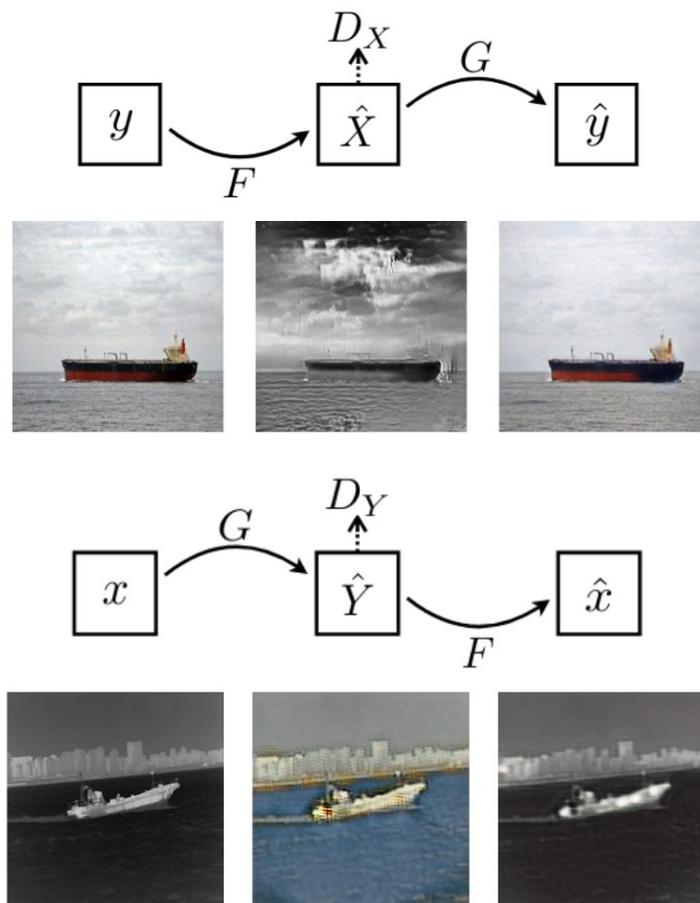


Ilustración 2.3: Pérdida de consistencia cíclica (*Cycle Consistency Loss*) utilizada en CycleGAN. La imagen superior muestra la conversión de una imagen de RGB a térmica, mientras que la imagen inferior muestra la conversión de una imagen térmica a RGB

2.1.3. Traducción de imagen a imagen no emparejada

La traducción de imagen a imagen no emparejada es una técnica que transforma imágenes de un dominio a otro sin datos emparejados. Utiliza una pérdida adversarial para hacer que la distribución de las imágenes transformadas sea similar a la del dominio objetivo. Además, introduce una correspondencia inversa y una pérdida de ciclo para garantizar la consistencia. Este método ha demostrado ser superior en tareas como la transferencia de estilo y la mejora de fotos, superando a otros enfoques que dependen de funciones de similitud predefinidas. [52] [6]

2.2. Tareas de Visión por Computador

2.2.1. Clasificación

La clasificación de imágenes implica asignar a una imagen una etiqueta específica para representar su contenido. Los modelos procesan una imagen como entrada y generan una predicción sobre la clase la que es más probable que pertenezca la imagen. [47]

2.2.2. Localización

La localización de objetos en visión por computadora se refiere al proceso de identificación y ubicación precisa de una única instancia de objeto dentro de una imagen. Este proceso genera un cuadro delimitador (*bounding box*) que rodea al objeto de interés. [40]

2.2.3. Detección

La detección de objetos en el campo de visión por computadora consiste en identificar y localizar objetos de interés presentes en una imagen. Este proceso no solo implica clasificar dichas entidades predefinidas, como por ejemplo la categoría *boat*, sino también en ubicar los objetos en la imagen. [53]



Ilustración 2.4: Tareas de Visión por Computador

2.2.4. Composición de imágenes

La composición de imágenes es una técnica utilizada en diversos campos como el comercio electrónico, el entretenimiento y la ampliación de datos. Este proceso puede clasificarse en dos tipos principales: guiada por texto y guiada por imágenes.

Composición Guiada por Texto [29]

En este enfoque, la composición de múltiples objetos se especifica únicamente a través de un mensaje de texto, sin limitar la apariencia de los objetos siempre y cuando su semántica se alinee con el mensaje. Aunque los modelos condicionados por texto han tenido grandes éxitos, a menudo son propensos a errores semánticos, especialmente cuando el mensaje de texto incluye múltiples objetos. Estos errores pueden incluir:

- **Fuga de atributos:** Características de un objeto que se aplican incorrectamente a otro.
- **Intercambio de atributos:** Atributos que se asignan a los objetos incorrectos.
- **Objetos faltantes:** Objetos mencionados en el texto que no aparecen en la imagen generada.

La Ilustración 2.5 muestra un representación de este proceso de Composición guiada por Texto.

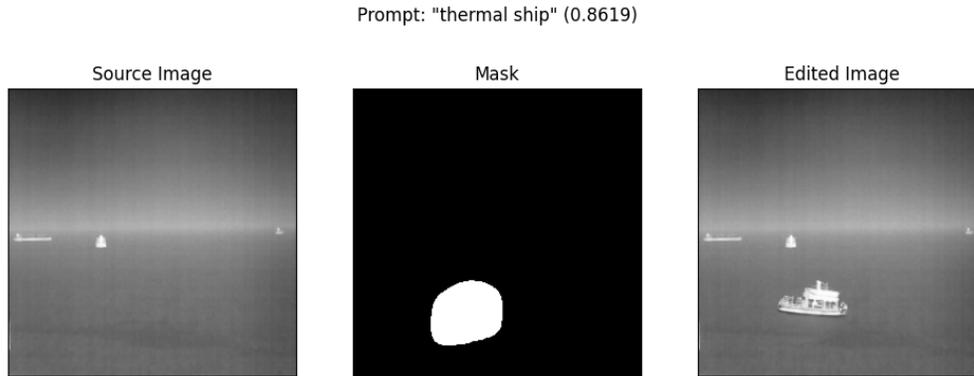


Ilustración 2.5: Proceso de aplicación de Composición guiada por Texto. Primero, se muestra la imagen de fondo, seguida de la imagen de la máscara que se aplicará. Finalmente, se presenta el resultado final obtenido al combinar la imagen de fondo y la máscara mediante la técnica de *Blended Diffusion*. Imagen generada con [4]

Composición Guiada por Imágenes

En contraste, la composición guiada por imágenes, como se muestra en la Ilustración 2.6, implica la incorporación de objetos y escenarios específicos a partir de fotos proporcionadas por el usuario, a menudo con la ayuda de un mensaje de texto. Sin embargo, la inclusión de imágenes reales adicionales presenta un desafío mayor, especialmente cuando se fusionan imágenes de diferentes dominios visuales, lo cual es el enfoque principal de la armonización de imágenes pictóricas.



Ilustración 2.6: Ejemplo de composición guiada por texto. De izquierda a derecha: fondo, barco, máscara del barco y resultado de la Composición Guiada por Imágenes. Imagen generada por [36]

Subtareas en la Composición Guiada por Imágenes La composición guiada por imágenes se divide convencionalmente en varias subtareas, cada una abordada por diferentes modelos y *pipelines*:

- **Colocación de objetos:** Determinar la posición adecuada de los objetos dentro de la imagen.

- **Mezcla de imágenes:** Fusionar objetos y fondos de manera cohesiva.
- **Armonización de imágenes:** Ajustar los elementos para que parezcan parte de una escena coherente.
- **Generación de sombras:** Crear sombras realistas que coincidan con la iluminación de la escena.

Cada una de estas subtarefas es crucial para lograr una composición visualmente equilibrada y realista.

2.3. Transfer Learning

Transfer Learning[37] es una técnica de transferencia de conocimiento que se realiza con un modelo que se ha entrenado previamente para una tarea específica. Dado que el entrenamiento se hizo previamente, significa más datos y un entrenamiento más rápido. El modelo preentrenado se puede usar sin modificaciones o ajustándose al nuevo problema específico.

2.4. Modelos de Detección de Objetos

2.4.1. YOLO (You Only Look Once)

YOLO, que significa You Only Look Once, fue introducido por Joseph Redmon en 2015. Es un algoritmo de aprendizaje profundo que utiliza redes neuronales convolucionales (CNN) para identificar objetos en imágenes en tiempo real, aunque a costa de menos precisión. La Ilustración 2.7, muestra su funcionamiento: la imagen se divide en una cuadrícula, dentro de la cual se predicen posibles *bounding boxes* y se asigna un nivel de certeza. Las cajas con probabilidades más bajas se eliminan. Finalmente, se aplica un proceso de *non-max suppression* para eliminar duplicados. YOLO es conocido por su velocidad y tiene una precisión ligeramente inferior a Faster R-CNN, por lo que es adecuado para aplicaciones donde se requiere una detección rápida y precisa de objetos en movimiento [38].

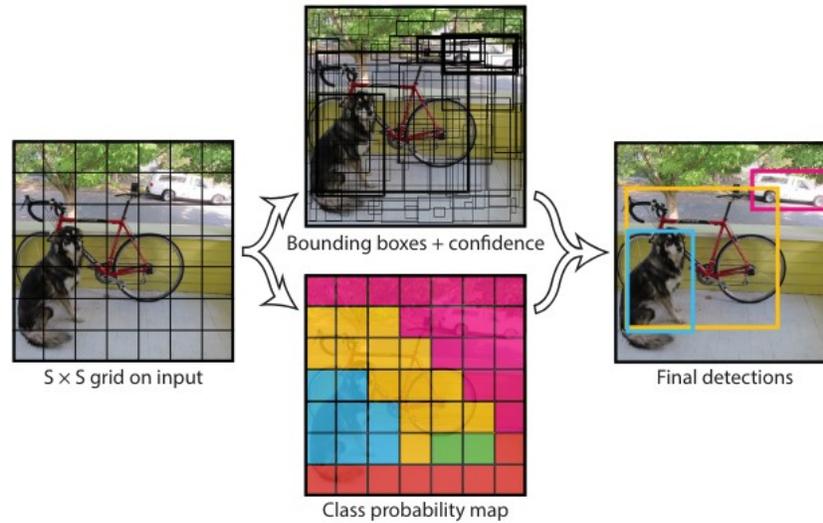


Ilustración 2.7: Funcionamiento de YOLO. Imagen obtenida de [38]

Con el avance de la tecnología en visión por computador, los modelos YOLO han revolucionado la detección de objetos al ofrecer un equilibrio único entre precisión y velocidad. Desde su versión inicial hasta las iteraciones más recientes como YOLOv9, estas redes neuronales han impulsado aplicaciones en tiempo real en una variedad de campos, desde la seguridad hasta la conducción autónoma. Cada nueva versión ha introducido mejoras significativas, optimizando tanto la eficiencia como la capacidad de adaptación a condiciones adversas, garantizando que continúen siendo líderes en el campo de la visión por computador.

YOLOv7

Introdujo cambios en la arquitectura que optimizaron tanto el backbone como la cabeza del modelo, resultando en una mejora notable en la precisión y velocidad en comparación con YOLOv6. Este modelo es capaz de mantener capacidades de procesamiento en tiempo real. Las innovaciones en YOLOv7 permitieron un mayor mAP (media de precisión promedio), crucial para aplicaciones que requieren detección rápida y precisa. [42] [45]

YOLOv8

YOLOv8 continuó construyendo sobre las mejoras de YOLOv7, incorporando técnicas avanzadas como las redes piramidales de características (FPN) y las redes de agregación de caminos (PAN) para mejorar la extracción de características a diferentes escalas. Estos avances permitieron una mayor precisión y velocidad de inferencia. YOLOv8 también mejoró el manejo de objetos pequeños y escenas abarrotadas. [42]

YOLOv9

Representa la última innovación en esta serie. Introduce un enfoque novedoso para la información de gradiente llamado *información de gradiente programable*, que permite un control más detallado sobre el proceso de aprendizaje. Esta versión también integra técnicas avanzadas como convoluciones profundas y mecanismos de atención, lo que permite que

el modelo capture mejor patrones complejos en los datos. El marco de entrenamiento de YOLOv9 incluye un conjunto integral de hiperparámetros y estrategias de aumento diseñados para maximizar el rendimiento durante períodos de entrenamiento prolongados. [46]

La Ilustración a continuación 2.8, muestra una gráfica con la comparación del rendimiento de algunos de los modelos de YOLO, en términos de Precisión Promedio (AP , *Average Precision*) y Fotogramas por Segundo (FPS) utilizando el conjunto de datos *MS COCO*.

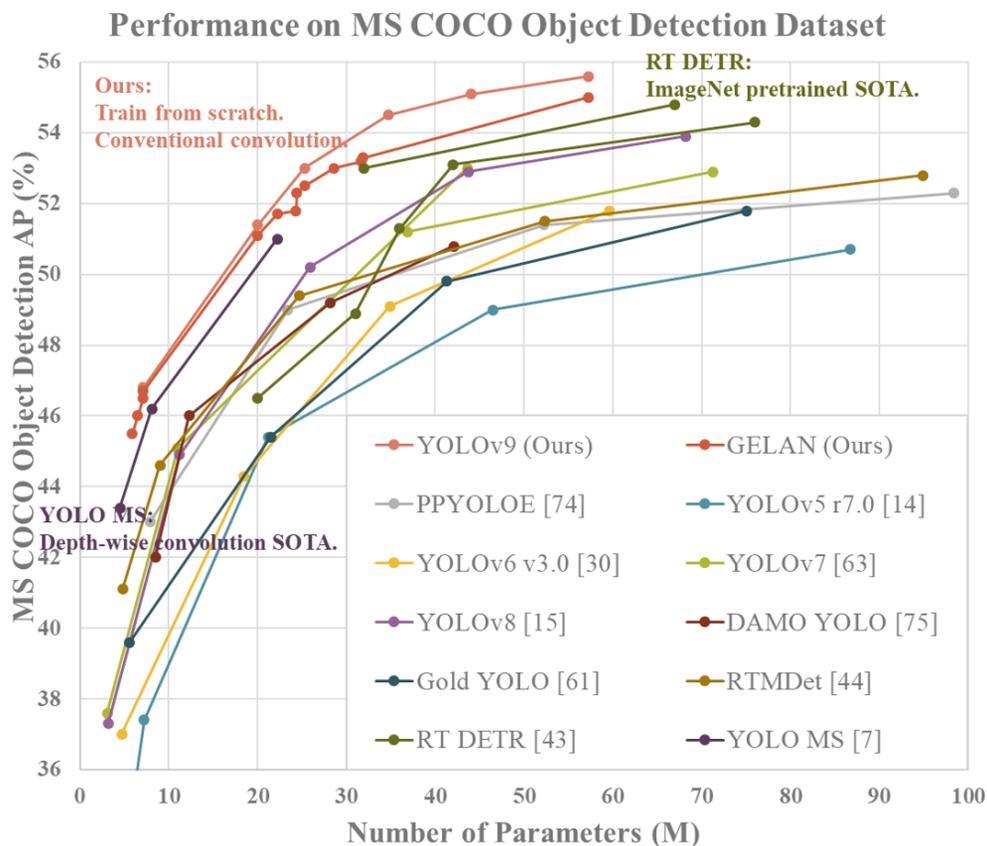


Ilustración 2.8: Rendimiento en el conjunto de datos de detección de objetos de *MS COCO*. Imagen obtenida de la publicación original de YOLOv9 [46]

2.4.2. Faster-RCNN (Region-based Convolutional Neural Networks)

Faster R-CNN es un modelo de detección de objetos que hace uso de una Red de Propuesta de Región (RPN) como se muestra en la Ilustración 2.9. La RPN tiene similitudes convolucionales con la red de detección, lo que facilita la generación de propuestas regionales a bajo costo. Predice simultáneamente los límites de los objetos y las puntuaciones de objetividad en cada posición. El sistema de detección de objetos consta de dos módulos: una red profunda totalmente convolucional que propone regiones y el detector Faster R-CNN que las emplea. Ambos módulos se combinan en una red unificada, en la que la RPN guía al detector en la búsqueda. Aunque es más lento que YOLO, es más preciso, lo que lo hace adecuado

para aplicaciones donde la precisión es prioritaria sobre la velocidad. [39]

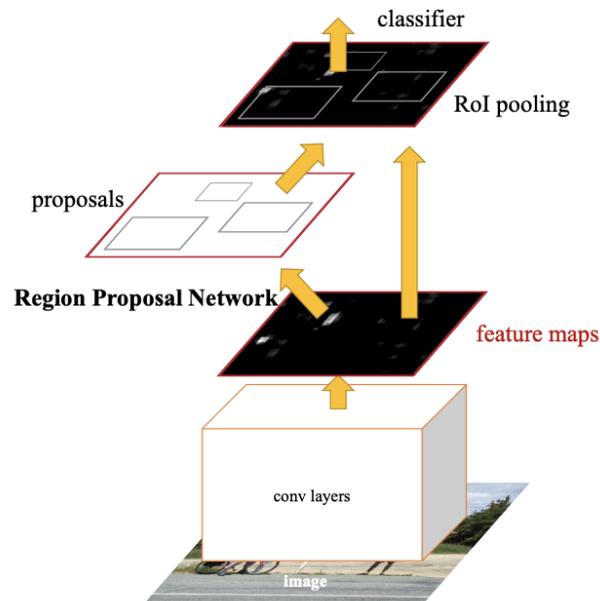


Ilustración 2.9: Estructura de Faster R-CNN. Imagen obtenida de [39]

2.4.3. CenterNet

CenterNet es un modelo de detección de objetos que utiliza la estimación de puntos clave para identificar puntos centrales y determinar diversas propiedades del objeto, como tamaño, ubicación en 3D, orientación y pose. A diferencia de los detectores tradicionales basados en cuadros delimitadores, CenterNet es diferenciable de extremo a extremo, lo que lo hace más simple, rápido y preciso. Al ser simple y eficiente, es ideal para aplicaciones con recursos computacionales limitados [11]. La siguiente Ilustración 2.10 muestra su arquitectura.

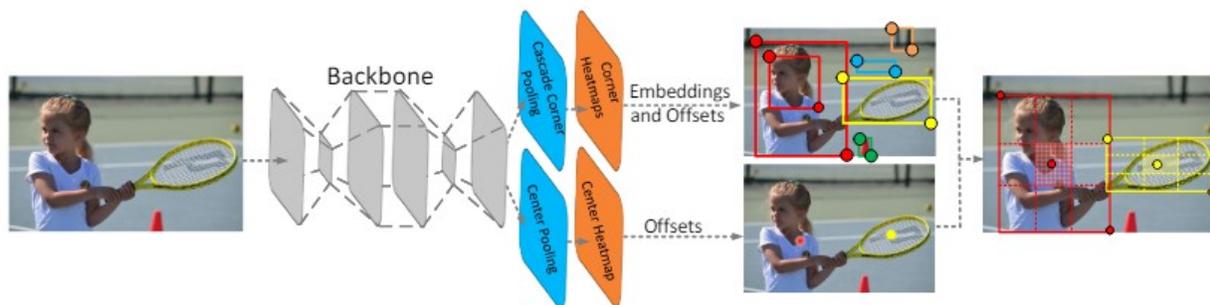


Ilustración 2.10: Arquitectura de CenterNet. Imagen obtenida de [11]

2.4.4. SSD

SSD (*Single Shot MultiBox Detector*) está diseñado para la detección de objetos en tiempo real, eliminando la necesidad de una red de propuesta regional utilizada por Faster R-CNN, que, aunque precisa, funciona a solo 7 fotogramas por segundo. SSD discretiza el espacio de salida en cuadros predeterminados de varias proporciones y escalas, generando puntuaciones y ajustando estos cuadros durante la predicción para mejorar la coincidencia con los objetos. Gracias a características de escala múltiple y cuadros predeterminados, SSD alcanza y a veces supera la precisión de Faster R-CNN, permitiendo un procesamiento en tiempo real [50] [27]. La arquitectura SSD se muestra en la Ilustración 2.11.

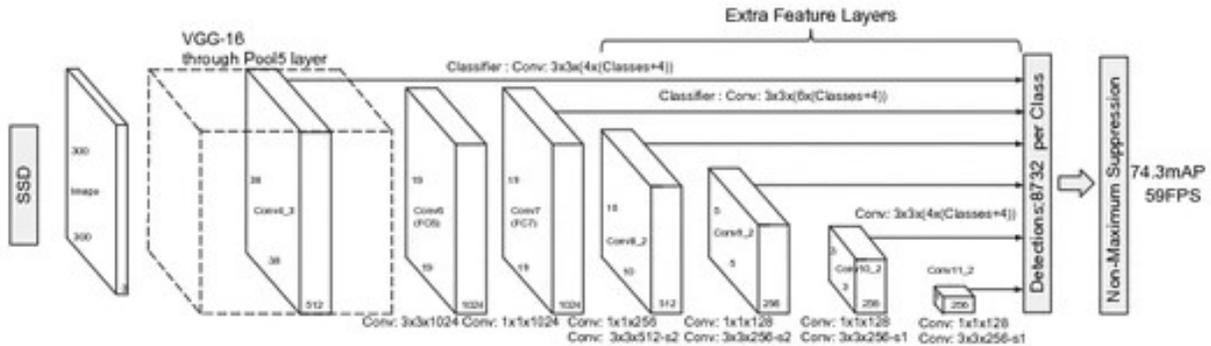


Ilustración 2.11: Arquitectura de SSD. Imagen obtenida de [27]

2.5. Conjunto de datos

Para el entrenamiento y evaluación de este proyecto, se han recopilado datos de diversos entornos marítimos, incluyendo mar abierto, puertos marítimos y canales, así como de diferentes tipos de barcos, como veleros y barcos de carga. Además, se han considerado situaciones ambientales adversas, como la calima, manchas en la lente y desenfoque, para asegurar que el modelo sea generalizado y robusto. Se han añadido imágenes del entorno real del mirador, específicamente para el caso de la plataforma. En el conjunto de datos, las imágenes utilizadas abarcan desde 2021 hasta la actualidad.

A continuación, en la Ilustración 2.12, se muestran algunas imágenes representativas del conjunto de datos utilizado para entrenar el modelo. Estas imágenes incluyen diversos tipos de barcos, mostrando la variedad y complejidad de los escenarios considerados.

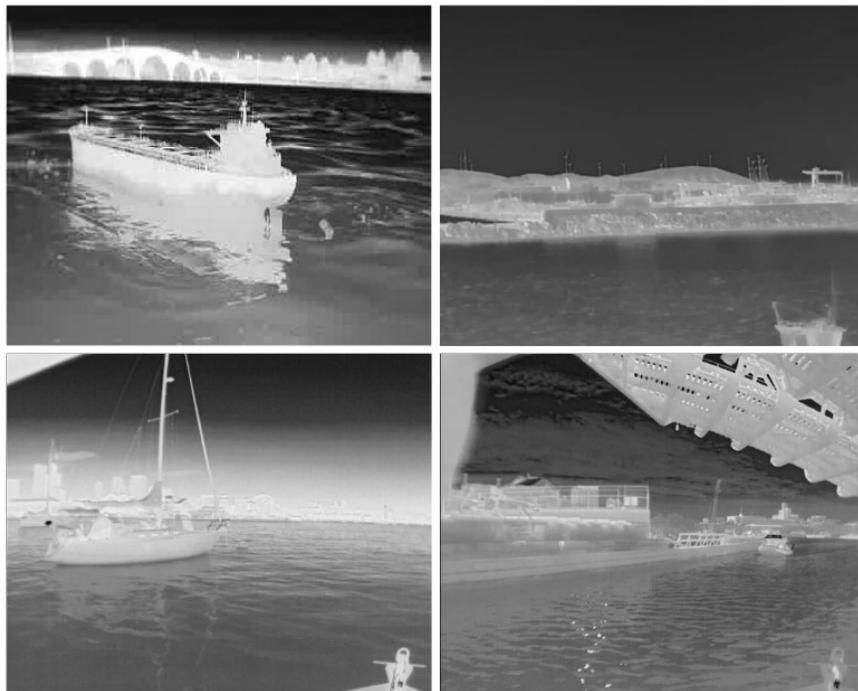


Ilustración 2.12: Ejemplos de imágenes de barcos contenidas en el Dataset generado

Debido a la escasez de imágenes térmicas de acceso libre, ya que la mayoría de datasets disponibles son repetitivos (véase Ilustración 2.13) o contenían la misma imagen una y otra vez o incluían datos en blanco y negro (véase Ilustración 2.13), se han tomado medidas para aumentar estos conjuntos a lo largo del desarrollo del proyecto. La falta de datos suficientes y variados representan una dificultad significativa para generar buenos resultados, ya que los modelos de detección y análisis requieren una diversidad adecuada de ejemplos para entrenarse efectivamente.



Ilustración 2.13: Ejemplos de imágenes descartadas del conjunto de datos. Las dos imágenes de la izquierda fueron descartadas por ser muy similares, mostrando el mismo barco y fondo, lo que no aporta valor adicional al entrenamiento. La imagen de la derecha fue descartada por ser en blanco y negro en lugar de térmica. Imágenes obtenidas de [4].

2.6. Métricas de Evaluación de los modelos

Para determinar si una detección es buena o no, tenemos los siguientes conceptos:

- **Verdadero positivo:** Se etiqueta correctamente una imagen, por ejemplo, categorizar una imagen de un barco como barco.
- **Verdadero negativo:** No se etiqueta correctamente una imagen, por ejemplo, no categorizar una imagen de barco como plataforma.
- **Falso positivo:** Se etiqueta una imagen cuando no debería, por ejemplo, la categorización de una imagen de un barco como plataforma.
- **Falso negativo:** No se etiqueta una imagen, pero debería, por ejemplo, no categorizar una imagen de barco como barco.
- **Precisión:** Mide el porcentaje de imágenes detectadas correctamente entre todas las predicciones positivas. Se calcula:

$$\frac{TP}{TP + FP}$$

- **Recall:** Mide el porcentaje de imágenes etiquetadas correctamente entre todas las predicciones reales. Se calcula:

$$\frac{TP}{TP + FN}$$

La métrica empleada no es exclusiva de YOLO, sino que se utiliza en distintos modelos de detección de objetos.

- **Precisión Media Promedio (mAP):** Evalúa la precisión promedio de todas las clases y ofrece un único valor para el modelo. La precisión evalúa la exactitud del modelo al detectar correctamente los objetos. [44] [20] [16]
- **Intersección sobre Unión (IoU):** Mide la precisión del tamaño y ubicación entre el cuadro delimitador real y el predicho. [44] [20]

Además de los valores numéricos, la evaluación proporciona salidas visuales que pueden ayudar a mejorar la comprensión del rendimiento del modelo. A continuación, se explicará cada una de ellas:

- **Matriz de confusión:** Muestra un recuento de los verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos, donde cada columna de la matriz representa la clase asignada por el modelo y cada fila la instancia real.
- **Curva de precisión:** Representa el porcentaje de verdaderos positivos con diferentes umbrales. Se calcula:

$$\frac{TP + TN}{\text{todos los casos}}$$

- **Curva de recuperación:** Muestra la frecuencia de los verdaderos positivos según los distintos umbrales. Se calcula:

$$\frac{TP}{TP + FN}$$

- **Curva de precisión-recuperación:** Visualiza el problema de clasificación, representa la precisión frente a la recuperación con distintos umbrales. Al analizarla, se puede decidir si se quiere un modelo preciso pero con menos resultados, o con más resultados y más falsos positivos.
- **Curva de puntuación F1:** Muestra el equilibrio del modelo entre falsos positivos y falsos negativos en distintos umbrales.

2.7. Captura de imágenes

2.7.1. Cámara térmica

Una cámara térmica, también conocida como cámara infrarroja, es un dispositivo sin contacto que captura la radiación infrarroja emitida por los objetos y la convierte en una imagen visual, permitiendo visualizar diferentes temperaturas. A diferencia de las cámaras convencionales, que dependen de la luz visible para formar imágenes, las cámaras térmicas capturan imágenes basadas en el calor emitido por los objetos.

Estos dispositivos están compuestos por una lente, un sensor térmico, un sistema de procesamiento electrónico y una carcasa mecánica, como se muestra en la Ilustración 2.14. La lente enfoca la energía infrarroja en el sensor, que captura las diferencias de temperatura de los objetos y las convierte en una imagen digital o analógica. Aunque la resolución de estas cámaras es menor que la de las cámaras de luz visible debido a la necesidad de detectar longitudes de onda más grandes, son altamente efectivas en entornos con visibilidad limitada, como en la oscuridad total o en condiciones de humo.

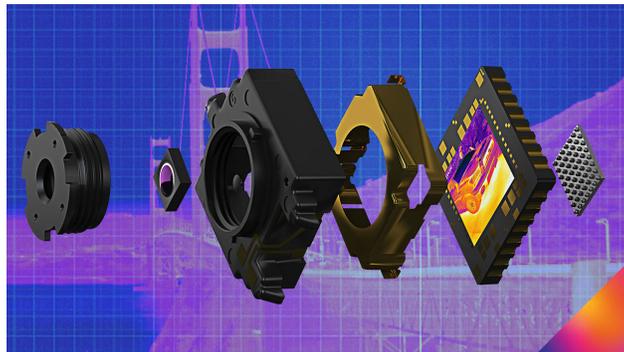


Ilustración 2.14: Estructura interna de una cámara térmica. Imagen obtenida de [13]

Además, las cámaras térmicas detectan la temperatura de los objetos al capturar diferentes niveles de luz infrarroja, la cual es invisible al ojo humano. Cuanto más caliente está un

objeto, más radiación infrarroja emite. Así, esto les permite a las cámaras térmicas funcionar en completa oscuridad o en ambientes llenos de humo.

Originalmente desarrolladas para la vigilancia y operaciones militares, las cámaras térmicas se utilizan ahora en una variedad de aplicaciones, incluyendo vigilancia, rescate y monitoreo industrial. Su capacidad para ver en función de la temperatura en lugar de la luz visible las convierte en herramientas valiosas en muchos campos industriales, de seguridad y científicos. [33] [24]

2.7.2. Cámara óptica

Las cámaras ópticas capturan imágenes utilizando luz visible y son las más comunes. Utilizan sensores como *Charge-Coupled Device* (CCD) o *Complementary Metal-Oxide-Semiconductor* (CMOS) para convertir la luz en señales eléctricas, que luego se procesan para crear imágenes digitales. Las cámaras ópticas se utilizan en fotografía, videografía, vigilancia y análisis científico debido a su alta resolución y capacidad para capturar detalles finos en diversas condiciones de iluminación. [51]

En comparación con las cámaras térmicas, las cámaras ópticas dependen de condiciones de iluminación adecuadas y no pueden operar en completa oscuridad sin fuentes de luz adicionales. Sin embargo, proporcionan imágenes más detalladas en términos de color y textura bajo buenas condiciones de iluminación.

2.7.3. Radar

El radar utiliza ondas de radio para detectar y localizar objetos. Funciona emitiendo pulsos de radiofrecuencia que rebotan en los objetos y regresan al receptor. Esto permite determinar la distancia, velocidad y características de los objetos. Los sistemas de radar son esenciales en navegación, meteorología, defensa y vehículos autónomos debido a su capacidad para operar en diversas condiciones climáticas y detectar objetos a largas distancias. En comparación con las cámaras térmicas, los radares no pueden proporcionar imágenes detalladas.

Con una clara comprensión del estado del arte y una metodología sólida y planificación, el siguiente paso es llevar a cabo el desarrollo del sistema.

Capítulo 3

Metodología

En este capítulo se presentará la metodología seguida para desarrollar el sistema, desde la obtención de imágenes hasta el entrenamiento y validación del modelo de detección.

3.0.1. Metodología Scrum

Scrum es la metodología ágil más ampliamente adoptada en el desarrollo de software, centrada en la entrega iterativa y rápida de productos de alta calidad. Cada proyecto se divide en sprints, periodos de tiempo fijos de aproximadamente dos semanas. Cada sprint comienza con una reunión de planificación donde el equipo, compuesto por equipos pequeños, auto-organizados y multifuncionales, define el conjunto de tareas a abordar, teniendo la flexibilidad de realizar ajustes según las necesidades emergentes.

Durante el sprint, estos equipos trabajan de manera iterativa e incremental en reuniones diarias de Scrum, para sincronizar esfuerzos y resolver impedimentos. Al final de cada sprint, se realiza una reunión de revisión donde se presenta el incremento del producto desarrollado y una retrospectiva para identificar mejoras en el proceso. Scrum facilita un enfoque que permite a estos equipos adaptarse a cambios en los requisitos y alcanzar objetivos de manera eficiente y adaptativa. [32]

3.0.2. Metodología Kanban

Kanban, que significa *tablero* en japonés, es una metodología ágil que visualiza el flujo de trabajo de un proyecto mediante el uso de tableros que muestran las tareas en diferentes etapas del proceso, como *To-Do*, *Doing* y *Done*. Esta metodología ayuda a mejorar la transparencia y la gestión del flujo de trabajo al hacer que el progreso y las tareas pendientes sean visibles para todos los miembros del equipo.

Además, Kanban limita el trabajo en curso para identificar los cuellos de botella y mejorar la eficiencia. Al fomentar un enfoque en la finalización de las tareas antes de comenzar nuevas,

Kanban optimiza la capacidad del equipo para entregar trabajo de manera constante y fluida. [1] [8] [19]

3.0.3. Metodología Scrumban

ScrumBan es una metodología ágil híbrida que combina elementos de los marcos de trabajo Scrum y Kanban. Su objetivo es aprovechar las fortalezas de ambos enfoques para mejorar la productividad, la flexibilidad y la eficiencia en el desarrollo de software y otros procesos de gestión de proyectos. [2]

Para satisfacer las necesidades específicas de este proyecto, se adaptó la metodología Scrumban para gestionarlo de manera efectiva. El proyecto se dividió en varios sprints de diferentes duraciones, cada uno enfocado en tareas concretas como la generación de imágenes sintéticas. La planificación de cada sprint se llevó a cabo en reuniones con los tutores, donde se revisaron los resultados anteriores y se ajustó el enfoque según fuese necesario.

La priorización de tareas en cada sprint se basó en análisis previos y en la dirección dictada por los resultados obtenidos. En este caso, dado que el dataset no era extenso, se decidió priorizar la creación y optimización de este importante componente. Además, se establecieron límites de trabajo en curso (WIP) por tarea para mantener un flujo constante de trabajo y evitar la sobrecarga. Por ejemplo, se asignó un número específico de líneas de trabajo permitidas simultáneas para cada tarea.

Además de la planificación inicial, se permitió que el equipo iniciara nuevas tareas de forma autónoma dentro del marco de cada sprint según fuera necesario. Esto proporcionó flexibilidad para adaptarse a cambios en las prioridades y en el contexto del proyecto, permitiendo que el equipo tome decisiones proactivas sobre cómo avanzar en el trabajo en función de las necesidades del proyecto en curso. El capítulo de desarrollo 5 ofrece numerosos detalles de cada una de las etapas, divididas en Sprints, que se han realizado para llevar a buen término este TFG.

Capítulo 4

Recursos utilizados y Presupuesto

4.1. Recursos

Una vez planificadas las actividades y etapas del proyecto, es fundamental identificar y detallar los recursos específicos que serán utilizados.

4.1.1. Hardware

Dispositivo personal: HP Laptop 15s-eq1xxx

Cámara

La cámara *Thermal & Optical Bi-spectrum PTZ network* integra diversas funciones como decodificar, lente óptica y térmica, enfoque automático y zoom de alta definición. Permite detectar barcos y otros objetos en el mar, así como medir la temperatura del entorno marítimo en tiempo real y detectar incendios. Ofrece una interfaz con imágenes en vivo de alta calidad, siendo ideal para la seguridad y monitorización de barcos en mares y puertos. [7]

Cámara térmica

En la Tabla 4.1 se presentan las especificaciones técnicas del módulo térmico utilizado para la obtención de imágenes en la zona del mirador.

Tabla 4.1: Especificaciones técnicas del módulo térmico utilizado en el sistema de detección. Los valores fueron obtenidos de Hikvision [18]

Característica	Valor
Image sensor	Vanadium Oxide Uncooled Focal Plane Arrays
Pixel Interval	17 μm
Response Waveband	8 μm to 14 μm
NETD	<40mk(@25°C,F#=1.0)
MRAD	0.57 to 0.11 mrad
Field of View	WIDE:20.56° x 16.51°; TELE:4.15° x 3.32°
Min. Focusing Distance	2 m
Aperture	F1.2

Cámara óptica

Las características técnicas del módulo óptico se encuentran resumidas en la Tabla 4.2.

Tabla 4.2: Características técnicas del módulo óptico complementario al módulo térmico. Los valores fueron obtenidos de Hikvision [18]

Característica	Valor
Max. Resolution	1920 × 1080
Image Sensor	1/1.9" Progressive Scan CMOS
Min. Illumination	Color: 0.002 lux @ (F1.5 , AGC On), B/W: 0.0002 lux @ (F1.5 , AGC On)
Field of View	58.7° - 2.0° (Wide-Tele)
Aperture Range	F1.5-F4.5
Focus Mode	Auto / Semiautomatic / Manual
WDR	120dB
Shutter Speed	PAL: 1-1/30,000s / NTSC: 1-1/30,000s
Day	IR Cut Filter
Night	IR Cut Filter

4.1.2. Software

Sistema operativo: Microsoft Windows 10 Home

4.1.3. Librerías

Se utilizaron diferentes bibliotecas para cada etapa del proyecto, desde el manejo de imágenes hasta la interpretación y examinación de datos.

- **NumPy:** Biblioteca de Python especializada en operaciones en arrays y matrices multidimensionales. NumPy facilita el análisis y la manipulación de grandes volúmenes de

datos, y es la base sobre la cual se construyen muchas otras herramientas científicas en Python. [17]

- **OpenCV:** Biblioteca de código abierto que ofrece el procesamiento y análisis de imágenes y vídeos en tiempo real, así como la implementación de diversas técnicas de visión por computador. [3]
- **Pillow:** Librería de Python para el procesamiento de imágenes que se emplea para manipular y preprocesar imágenes. ¹
- **PyTorch:** Utilizado para la construcción de redes neuronales. Se usa para realizar tareas como visión por computador y procesamiento del lenguaje natural (NLP). [21]

4.1.4. Otras herramientas

- **Anaconda:** Sistema de gestión de entornos y paquetes. ²
- **Trello:** Aplicación web que permite coordinar proyectos en equipo mediante un tablero Kanban, en el cual se priorizan y organizan las tareas.
- **Slack:** Plataforma de mensajería diseñada para equipos, la cual permite reuniones en equipo o con clientes, además de gestionar los canales de comunicación.
- **CVAT:** CVAT (*Computer Vision Annotation Tool*) es una herramienta web que permite la anotación de imágenes y vídeos. ³

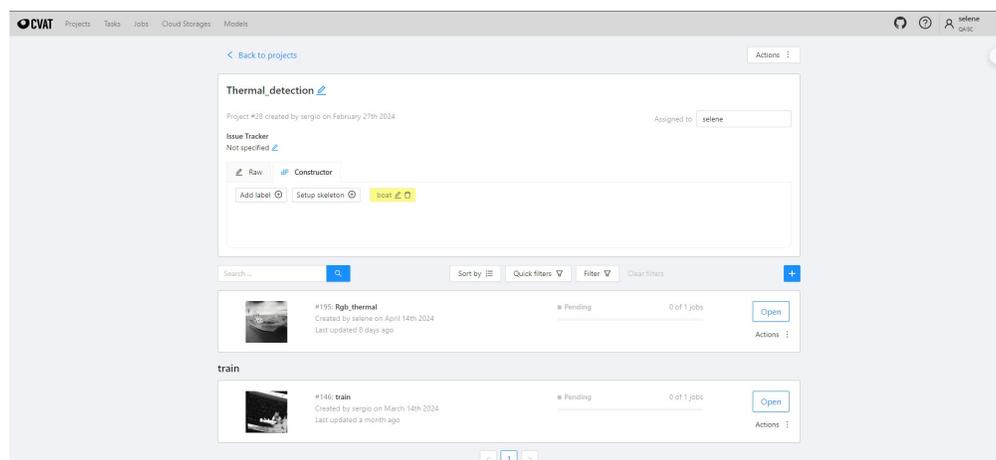


Ilustración 4.1: Pantalla de inicio del proyecto, con los distintos datasets a etiquetar en CVAT

¹<https://pypi.org/project/pillow/>

²<https://www.anaconda.com/download>

³<https://www.cvat.ai/>

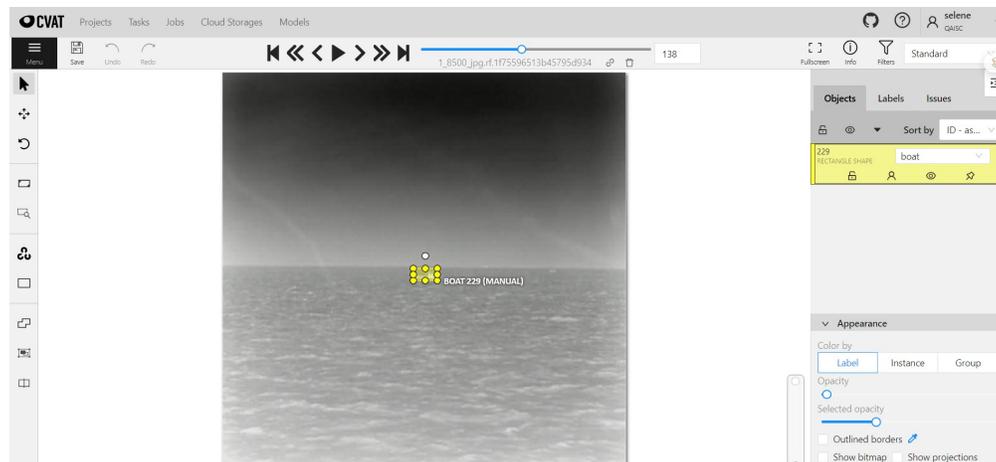


Ilustración 4.2: Ejemplo de imagen etiquetada utilizando la herramienta CVAT

La Ilustración 4.1 ilustra la pantalla de inicio del proyecto en CVAT, destacando los distintos datasets disponibles para la etiquetación del proyecto. Mientras que la Ilustración 4.2 presenta un ejemplo de imagen etiquetada con la herramienta CVAT para detección de barcos.

- **ONVIF:** Define protocolos de red para dispositivos de videovigilancia de interoperabilidad. Se uso para operaciones como CRUD de presets en cámaras y acciones de movimiento. ⁴
- **ISAPI (*Intelligent Security API*):** Protocolo RESTful basado en HTTP utilizado para la comunicación entre la cámara dual-térmica de Hikvision, NVR y sistemas cliente. ⁵

Tras identificar y detallar los recursos necesarios para el desarrollo del sistema, es fundamental evaluar y asignar los costos asociados a cada uno de estos elementos. En la siguiente sección, se presenta el presupuesto del proyecto, proporcionando un desglose financiero.

4.2. Presupuesto

En esta sección se detallará el presupuesto necesario para la realización del proyecto, desglosado en dos categorías principales: el costo de los equipos físicos y el costo del personal involucrado.

⁴<https://www.onvif.org/profiles/specifications/>

⁵<https://tpp.hikvision.com/tpp/IntegrationCenter>

4.2.1. Coste de equipos físicos

Tabla 4.3: Se detallan los costos asociados a los componentes de hardware necesarios para el desarrollo del sistema de detección

Dispositivo	Coste
Portátil	319€
Cámara Térmica	59.033,64€*
4 GPUs NVIDIA RTX 3060 Ti12GB	2.600€*
Total	61.952,64€

Nota: Cabe destacar que el coste de la cámara es aproximado y no incluye los gastos de instalación. Igualmente para este TFG el acceso a los datos ha sido provisto por parte de la empresa como parte de un convenio de colaboración entre la empresa y la ULPGC. Se añade el precio orientativamente.

4.2.2. Coste personal

El costo del alumno ha sido estimado en base al valor de la beca de la Fundación Universitaria de Las Palmas (FULP) ⁶.

Tabla 4.4: Costo estimado del personal involucrado en el proyecto

Personal	Nº Horas	Coste/Hora	Coste
Alumno	300	7€	2.000€
Tutor 1	20	100€	2.000€
Tutor empresa	40	50€	2.000€
Total			6.000€

4.2.3. Presupuesto final

Tabla 4.5: Costo total del proyecto

Tipo de Coste	Coste
Coste de equipos físicos	61.952,64€
Coste personal	6.000€
Total	67.952,64€

Comprender el estado actual de la tecnología y las prácticas en el campo de la detección de embarcaciones es fundamental para situar el proyecto en el contexto correcto y comprender las decisiones tomadas. La siguiente sección proporciona una revisión exhaustiva del estado del arte, explorando las tecnologías existentes.

⁶<https://www.fulp.es/empresas/practicas/extracurriculares>

Capítulo 5

Desarrollo

5.1. Sprint Zero: Análisis y selección del modelo

Se ha llevado a cabo un análisis detallado para seleccionar el modelo más adecuado para la detección de barcos en imágenes térmicas, considerando los requisitos específicos del proyecto. En términos de precisión, es importante que el modelo sea lo suficientemente preciso para identificar los barcos en las imágenes térmicas. En este caso, Faster R-CNN es una alternativa que proporciona una alta precisión ya que se basa en propuestas de regiones, permitiendo una detección precisa y detallada. Sin embargo, esta precisión viene a costa de la velocidad de procesamiento, ya que Faster R-CNN es considerablemente más lento debido a su arquitectura de dos etapas.

CenterNet también ofrece una precisión competitiva y un enfoque innovador basado en puntos clave para detectar objetos, lo cual lo hace más rápido que Faster R-CNN pero generalmente más lento que YOLO. SSD ofrece una precisión razonable y se destaca en la detección de objetos de varios tamaños gracias a su uso de múltiples escalas de características, pero su precisión puede ser ligeramente inferior en comparación con Faster R-CNN y CenterNet.

La velocidad de procesamiento es un factor crítico, ya que el sistema debe funcionar en tiempo real. Aquí es donde YOLO sobresale, pues es capaz de procesar imágenes a una velocidad muy alta, lo que lo hace ideal para aplicaciones en tiempo real. Aunque Faster R-CNN ofrece una precisión superior, su lentitud lo hace menos adecuado para este proyecto. CenterNet, aún más rápido que Faster R-CNN, todavía no alcanza la velocidad de YOLO. SSD también ofrece una buena velocidad de procesamiento, pero nuevamente, YOLO ofrece un mejor equilibrio entre velocidad y precisión.

Dadas las capacidades de hardware disponibles compartidas entre todos los usuarios de la empresa, un modelo como YOLO, que es menos intensivo en recursos computacionales en comparación con Faster R-CNN, sería más adecuado. La implementación de YOLO es más escalable y eficiente, lo que lo hace más compatible con el hardware existente. Faster R-CNN, con su mayor complejidad, requeriría más recursos computacionales, lo que podría no ser viable con el hardware disponible.

El contexto de aplicación implica condiciones de baja visibilidad y adversas, típicas de las imágenes térmicas. YOLO ha demostrado ser eficaz en diversos entornos y condiciones, debido a su robustez y capacidad de detección rápida. Aunque CenterNet y SSD también son modelos robustos, YOLO ofrece un rendimiento superior en términos de velocidad y precisión combinada, lo cual es crucial en este contexto.

El conjunto de datos del que se dispone es relativamente limitado pero variado en términos de ángulos, tamaños y tipos de barcos. Esto hace que YOLO sea una opción adecuada, ya que es robusto cuando se trata de conjuntos de datos limitados y su capacidad para generalizar bien a partir de estas imágenes. Además, la detección de una sola clase de objetos, simplifica el proceso de entrenamiento y mejora el rendimiento. Faster R-CNN puede ofrecer una mejor precisión con más datos, pero su rendimiento no es tan eficiente con el conjunto de datos limitados.

En cuanto a la facilidad de implementación y compatibilidad, YOLO es la opción preferida, ya que la empresa ya tiene este modelo implementado para otras detecciones. Esto no solo facilita la integración, sino que también reduce los costos de implementación y el tiempo necesario para la puesta en marcha. Faster R-CNN y CenterNet, aunque técnicamente viables, requerirían una mayor inversión en tiempo y recursos para la implementación, lo cual no es ideal dado el plazo y presupuesto limitados.

La experiencia previa es únicamente con YOLO, lo que representa una ventaja significativa. Esto minimiza la curva de aprendizaje y asegura una implementación más rápida y eficiente. Modelos como Faster R-CNN, CenterNet y SSD, aunque podrían ofrecer ventajas en ciertas áreas, requerirían un período considerable de entrenamiento y adaptación, lo cual no es práctico dentro del plazo para llegar a los objetivos.

La Tabla a continuación 5.1 hace uso de colores y texto para evaluar distintos modelos de detección de objetos en varios factores. Por ejemplo, si un modelo requiere altos requisitos de hardware, significa que tiene un nivel alto en esta categoría, indicando que necesita un sistema potente para funcionar adecuadamente.

Modelo Factor	YOLO	Faster R-CNN	CenterNet	SSD
Precisión (mAP)	Alta	Muy Alta	Muy Alta	Alta
Velocidad (FPS)	Muy Alta	Muy Baja	Moderada	Muy Alta
Eficiencia en el entrenamiento	Alta	Moderada	Moderada	Alta
Requerimientos de Hardware	Moderada	Alta	Moderada	Moderada
Robustez en condiciones adversas	Muy Alta	Muy Alta	Muy Alta	Alta
Facilidad de Implementación	Muy Alta	Baja	Moderada	Alta
Compatibilidad	Alta	Baja	Moderada	Alta
Experiencia Previa	Alta	Baja	Baja	Baja

Tabla 5.1: Análisis de modelos de detección de objetos

En conclusión, considerando todos estos factores, YOLO se destaca como la opción más adecuada para la detección de barcos en imágenes térmicas. Su capacidad para mantener un alto rendimiento en términos de velocidad y precisión, la compatibilidad con el hardware y sistemas existentes, la facilidad de implementación y la experiencia del personal hacen de YOLO la elección más lógica y eficiente para este proyecto. Aunque otros modelos como Faster R-CNN, CenterNet y SSD tienen sus propias ventajas, las características específicas de YOLO lo hacen indiscutiblemente superior en este contexto, ofreciendo un balance óptimo entre precisión, velocidad y facilidad de integración.

5.2. Sprint 1: Captura de Imágenes de la Cámara Térmica

En este sprint, se utilizarán las librerías ISAPI y ONVIF para establecer la conexión con la cámara dual-térmica. Una vez establecida la conexión, se procederá a la configuración y ejecución de las operaciones necesarias para capturar imágenes térmicas. La cámara dual-térmica permite generar distintos presets en diferentes posiciones. Estos presets son configuraciones predefinidas que permiten enfocar automáticamente la cámara en zonas específicas de interés, como un aerogenerador o un puerto, con distintos niveles de zoom. Esto facilita la obtención de imágenes detalladas y relevantes de áreas críticas.

El siguiente Algoritmo 1 detalla el proceso de conexión a la cámara, la obtención de las imágenes y su almacenamiento local:

Algorithm 1 Proceso para capturar imágenes en la cámara térmica de la zona del Mirador

```
1: Create camera instance
2: Create PTZ service
3: Initialize presets with positions, zoom levels, and identifiers
4: if presets already defined then
5:   Continue to the next step
6: end if
7: for each preset in presets do
8:   Create camera instance
9:   Create PTZ service
10:  attempts  $\leftarrow$  0
11:  repeat
12:    Move to preset position
13:    current_position  $\leftarrow$  get_camera_position(ptz_services)
14:    if current_position  $\neq$  preset.position then
15:      attempts  $\leftarrow$  attempts + 1
16:      if attempts > max_attempts then
17:        Log error and terminate execution
18:      end if
19:    end if
20:  until current_position == preset.position
21:  Send request to capture image
22:  response  $\leftarrow$  REQUESTS.GET(URL)
23:  if response.status_code == 200 then
24:    image_object  $\leftarrow$  IMAGE.OPEN(BYTESIO(RESPONSE.CONTENT))
25:    Save image_object locally with identifier and timestamp
26:  else
27:    Log error with response status code
28:  end if
29: end for
```

5.3. Sprint 2: Evaluación Dataset COCO

Para detectar embarcaciones en cámaras térmicas, inicialmente se decidió utilizar el dataset COCO de YOLO para evaluar su precisión y determinar si puede satisfacer las necesidades del proyecto. El dataset COCO es conocido por su eficiencia y precisión en la detección de objetos en imágenes de diversos contextos, lo que lo convirtió en una opción atractiva para una evaluación preliminar.

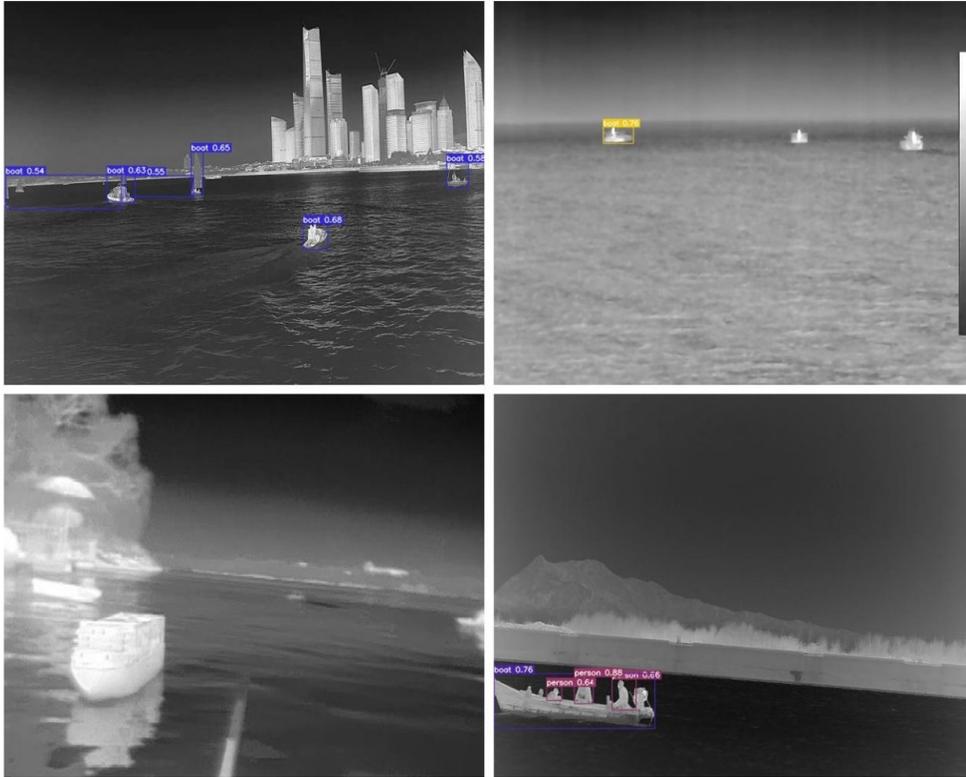


Ilustración 5.1: Resultados de inferencia con el dataset COCO con imágenes térmicas

Durante las pruebas, el dataset COCO mostró cierta capacidad para detectar barcos en imágenes térmicas, como se observa en la Ilustración 5.1. Sin embargo, estas detecciones ocurrieron en un número muy limitado de casos. El dataset presentó dificultades para identificar correctamente los barcos en la mayoría de las imágenes debido a las características específicas de las imágenes térmicas, que difieren significativamente de las imágenes convencionales utilizadas para entrenar el modelo COCO.

Dada la baja tasa de detección, se ha decidido que no es viable utilizar el dataset COCO de YOLO tal cual para nuestro propósito. Por lo tanto, el objetivo principal será desarrollar un dataset para la detección de barcos desde cero.

5.4. Sprint 3: Evaluación Inicial y Preparación del Modelo Base

5.4.1. Análisis de Datos Iniciales

La detección de objetos en imágenes térmicas es crucial para la vigilancia, sin embargo, la falta de un dataset robusto y general requirió comenzar desde cero. El conjunto de datos original surge de la integración de varios datasets, cada uno procedente de una localización específica. Dicha unificación nos llevó a observar una notable repetición de imágenes de barcos a lo largo de estos conjuntos de datos, como es el caso de la Ilustración 5.2. Para proteger la calidad y el equilibrio del conjunto de datos, se llevó a cabo un minucioso análisis individual de cada imagen para evitar duplicidades y redundancias, resultando en un dataset final compuesto por 315 imágenes, distribuidas en un 70 % para entrenamiento y un 30 % para pruebas.

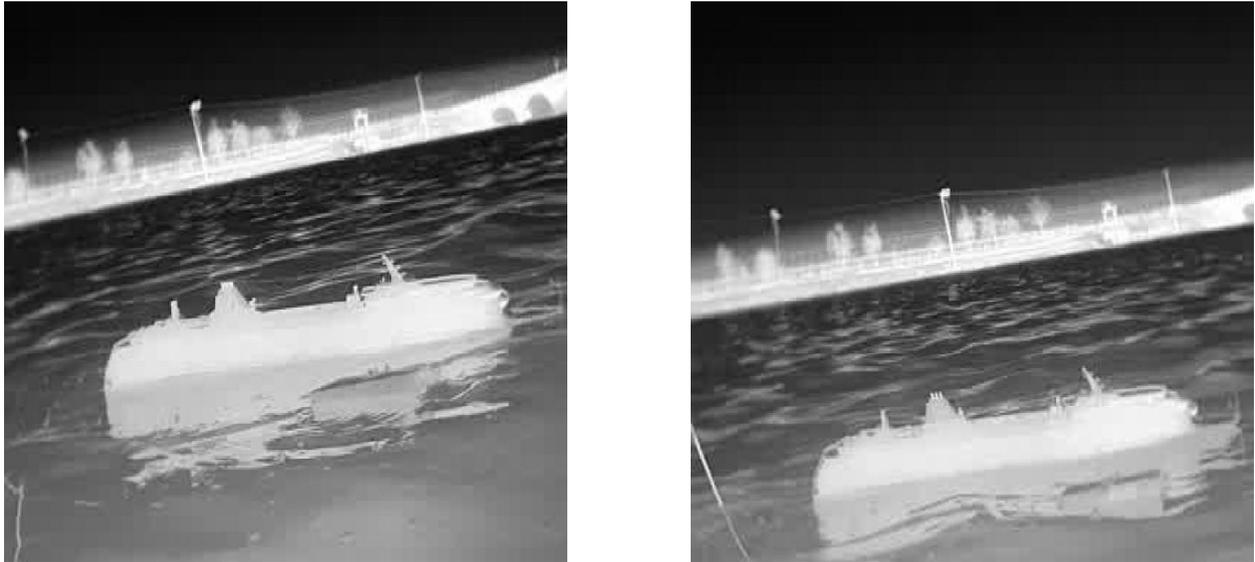


Ilustración 5.2: Ejemplo de imágenes con el mismo barco y entorno, mostrando la similitud de imágenes dentro de un conjunto de datos. Imagen obtenida de [30]

Es importante destacar que las imágenes se emplean en el ámbito de la detección de barcos sin una clasificación específica por tipo de embarcación. Esta falta de clasificación puede repercutir en el análisis del balance de clases dentro del dataset, afectando a la precisión de los modelos finales. Además, el tamaño limitado del conjunto de datos, con tan solo 315 imágenes, puede resultar insuficiente para entrenar modelos de aprendizaje profundo con capacidad de generalización adecuada.

También es relevante mencionar que algunos de los conjuntos de datos contribuyentes consisten en imágenes en blanco y negro (véase la Ilustración 5.3), sin incluir imágenes térmicas. Asimismo, se identificaron imágenes con gran dificultad, caracterizadas por la presencia de

múltiples oclusiones (véase la Ilustración 5.4), lo cual representó un desafío adicional para el dataset original. Debido a la unión de varios conjuntos de datos, las etiquetas asociadas a las imágenes eran distintas, siendo necesario utilizar únicamente la etiqueta de presencia de barco para mantener la coherencia en el conjunto de datos.



Ilustración 5.3: Imagen en blanco y negro de un entorno marítimo. Imagen obtenida de [10]



Ilustración 5.4: Imagen con barcos ocluidos. Imagen obtenida de [9]

5.4.2. Entrenamiento del Modelo Inicial

Tras completar el entrenamiento con el conjunto de datos inicial, se procede a examinar los resultados obtenidos en la Ilustración 5.5. Un 67% de los barcos se identifican correctamente, reflejado en la tasa de verdaderos positivos, indicando una capacidad moderada del modelo para reconocer barcos. Sin embargo, la tasa de fallos es elevada, con un 33% de falsos negativos, lo cual es preocupante y sugiere la necesidad de mejorar la fiabilidad del modelo.

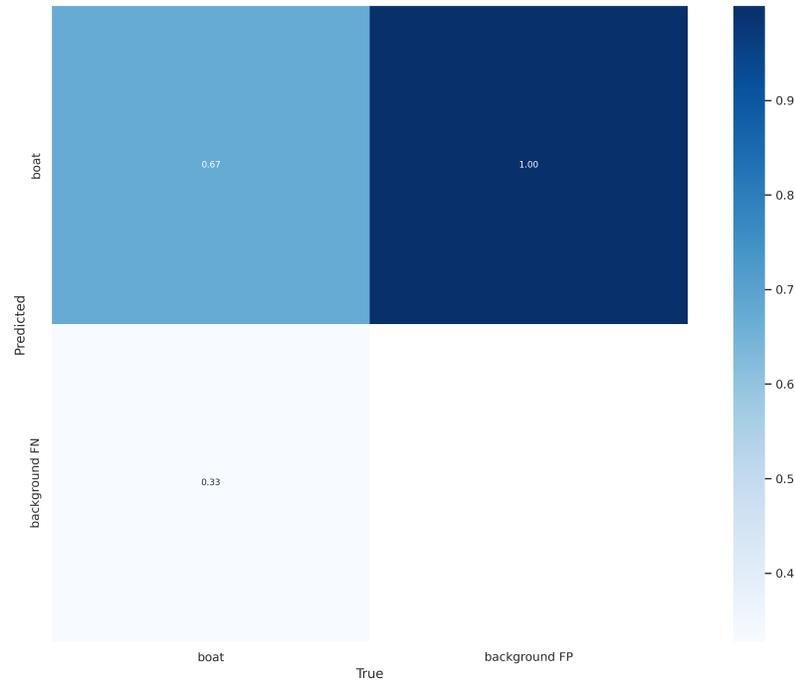


Ilustración 5.5: Matriz de confusión del entrenamiento del dataset inicial con distribución aleatoria.

En cuanto a las métricas de la Ilustración 5.6, Box disminuye aproximadamente de 0.7 a 0.3, con algunas fluctuaciones a lo largo del entrenamiento. La val Box disminuye de 0.12 a 0.07, aunque con considerables fluctuaciones, indicando cierta inestabilidad en el rendimiento del modelo durante la validación. Las fluctuaciones en la pérdida de cajas durante la validación son mayores que las observadas durante el entrenamiento. La precisión aumenta, pero con fluctuaciones bruscas. Mientras que el Recall también muestra un incremento significativo, pero con menos fluctuaciones en comparación con la precisión.

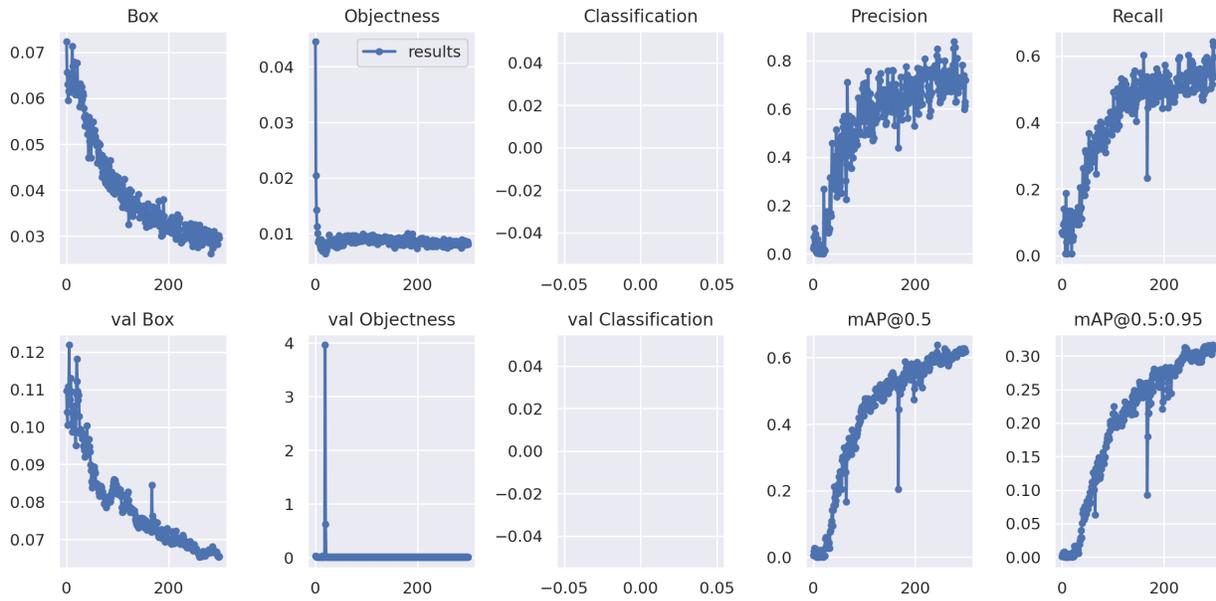


Ilustración 5.6: Resultados del entrenamiento del dataset inicial con distribución aleatoria.

El modelo es insuficiente para la detección de barcos, por lo que hay un margen significativo de mejora. La alta tasa de falsos negativos es preocupante y debe abordarse para mejorar la fiabilidad del modelo.

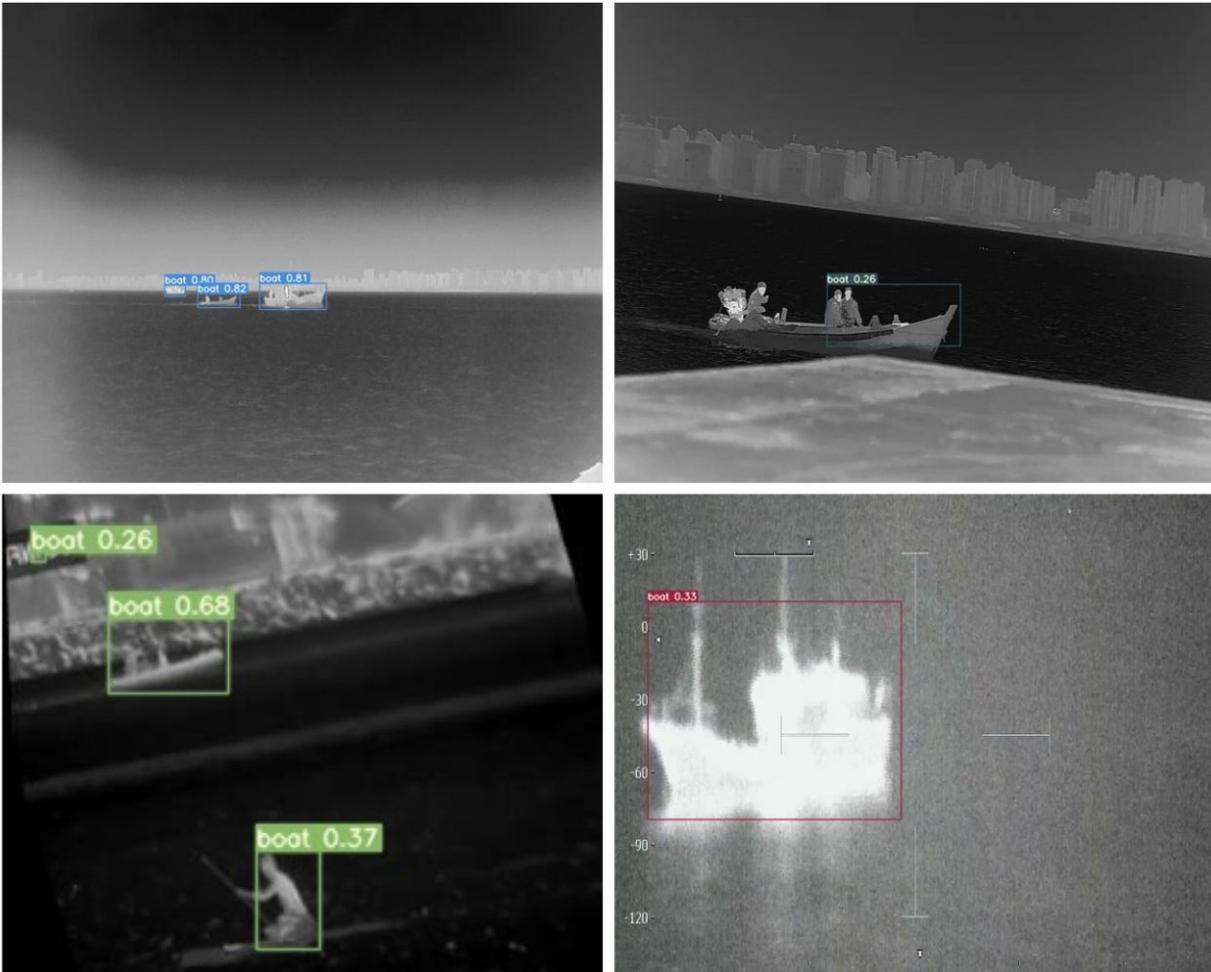


Ilustración 5.7: Detecciones realizadas por el modelo inicial con distribución aleatoria

A pesar de haber detectado barcos con una puntuación alta, como se observa en la Ilustración 5.7, algunos tienen una puntuación bastante baja. Se observan casos donde objetos parcialmente visibles o cortados son incorrectamente detectados como barcos, posiblemente debido a la similitud con la estructura de un barco. Además, no todos los barcos están completamente contenidos dentro de los cuadros delimitadores, lo que puede indicar problemas de detección parcial. Otro desafío identificado es la detección de barcos en medio de estructuras o vegetación en el fondo de la imagen, complicando la identificación y clasificación correcta de los objetos. Además, se presentan complicaciones en casos de situaciones adversas como la calima.

5.5. Sprint 4: Distribución de Datos

En la primera fase de nuestro proyecto, se realizaron entrenamientos utilizando una distribución aleatoria de los datos. Aunque esta estrategia funcionaba correctamente en algunas

ocasiones, en otros casos se observaban problemas significativos en la matriz de confusión, con valores que llegaban a 0,1. Esto indicaba que, para nuestro dataset inicial, la distribución aleatoria era descontrolada y no proporcionaba resultados consistentes ni fiables.

Para abordar este problema, se decidió implementar una distribución basada en histogramas como se observa en el Algoritmo 2. En este enfoque, los valores extremos (tanto altos como bajos) se asignaron al conjunto de entrenamiento, mientras que los valores medios se destinaron al conjunto de validación, manteniendo una proporción de 70 % para entrenamiento y 30 % para validación. Este método permitió una distribución más equilibrada y controlada de los datos.

Como se mencionó anteriormente, al no basarse en una clasificación por tipos de barcos, se descartó una distribución por clases. Por lo tanto, se dio un enfoque en el análisis del contenido de las imágenes. Este método de distribución por histogramas resultó ser adecuado. Las imágenes térmicas, al tener rangos de valores similares, se prestan a este tipo de distribución. La implementación de histogramas para organizar las imágenes según sus características intrínsecas permitió mantener la homogeneidad en los conjuntos de datos y mejorar la eficacia del entrenamiento de los modelos.

Al asignar los valores extremos al entrenamiento y los valores medios a la validación, aseguramos que el modelo se entrene con una gama amplia de datos, mientras se valida con un subconjunto representativo y balanceado. Esto mejora la capacidad del modelo para generalizar a nuevos datos.

La matriz de confusión generada a partir del entrenamiento del dataset inicial con distribución por histogramas se presenta en la Ilustración 5.8. Esta matriz muestra un valor de 66 % para la clase *boat*. Este resultado indica que el modelo tiene una precisión moderada en la identificación de barcos en las imágenes térmicas. Además, el 34 % de las instancias de *boat* fueron incorrectamente clasificadas como *background FN*, mostrando que el modelo falla en detectar algunos barcos, lo cual afecta negativamente al recall.

Algorithm 2 Preprocesamiento y División de Imágenes en Conjuntos de Entrenamiento y Validación mediante Histogramas

```

1: Initialize  $images \leftarrow []$  and  $labels \leftarrow []$ 
2: Set  $target\_size \leftarrow (1280, 1280)$ 
3:  $image\_names \leftarrow$  list of files in  $images\_folder\_path$ 
4:  $label\_names \leftarrow$  list of files in  $labels\_folder\_path$ 
5: for each  $filename$  in  $image\_names$  do
6:    $image\_path \leftarrow images\_folder\_path + filename$ 
7:    $image \leftarrow$  read image from  $image\_path$  using OpenCV
8:   try
9:      $image \leftarrow$  resize image to  $target\_size$ 
10:  catch Exception as  $e$ 
11:    print error message
12:   $image \leftarrow$  resize image to  $target\_size$ 
13:  if  $image$  is not None then
14:    append  $image$  to  $images$ 
15:     $label\_path \leftarrow$  corresponding label file path
16:    if label file exists then
17:       $label \leftarrow$  read label from  $label\_path$ 
18:      append  $label$  to  $labels$ 
19:    end if
20:  end if
21: end for
22: Convert  $images$  and  $labels$  to numpy arrays
23:  $image\_features \leftarrow$  compute color histograms for all  $images$ 
24:  $distances \leftarrow$  compute Euclidean distances between  $image\_features$ 
25:  $sorted\_indices \leftarrow$  sort indices based on  $distances$ 
26: Split indices into training and validation sets
27: for each  $idx$  in training set do
28:   Copy image and label to training folder
29: end for
30: for each  $idx$  in validation set do
31:   Copy image and label to validation folder
32: end for
33: Print the number of images and labels copied to each set

```

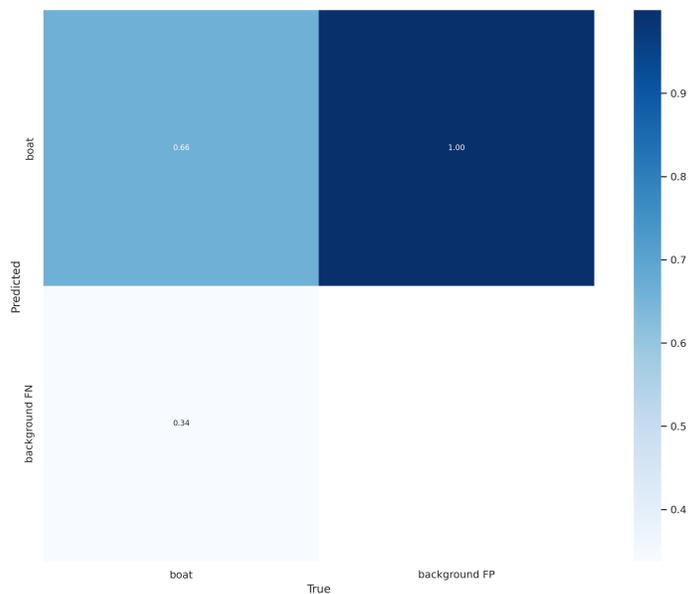


Ilustración 5.8: Matriz de confusión del entrenamiento del dataset inicial con distribución de histograma

Esto sugiere que, aunque el modelo ha aprendido a reconocer barcos en más de la mitad de los casos, todavía hay un margen considerable para mejoras en términos de precisión y reducción de falsos positivos y falsos negativos.

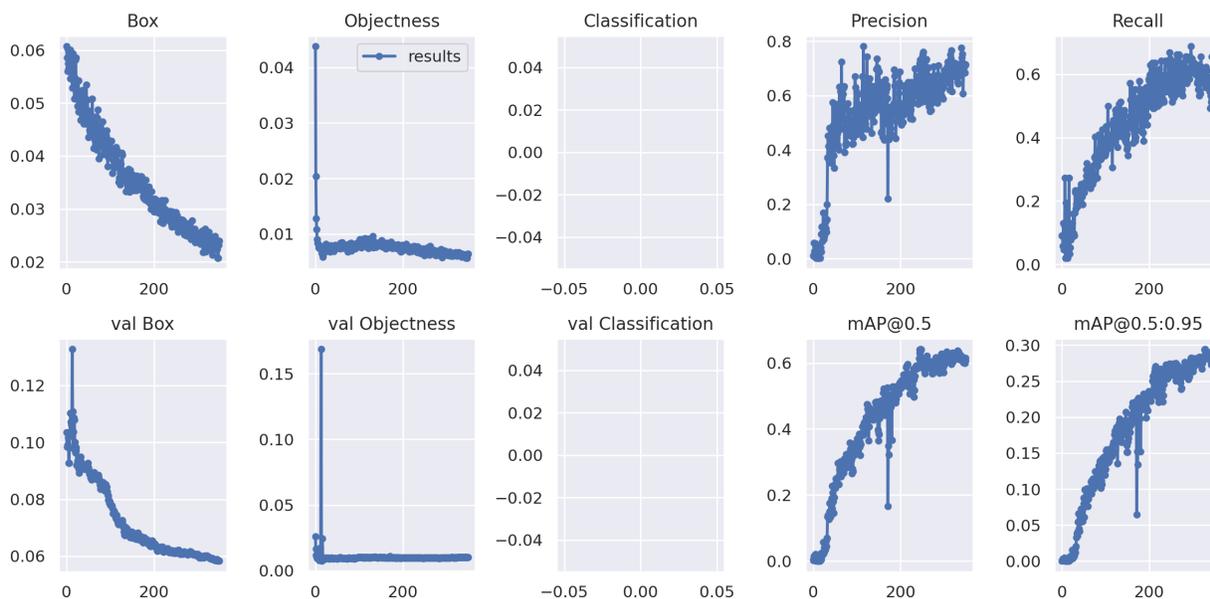


Ilustración 5.9: Resultados del entrenamiento del dataset inicial con distribución de histograma

Por otro lado, en la Ilustración 5.9, se observa que la métrica Box disminuye aproxima-

damente de 0.06 a 0.02, con fluctuaciones menores en comparación con el dataset inicial con distribución de histograma. En cuanto a la val Box, muestra una disminución de 0.12 a 0.06, estabilizándose antes y con menores fluctuaciones. Esto indica una mejor estabilización y menor variabilidad, lo que sugiere que el modelo generaliza mejor y tiene un rendimiento más confiable en datos de validación. La Precisión aumenta rápidamente, aunque con fluctuaciones y caídas bruscas. El Recall presenta más fluctuaciones pero menos caídas abruptas en comparación con la distribución aleatoria.

En la siguiente Ilustración 5.10, se muestra la inferencia del Dataset Inicial con distribución mediante histogramas. Al igual que el Dataset Inicial con distribución aleatoria, tiene valores estables y bajos en las detecciones del modelo. Los problemas observados en la inferencia de distribución aleatoria, como detecciones incompletas y falsos positivos, se reflejan en estos resultados también, pero en menor medida.

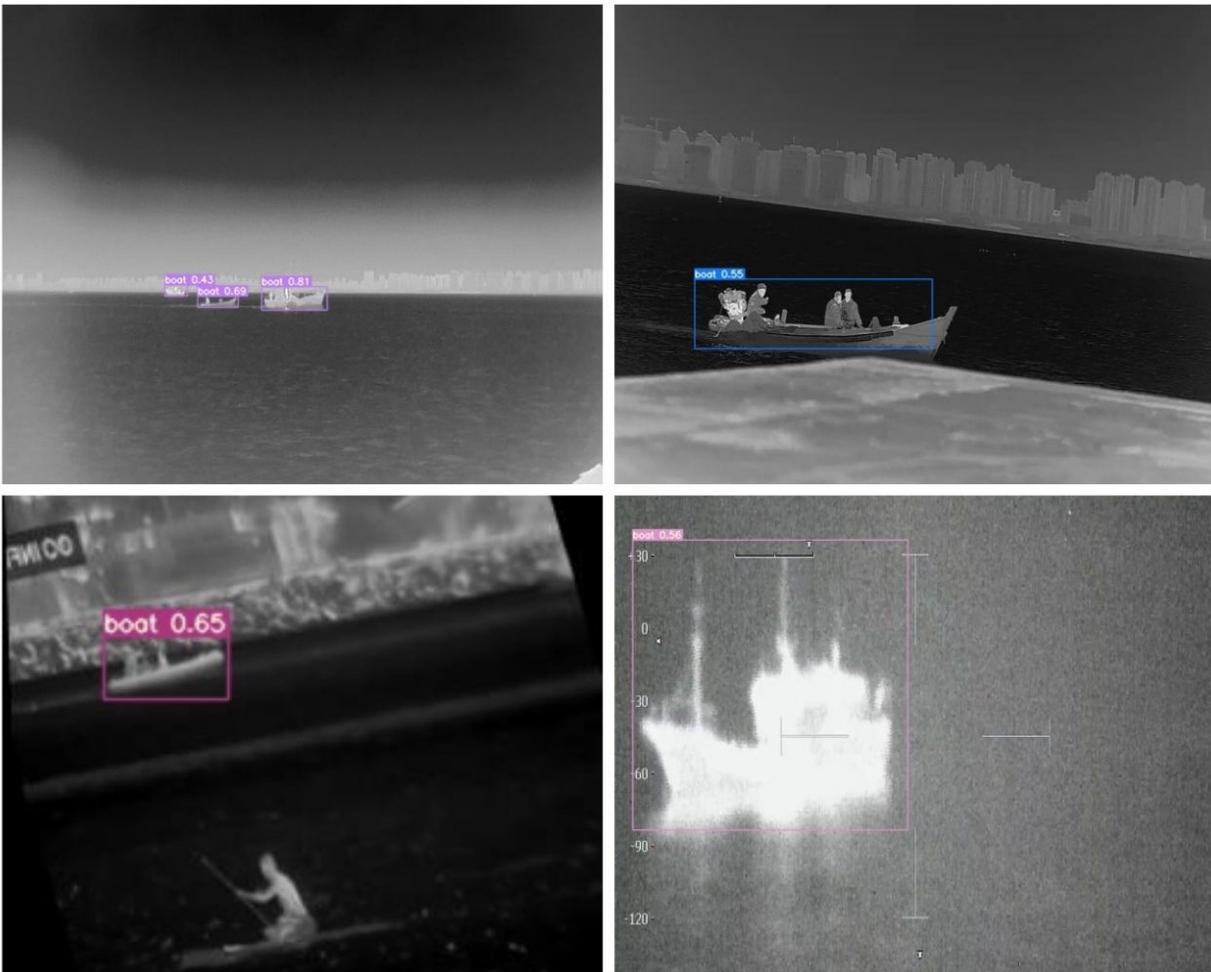


Ilustración 5.10: Detecciones realizadas con el dataset inicial con distribución de histograma

El modelo está lejos de ser perfecto y tiene dificultades con ciertas imágenes que contienen barcos, posiblemente debido a variaciones en las condiciones de las imágenes térmicas. Dado el

resultado, se ha optado por ampliar el dataset mediante varias propuestas que se comentarán posteriormente.

5.6. Sprint 5: Creación y Evaluación de Imágenes Sintéticas

5.6.1. IRSships

Para generar imágenes sintéticas, se ha utilizado un conjunto de datos de imágenes infrarrojas (IR) llamado IRShips, creado en CounterSim, un software de simulación que genera imágenes en diversas longitudes de onda, como la infrarroja.

Tipos y Clases de Barcos

Este dataset incluye cuatro tipos y diez clases de barcos, como por ejemplo, Ferry Armourique (ver Ilustración 5.11) o Destroyer Akizuki (ver Ilustración 5.12).

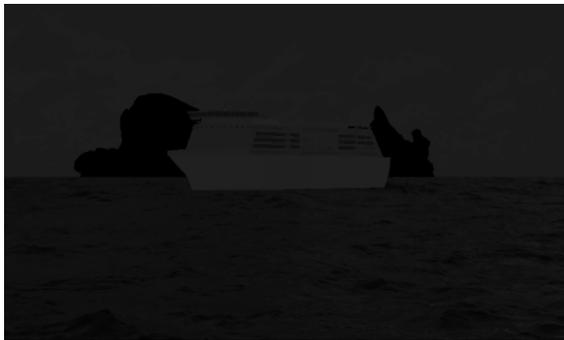


Ilustración 5.11: Ejemplo de Barco Ferry Armourique generado con [49]



Ilustración 5.12: Ejemplo de Barco Destroyer Akizuki generado con [49]

Estas imágenes simulan ser capturadas desde diversas distancias, orientaciones y ángulos de elevación, con una resolución de 1024x512 píxeles.

Generación de Imágenes Sintéticas

Para acceder a las imágenes sintéticas, se utiliza el archivo *urls.yaml*, que contiene la URL de la imagen, la referencia, la licencia y las transformaciones aplicadas a la imagen. Además, se utiliza un archivo de metadatos llamado *metadata.csv*, donde se especifica:

- Nombre de la imagen
- Tipo y clase de barco
- Temperatura del barco, del mar y del ambiente
- Posición y etiqueta

Además, se emplean máscaras binarias (ver las Ilustraciones 5.13, 5.14, 5.15 y 5.16) para cada uno de los componentes de la imagen, lo que permite un mejor manejo de los datos y facilita la segmentación de los objetos de interés.

Ejemplo de Máscaras Binarias

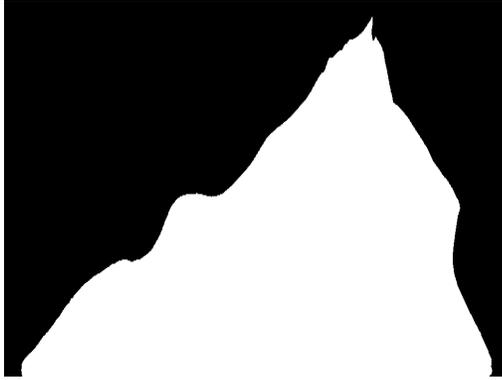


Ilustración 5.13: Máscara iceberg. Imagen obtenida de [49]

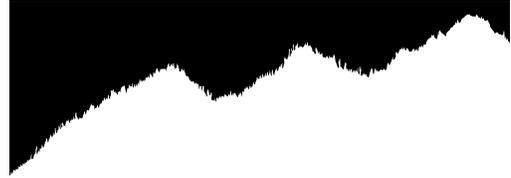


Ilustración 5.14: Máscara plataforma petrolífera. Imagen obtenida de [49]



Ilustración 5.15: Máscara montaña. Imagen obtenida de [49]



Ilustración 5.16: Máscara cielo. Imagen obtenida de [49]

Resultados



Ilustración 5.17: Imágenes sintéticas generadas con apariencia térmica del entorno marítimo a una temperatura de seis grados. Imagen generada con IRSships [49]



Ilustración 5.18: Imágenes sintéticas generadas con apariencia térmica del océano a una temperatura de diecisiete grados. Imagen generada con IRSships [49]



Ilustración 5.19: Imagen sintética generada con apariencia térmica de un barco con temperatura de cuatro grados. Imagen generada con IRSships [49]

Es crucial entender que los resultados (ver las Ilustraciones 5.17, 5.19 y 5.18) generados por IRShips son imágenes infrarrojas sintéticas, no imágenes térmicas reales. Aunque estas imágenes intentan emular la apariencia de las imágenes térmicas, presentan diferencias significativas. Las imágenes térmicas reales capturan directamente la radiación infrarroja emitida por los objetos, mientras que las imágenes sintéticas son simulaciones que pueden no reflejar con precisión todas las complejidades del entorno real. Estas variaciones pueden resultar en colores distintos para la misma temperatura, lo que puede confundir al modelo al interpretar los datos. Dado que nuestro modelo requiere precisión y coherencia en la representación de la temperatura, las imágenes generadas por IRShips no son adecuadas para este contexto.

5.7. Sprint 6: Entrenamiento de GAN para la Generación de Imágenes

5.7.1. CycleGAN

Esta sección se centra en los desafíos y limitaciones inherentes de CycleGAN en la generación de imágenes térmicas a partir de imágenes RGB. Se llevará a cabo un análisis detallado de los obstáculos encontrados. Además, se examinarán las implicaciones de estas limitaciones en la detección y clasificación de embarcaciones en entornos marítimos.

Limitaciones en el Acceso a Datos

Aunque se ha creado un dataset por el autor, como ThermalWorld, para entrenar modelos de traducción de imágenes, no se ha podido obtener acceso a este dataset después de múltiples intentos. Esta falta de acceso restringe la capacidad de realizar pruebas y evaluaciones exhaustivas.

Desafíos en la Interpolación de Imágenes

Al utilizar la interpolación y probar los diferentes métodos de interpolación proporcionados por CycleGAN, se observó que el modelo, por defecto, trabaja con imágenes de tamaño 256x256, se intentó entrenar el modelo base utilizando imágenes de tamaño 1280x1280, pero el hardware no tenía la capacidad para manejar esta resolución.

Posteriormente, tras entrenar el modelo con las imágenes de 256x256 píxeles, se intentó generar imágenes con una resolución de 1280x1280 píxeles, sin embargo, las imágenes generadas se asemejaban más a imágenes en blanco y negro en lugar de imágenes térmicas. A pesar de que las imágenes en 256x256 se etiquetaron y se añadieron al dataset inicial, los resultados empeoraron en comparación con el entrenamiento sin estas nuevas imágenes.



Ilustración 5.20: Imágenes de barcos generadas mediante CycleGAN [23] [52] de 1280x1280px.

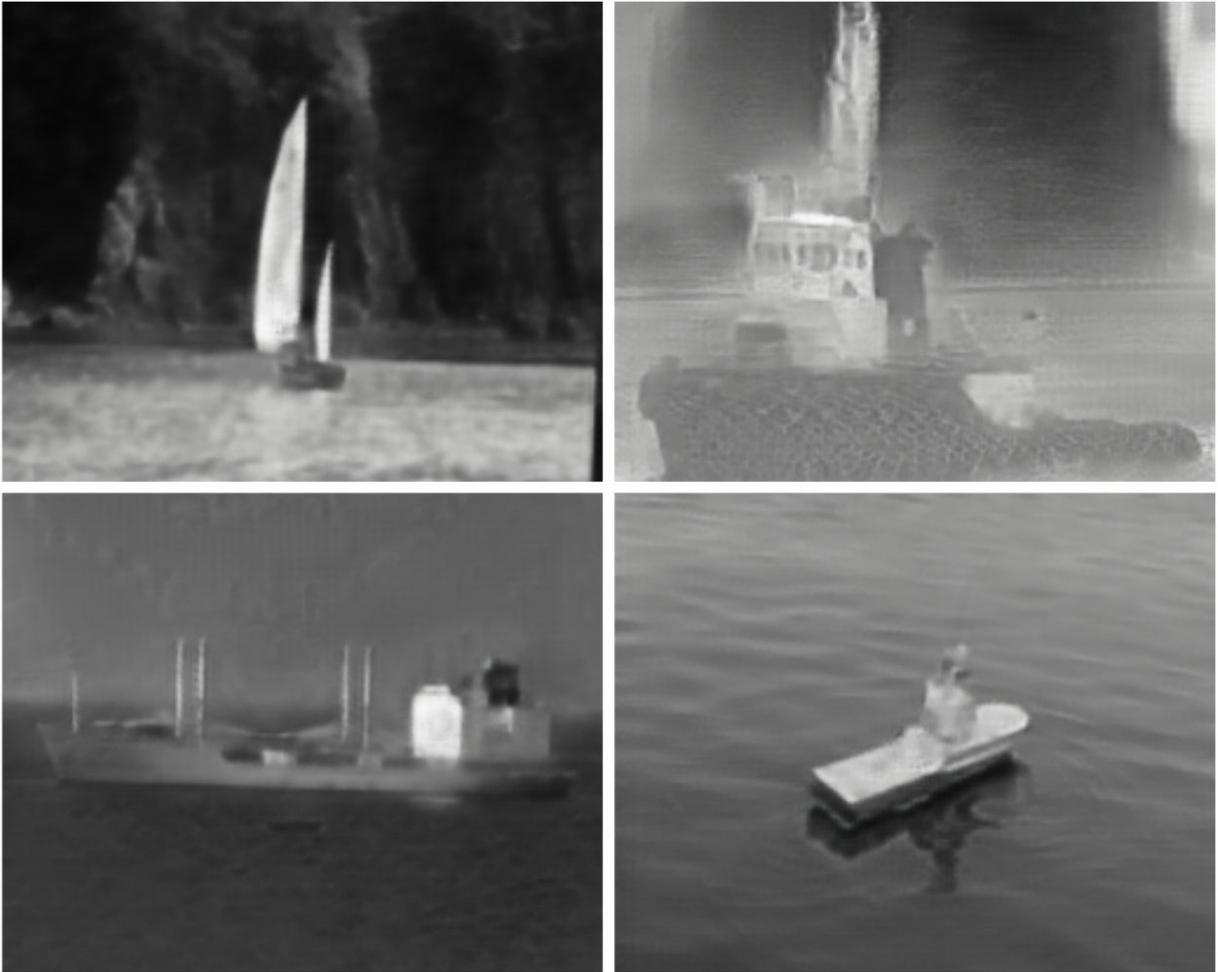


Ilustración 5.21: Imágenes de barcos generadas mediante CycleGAN [23] [52] de 256x256px.

La Ilustración 5.21 muestra los resultados obtenidos al generar imágenes de tamaño 256x256 píxeles, mientras que la Ilustración 5.20 presenta los resultados obtenidos con imágenes de tamaño 1280x1280 píxeles. Estos resultados destacan los desafíos significativos encontrados al aplicar la interpolación y el escalado de imágenes en CycleGAN para la generación de imágenes térmicas.

Implementación y Evaluación

El desempeño del modelo se ve afectado por la inclusión de imágenes generadas por CycleGAN. Estas imágenes, destinadas a representar las imágenes térmicas, no logran una reproducción exacta de las imágenes térmicas reales, lo cual tiene un impacto negativo en la capacidad del modelo para realizar sus tareas. Además, la generación de imágenes por parte de CycleGAN puede generar un desequilibrio en las clases de barcos, ya que los tipos de barcos presentes en las imágenes térmicas pueden diferir considerablemente de los que están presentes en las imágenes ópticas empleadas para generar las imágenes térmicas. Por ejemplo, mientras que en las imágenes térmicas pueden predominar veleros, en las imágenes

ópticas pueden predominar barcos de carga, lo que complica la capacidad del modelo para generalizar adecuadamente entre ambos tipos de imágenes.

El proceso de entrenamiento del modelo implicó la combinación de imágenes de barcos de un conjunto de datos inicial con imágenes generadas por CycleGAN, con el propósito de ampliar el tamaño del conjunto de datos. Este proceso resultó en un conjunto de datos combinado de 416 imágenes, incluyendo las 315 imágenes del conjunto de datos inicial y 101 imágenes generadas por CycleGAN.

En la Ilustración 5.22, se observa que el modelo logró identificar correctamente el 32 % de los barcos térmicos, mostrando una precisión baja en la clasificación de embarcaciones, lo que refleja una precisión baja en la clasificación de embarcaciones. Este resultado sugiere que el modelo tiene dificultades para distinguir adecuadamente los barcos del fondo en las imágenes térmicas.

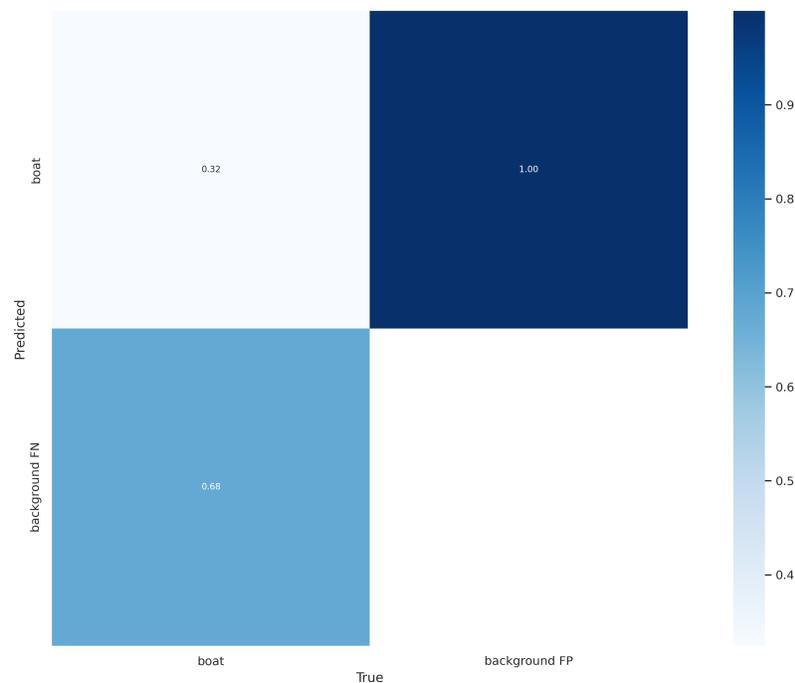


Ilustración 5.22: Matriz de confusión del entrenamiento del dataset inicial y de las imágenes generadas con CycleGAN.

Los resultados del entrenamiento se observan en la Ilustración 5.23. En ella se revela que la precisión contiene grandes fluctuaciones, cayendo desde aproximadamente 0.8 hasta casi 0.4 en algunos puntos. Estas caídas significativas indican inestabilidad en la capacidad del modelo para identificar correctamente los barcos, posiblemente debido a variaciones en las condiciones de las imágenes. Esta inestabilidad es una de las métricas más preocupantes, ya que afecta directamente la fiabilidad del modelo en la clasificación de embarcaciones.

Asimismo, el recall también muestra fluctuaciones, indicando la inconsistencia en la detección completa de los barcos. Las fluctuaciones en el recall son igualmente preocupantes,

ya que reflejan una variabilidad en la detección de barcos, afectando la cobertura total de detección.

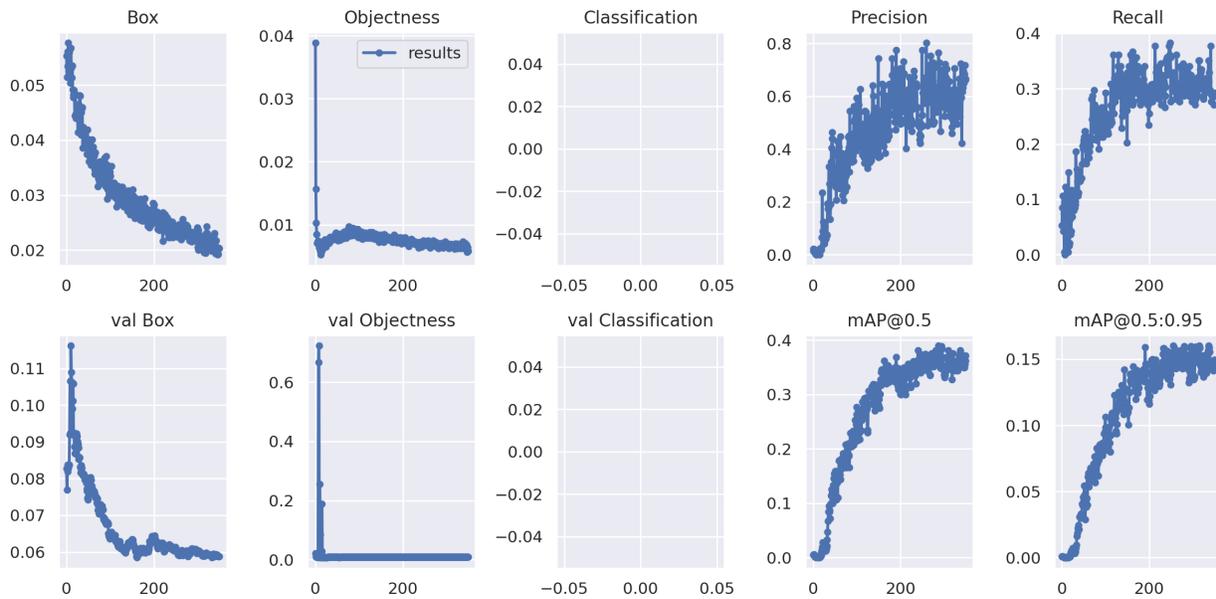


Ilustración 5.23: Resultados del entrenamiento del conjunto de datos inicial e imágenes generadas con CycleGAN.

Tanto Box como Objectness muestran fluctuaciones considerables al inicio del entrenamiento y validación, estas se estabilizan al final, indicando un ajuste progresivo del modelo.

Al comparar la precisión de detección con el sprint previo, se ha observado una disminución significativa al agregar imágenes generadas con CycleGAN, lo que provocó una duplicación de la tasa de falsos negativos y una mayor cantidad de barcos no detectados en la predicción del modelo actual. A causa de esto, las imágenes generadas por CycleGAN en el dataset han deteriorado la precisión del modelo para detectar embarcaciones. El dataset inicial muestra una mejor capacidad para identificar correctamente los barcos y presenta una tasa de falsos negativos notablemente inferior. A continuación, en la Ilustración 5.24 se examinan los resultados de la inferencia, los cuales muestran el desempeño del modelo en distintos contextos.

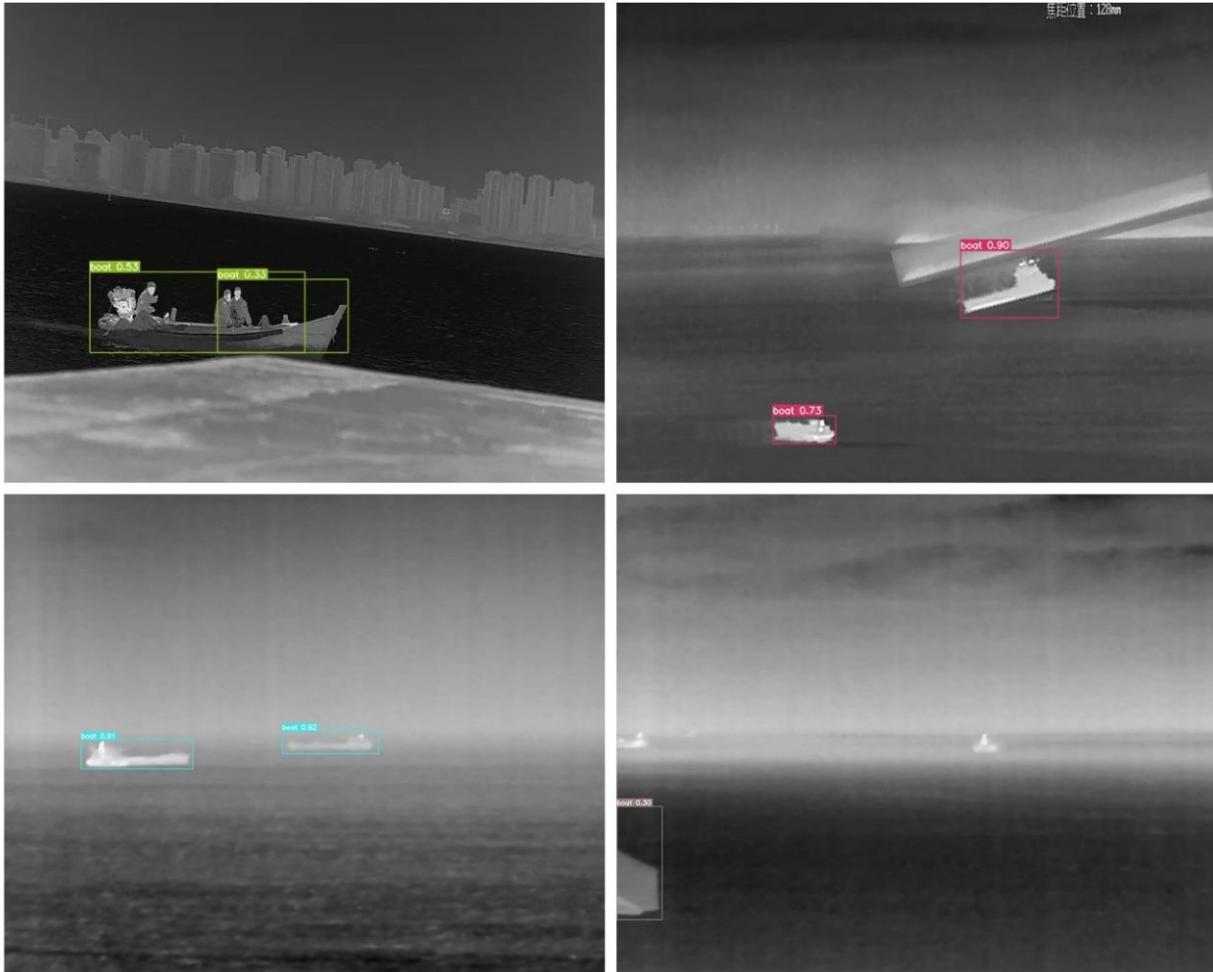


Ilustración 5.24: Detecciones realizadas con el dataset inicial y CycleGAN.

A pesar de que CycleGAN ofrece una solución robusta para la traducción de imágenes entre dominios sin necesidad de datos emparejados, se han encontrado limitaciones significativas en su efectividad, específicamente en la generación de imágenes térmicas a partir de imágenes RGB. La diversidad de esquemas de colores presentes en las imágenes RGB dificulta la traducción precisa a imágenes térmicas, lo que afecta el rendimiento del modelo de detección. Por lo tanto, aunque CycleGAN sigue siendo una herramienta valiosa en muchos contextos, para la generación de imágenes térmicas a partir de imágenes RGB, se requieren enfoques alternativos que aborden específicamente esta complejidad de datos.

5.8. Sprint 7: Técnicas de Composición de Imágenes

5.8.1. Blended Diffusion

En esta sección, se exploran los desafíos encontrados durante la aplicación de la técnica de *Blended Diffusion*[4] en la generación de imágenes. Se abordan tanto los resultados obtenidos como los inconvenientes enfrentados durante el proceso.

Preparación de datos

Para aplicar *Blended Diffusion* de manera efectiva, es crucial seleccionar imágenes de fondo que proporcionen un entorno adecuado y coherente para la generación de las nuevas composiciones visuales. Se han utilizado imágenes del entorno del mirador o datasets de imágenes marítimas sin barcos [35] para realizar pruebas preliminares y ajustes en el proceso.

La elección de máscaras adecuadas para guiar el proceso de *Blended Diffusion* también presenta sus propios desafíos. Uno de los inconvenientes principales es la variación en el tamaño de las máscaras y su impacto en la calidad de los resultados. A medida que las máscaras se modifican en tamaño, se vuelven más propensas a generar anomalías cromáticas e inconsistencias visuales en las imágenes generadas.

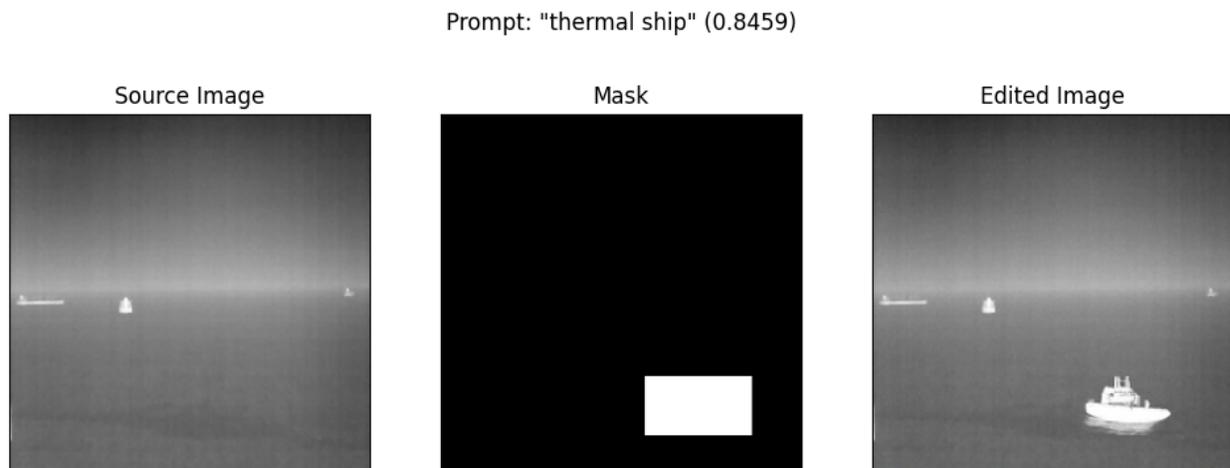


Ilustración 5.25: Proceso de aplicación de *Blended Diffusion*. Primero, se muestra la imagen de fondo, seguida de la imagen de la máscara que se aplicará. Finalmente, se presenta el resultado final obtenido al combinar la imagen de fondo y la máscara mediante la técnica de *Blended Diffusion*. Imagen generada con el código publicado en [4].

Resultados

Esta sección presenta los resultados obtenidos mediante *Blended Diffusion*. Se incluyen

ejemplos de imágenes generadas utilizando esta técnica para mostrar los resultados alcanzados.



Ilustración 5.26: Imágenes generadas con *Blended Diffusion* [4]

Inconvenientes

A pesar de obtener algunos resultados positivos, las imágenes presentan inconsistencias significativas en relación con el entorno y el color representativo de la temperatura. Estas inconsistencias se presentan de diversas maneras: las imágenes pueden resultar más oscuras o más claras de lo que requiere el entorno circundante o en comparación con el resto de los barcos presentes en la imagen.

Además, aunque en el *prompt* se especifique claramente un barco térmico, frecuentemente se observan anomalías cromáticas, como se observa en la Ilustración 5.27, con colores no deseados en los barcos o incluso manchas de color que no se ajustan a las especificaciones. Este problema aumenta con respecto al tamaño de la máscara aplicada, cuanto mayor es la máscara, más evidentes son los colores. En el caso de máscaras pequeñas, en muchas ocasiones, la generación falla por completo y no se dibuja nada en la imagen.

Otro problema es la falta de contexto en la generación de imágenes, lo que resulta en imágenes con sombras inapropiadas, tal y como se analiza en la Ilustración 5.28, o sobre un remolque; características que no están relacionadas con imágenes de barcos en mar abierto. A pesar de especificar el entorno en el *prompt*, se ha observado que los resultados son mejores cuando el *prompt* es más corto y tiene menos información.

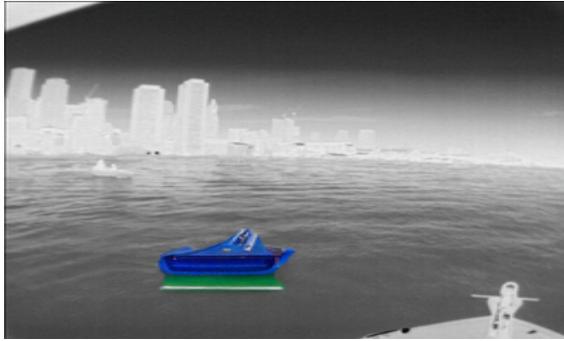


Ilustración 5.27: Barco a color. Imagen generada con [4]



Ilustración 5.28: Barco con sombra. Imagen generada con [4]

Se utilizó un modelo de difusión con una resolución de 256x256 píxeles, ya que no se pudo utilizar uno de 512x512 píxeles debido a limitaciones de hardware, lo que impidió la ejecución del programa con esta configuración. El uso de un modelo de mayor resolución permitirá una mejor calidad en las imágenes generadas, lo cual es crucial para aplicaciones más precisas y detalladas.

Por lo tanto, los resultados satisfactorios son escasos y el proceso de generación y verificación de estas imágenes consume una cantidad considerable de tiempo y recursos computacionales.

5.8.2. OpenCV

Una de las capacidades clave de OpenCV es su capacidad para realizar composición de imágenes, la cual es útil en una variedad de aplicaciones, como la superposición de imágenes, y la combinación de elementos visuales para crear efectos visuales impresionantes.

Mezcla de Imágenes con `cv2.addWeighted()`

`cv2.addWeighted()` es una función de OpenCV que permite fusionar dos imágenes con el mismo tamaño, asignando a cada una un peso específico. Esto es útil para superponer imágenes de barcos en diferentes fondos con diferentes niveles de transparencia, creando así nuevas imágenes que contienen múltiples instancias de barcos en varios contextos.

El Algoritmo anterior 3 muestra el proceso de combinar dos imágenes utilizando OpenCV. Además, se presenta su resultado en la Ilustración 5.29.

Algorithm 3 Image Blending with OpenCV

-
- 1: Define the file paths for img1 and img2
 - 2: **if** img1 or img2 files do not exist **then**
 - 3: Display an error message and terminate the execution
 - 4: **end if**
 - 5: Read the images img1 and img2
 - 6: **if** loading of img1 or img2 fails **then**
 - 7: Display an error message and terminate the execution
 - 8: **end if**
 - 9: Resize img2 to the size of img1
 - 10: Blend img1 and img2 with a blending ratio (0.1 for img1 and 1 for img2)
 - 11: Get the current working directory
 - 12: Define the filename for saving the blended result
 - 13: Save the blended image in the current working directory
 - 14: Display a message with the path where the blended image was saved
 - 15: Create a resizable window for displaying the blended image
 - 16: Display the blended image on the screen
 - 17: Wait for a key press and then close the window
-

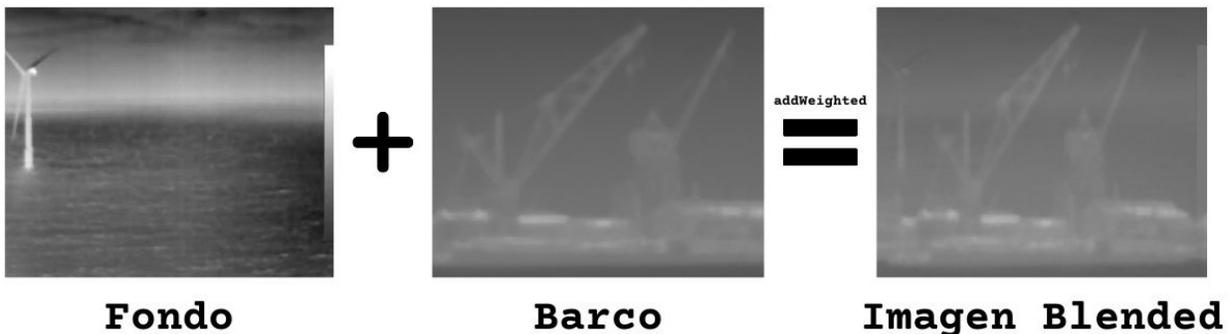


Ilustración 5.29: Resultado de fusionar dos imágenes del mismo tamaño con el algoritmo de Image Blending with OpenCV 3

La función `cv2.addWeighted()` es útil cuando se desea fusionar dos imágenes con el mismo tamaño y se desea tener control sobre la opacidad de cada imagen en la composición. Sin embargo, puede haber pérdida de nitidez en los detalles cuando las imágenes tienen diferentes resoluciones. Además, obliga a los barcos a estar en primer plano para cada uno de los fondos.

Expansión con `cv2.copyMakeBorder()`:

`cv2.copyMakeBorder()` expande una imagen existente creando un lienzo más grande alrededor de ella. Esta técnica es útil para colocar objetos, como barcos, en el centro del fondo, lo que permite un mayor control sobre su ubicación y evita que los objetos se corten o se pierdan en el borde de la imagen.

Sin embargo, esta técnica puede introducir píxeles que no coincidan perfectamente con el entorno circundante, lo que resulta en un ruido no deseado o en una apariencia poco natural en la imagen final. Encontrar siempre este ruido en las imágenes puede confundir al modelo y hacerle aprender que cada vez que existe ese ruido, puede haber un barco, cuando en realidad no es así. Esto puede llevar a una reducción en la precisión del modelo, ya que podría identificar erróneamente cada vez que analiza dicho ruido.

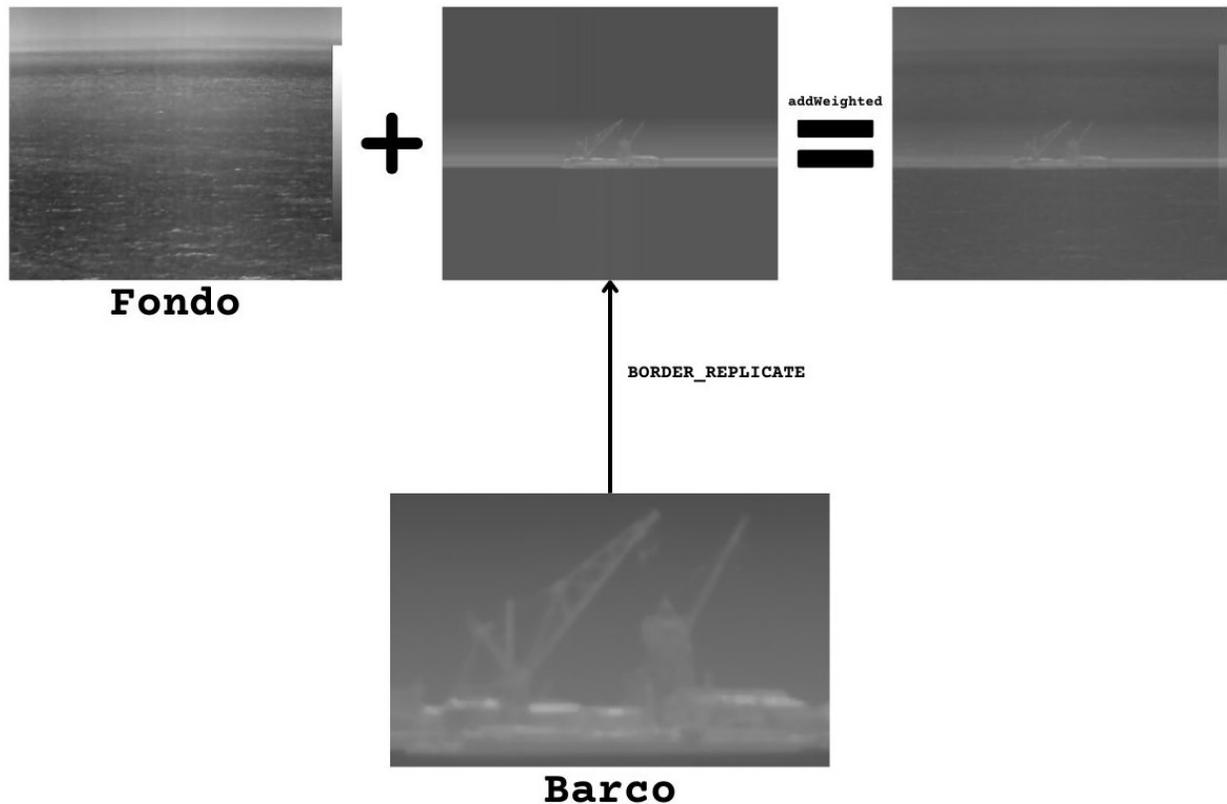


Ilustración 5.30: Composición de imagen usando `cv2.copyMakeBorder()` y `addWeighted`. El Fondo muestra la imagen de fondo utilizada, mientras que el Barco muestra el objeto que se superpone. La imagen resultante combina estos elementos usando `addWeighted` después de aplicar `cv2.copyMakeBorder()` para replicar el borde del barco

La Ilustración 5.30 muestra un ejemplo de esta técnica. Aunque OpenCV ofrece métodos efectivos para la composición de imágenes, es importante considerar las limitaciones de estas técnicas al aplicarlas en la creación de datasets para la detección de barcos. La pérdida de detalles, el ruido introducido y las inconsistencias en la iluminación y el fondo pueden afectar la calidad y la utilidad de los datos generados.

5.8.3. Libcom

LibCom es una herramienta que permite la creación de imágenes compuestas mediante la combinación de varias técnicas avanzadas de procesamiento de imágenes. [36]

Para generar una imagen compuesta, se necesitan tres elementos:

- **Imagen del Barco:** La imagen del objeto en primer plano. Esta imagen fue obtenida de IRShips [48], que contiene diferentes tipos y clases de embarcaciones, de distintos tamaños, ángulos y temperaturas.
- **Máscara del Barco:** Una máscara que define la región del objeto en primer plano.
- **Imagen de Fondo:** La imagen de fondo sobre la cual se integrará el objeto en primer plano. Las imágenes de fondo fueron obtenidas de MassMIND [35], cuyo dataset contiene imágenes térmicas de un contexto marítimo.

Para ello aprovecha varios componentes:

- **Image Harmonization:** Ajusta los colores y la iluminación del primer plano y el fondo para crear una imagen compuesta coherente. Se utilizan técnicas como la transferencia de color y la armonización basada en el aprendizaje profundo.
- **Image Matting:** Extrae con precisión el objeto del primer plano de su fondo original. Se emplean algoritmos de mateado de imágenes profundas que crean alfa, definiendo los niveles de transparencia de diferentes partes del objeto en primer plano.
- **Image Blending:** Una vez extraído y armonizado el primer plano, se mezcla con el nuevo fondo. Se utilizan métodos como la combinación Poisson o Gaussian, para suavizar los bordes e integrar el primer plano en el fondo.
- **Reflectance and Illumination Modeling:** Para mejorar aún más el realismo, Libcom modela las propiedades de reflectancia y las condiciones de iluminación de la escena. Esto incluye ajustar sombras, luces y otros efectos de iluminación para que coincidan con el entorno.

El uso de estas técnicas intenta garantizar un resultado final que sea visualmente convincente.

Problemas con la Máscara.

Debido a que la imagen del barco no podía ser PNG, no se podían eliminar los bordes de las imágenes para crear una mejor integración. Por lo tanto, se generó para cada imagen de un barco una máscara del mismo tamaño que dicha imagen. Esto provoca que los bordes no se integren correctamente, resultando en bordes poco realistas.

A pesar de intentar obtener los bordes de las imágenes mediante métodos como Canny, dichos bordes casi no se obtenían, por lo cual se optó por usar las máscaras mencionadas.

Método Gaussian.

Se utiliza un filtro gaussiano para suavizar la imagen, reduciendo el ruido y las discrepancias menores en los bordes. Esto ayuda a que la transición entre el primer plano y el fondo sea más suave.

La función gaussiana se aplica convolucionando la imagen con un núcleo gaussiano. Este proceso promedia los valores de los píxeles en una vecindad definida, dando mayor peso a los píxeles cercanos al centro del núcleo.

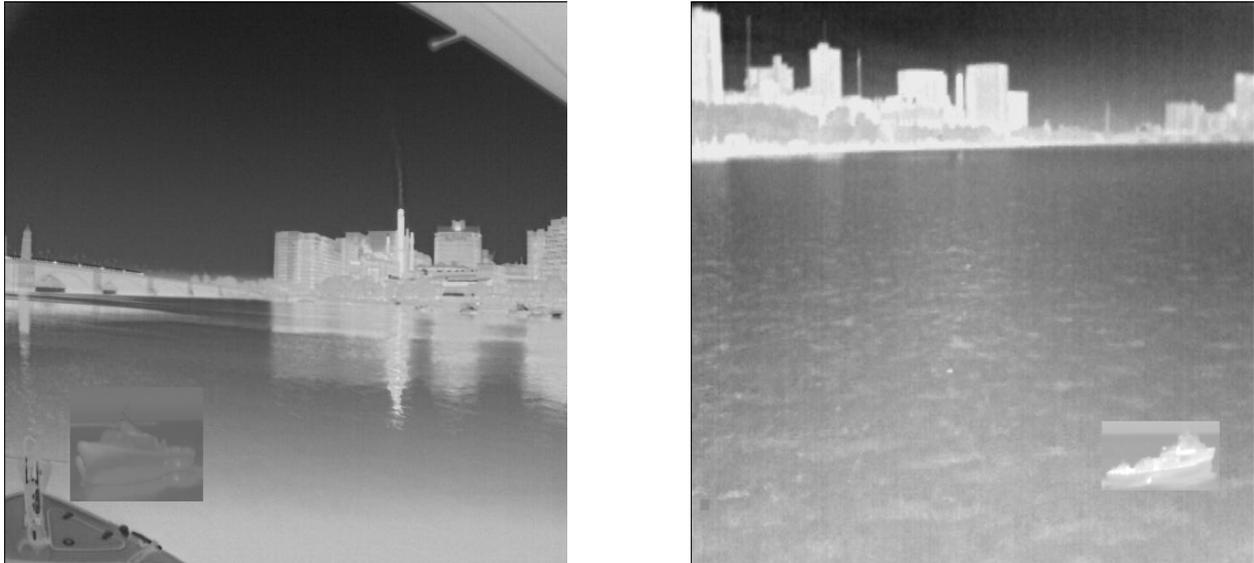


Ilustración 5.31: Imagen compuesta haciendo uso de gaussian. Imagen generada con [36]

Los bordes de las imágenes son notorios al intentar generar una imagen compuesta con Gaussian, lo que resulta en una integración poco realista, como se puede observar en la Ilustración 5.31. La calidad de los bordes es notablemente inferior a la obtenida con el método Poisson, independientemente de la imagen del barco y el fondo utilizados.

Método Poisson.

Este método se emplea para combinar de manera continua el objeto en primer plano con el fondo, preservando las texturas y la iluminación natural.

El algoritmo de mezcla de Poisson ajusta el campo de gradiente (cambios de intensidad) del área de mezcla para que coincida con el de la imagen de fondo. Esto se logra resolviendo una ecuación de Poisson con condiciones de contorno definidas por la imagen de origen.

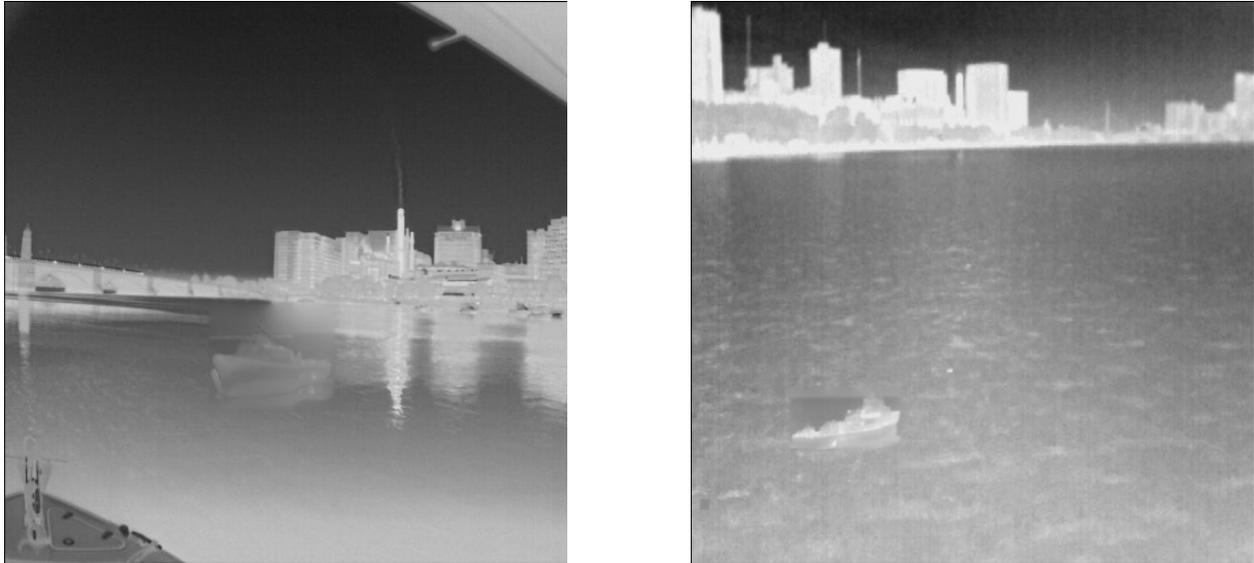


Ilustración 5.32: Imagen compuesta haciendo uso de Poisson. Imagen generada con [36]

Debido a que Poisson preserva mejor las texturas y la iluminación natural del fondo y del objeto en primer plano, los bordes generados son menos notorios en comparación con Gaussian. Aunque Poisson a veces produce resultados satisfactorios, logrando bordes más suaves y realistas dependiendo de la imagen del barco y el fondo, sigue sin obtener buenos resultados con frecuencia, como se muestra en la Ilustración 5.32.

Aunque Gaussian es más sencillo, tiende a hacer los bordes más notorios, mientras que el método Poisson, aunque más complejo, puede lograr una mejor integración de los bordes en ciertos escenarios específicos. Por lo tanto, la precisión en ambos casos es baja debido a la poca frecuencia de generar buenos resultados.

5.8.4. Superposición de Recortes

Como se mencionó anteriormente en la subsección Libcom 5.8.3, no se realizaba una buena composición de imágenes debido a la mala integración de los bordes de las imágenes de primer plano. Por lo tanto, se optó por reutilizar las imágenes de los barcos y fondo para replantearlo.

En este caso, se optó por eliminar manualmente el fondo de las imágenes, ya que, como se mencionó, no se detectaban los bordes de los barcos y no se podía hacer mediante métodos de detección de barcos.

Se generó un programa que coloca un barco sobre el fondo, sin salirse de la imagen, es decir, que la embarcación se vea completa y situada en $3/4$ de la parte superior de la imagen, ya que la parte superior no suele tener mar, a menos que la imagen de fondo sea todo el mar, y no tenga horizonte. Además, varía con el tamaño del barco.

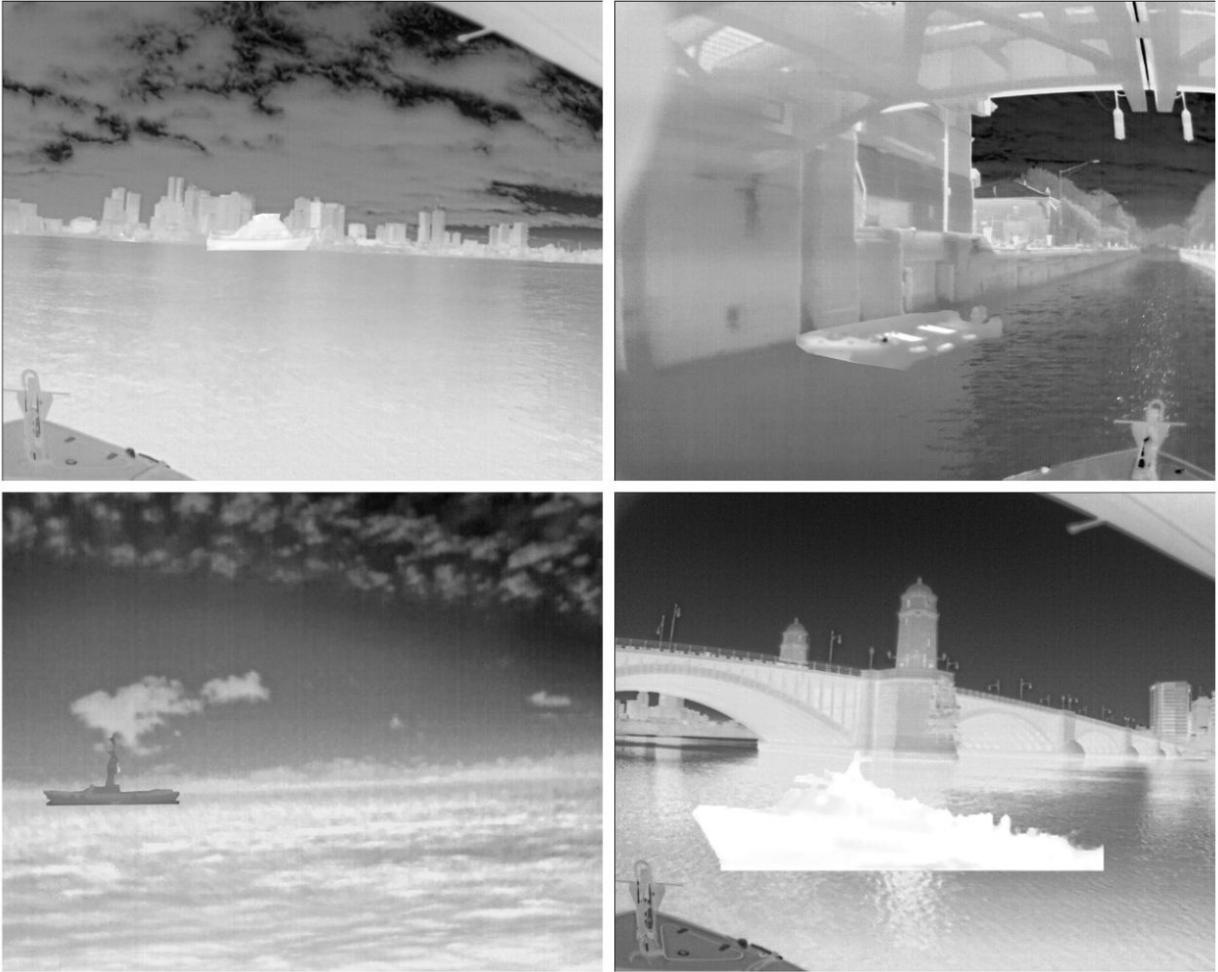


Ilustración 5.33: Resultados de composición de imágenes de barcos sobre fondos marítimos

Aunque la temperatura del barco no es la más adecuada para la integración con el entorno, como se analiza en la Ilustración 5.33, genera resultados buenos con más frecuencia que en las Ilustraciones anteriores 5.31 5.32. Es verdad que hay imágenes de barcos que, dependiendo del entorno o del tamaño del barco, se ven con mejor o peor calidad; no obstante, la frecuencia de buenos resultados permite generar imágenes para realizar un entrenamiento.

Implementación y Resultados

Tras analizar las imágenes generadas, se observó que, a pesar de la gran cantidad de imágenes de barcos, muchas estaban repetidas en diferentes tamaños o rotaciones. Por lo tanto, se añadieron 185 imágenes al dataset inicial, logrando un total de 500 imágenes para el entrenamiento.

Como se observa en la Ilustración 5.34, la matriz de confusión muestra el rendimiento del modelo en la detección de la clase *barco*. La tasa de verdaderos positivos (TP) es del 70 %, lo que indica que el modelo identifica correctamente los barcos el 70 % del tiempo. Sin embargo, la tasa de falsos negativos (FN) es del 30 %, lo que significa que el modelo no detecta barcos

presentes en el 30 % de las ocasiones. Esto es un área de preocupación, ya que una alta tasa de falsos negativos puede reducir significativamente la fiabilidad del modelo.

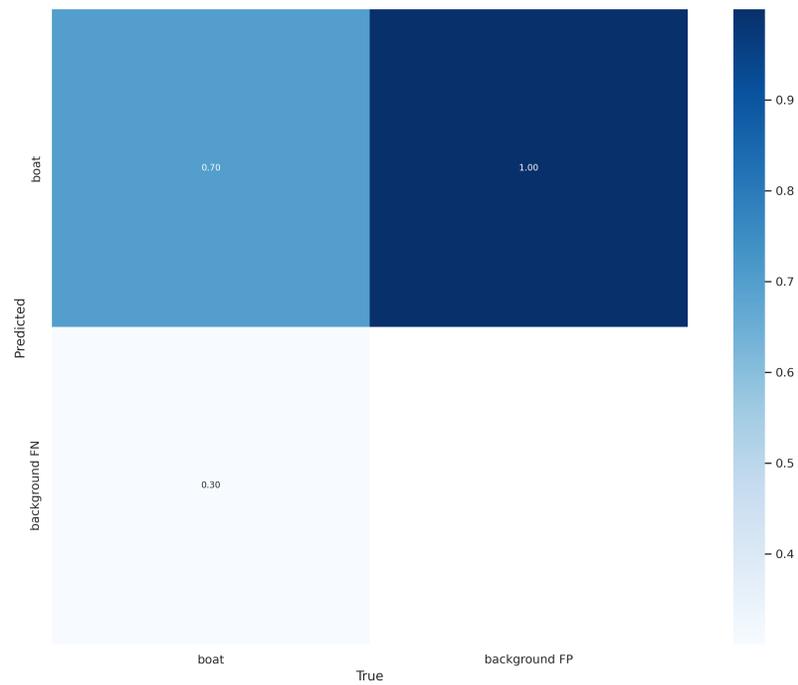


Ilustración 5.34: Matriz de confusión del entrenamiento del dataset inicial y Superposición de Recortes

Analizando los resultados de la Ilustración 5.35, tanto Box como Objectness presentan una disminución constante, con algunas fluctuaciones iniciales notables, especialmente en los valores de validación.

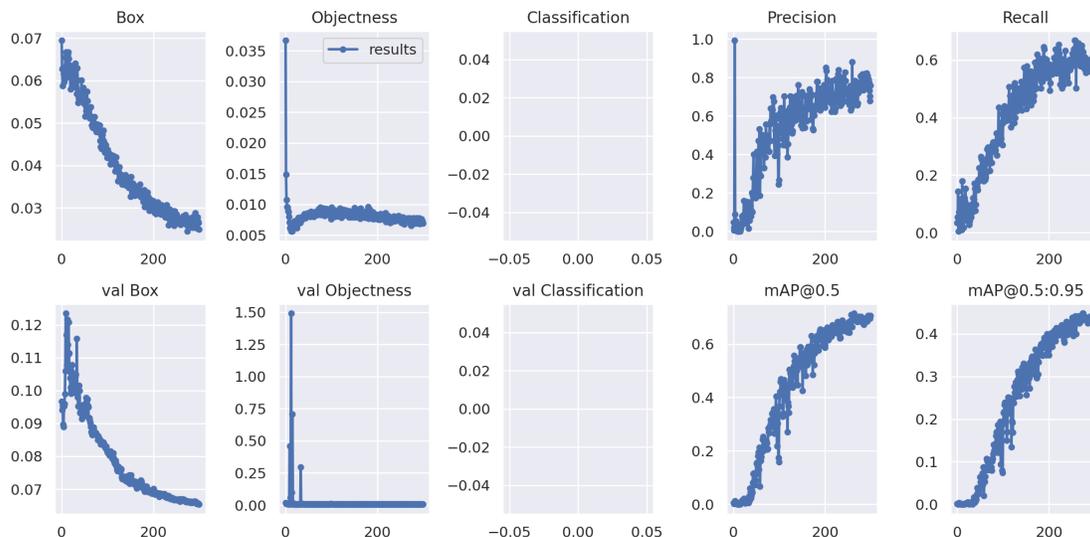


Ilustración 5.35: Resultados del entrenamiento del dataset inicial y Superposición de Recortes

La precisión y el recall aumentan con el tiempo, mostrando fluctuaciones significativas pero estabilizándose en valores altos, lo que demuestra una buena capacidad de detección y clasificación del modelo. En contraste, $mAP@0.5$ y $mAP@0.5:0.95$ tienen fluctuaciones más leves, indicando una precisión media más estable.

La técnica de composición de imágenes recortadas sobre un fondo puede introducir ruido que no coincide perfectamente con el entorno circundante, lo que podría llevar a una apariencia poco natural y confundir al modelo. Este ruido constante puede llevar al modelo a asociar erróneamente la presencia de ruido con la presencia de barcos, lo que no es deseable y puede afectar negativamente a la precisión del modelo.

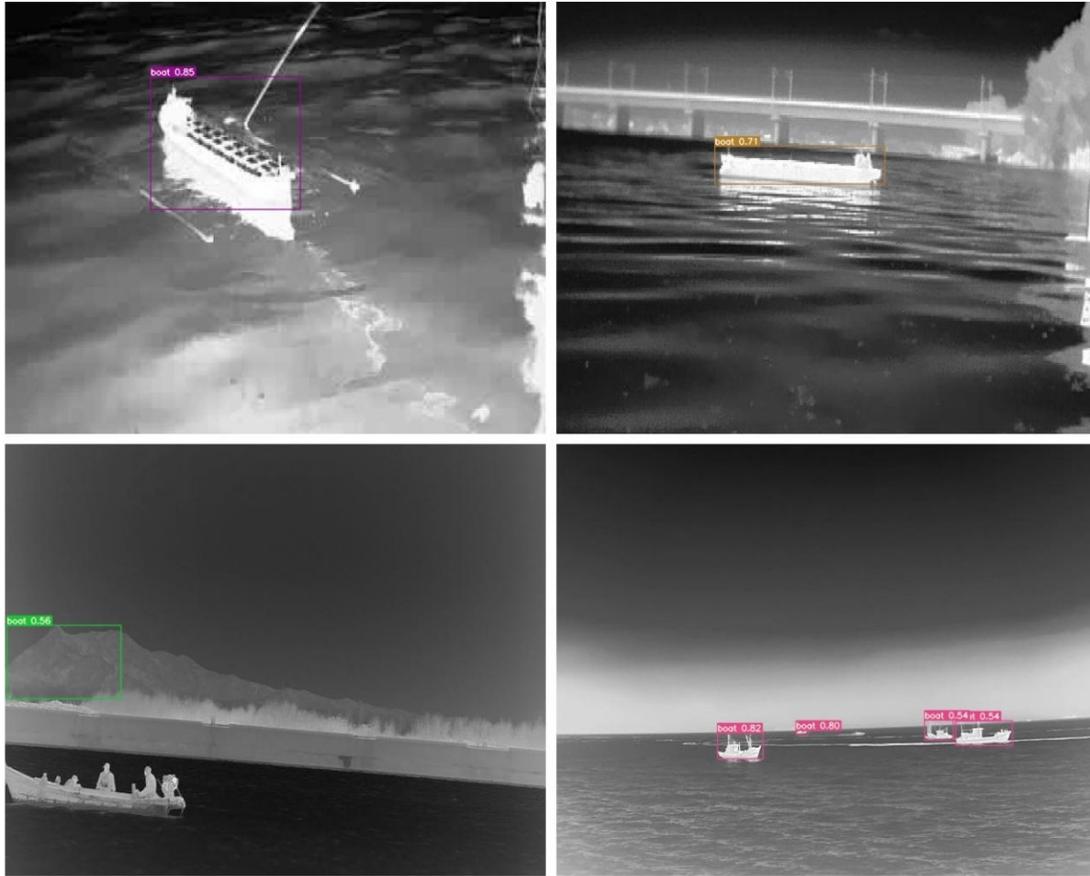


Ilustración 5.36: Inferencia del entrenamiento del dataset inicial y Superposición de Recortes

Si se analiza los resultados de la inferencia de la Ilustración 5.36, todavía enfrenta dificultades con la detección de imágenes afectadas por la calima. Aunque las falsas detecciones de objetos como barcos continúan, son menos frecuentes que en los conjuntos de datos originales. Además, las obstrucciones siguen presentando un desafío persistente.

Los resultados de las métricas indican que el modelo está mejorando continuamente en términos de precisión y recall, así como en la confianza en la detección de objetos. Aunque hay fluctuaciones iniciales en algunas métricas, estas se estabilizan en valores aceptables, sugiriendo una buena capacidad del modelo para detectar barcos en cámaras térmicas. No obstante, se identificó un problema con una tasa de falsos negativos del 30 %, lo cual puede reducir la fiabilidad del modelo en situaciones críticas.

5.9. Sprint 8: Generación de Datos mediante Inteligencia Artificial

5.9.1. Aplicaciones Online

Las herramientas en línea basadas en inteligencia artificial han revolucionado la manera en que generamos y manipulamos imágenes. Estas aplicaciones permiten la creación de imágenes de alta calidad mediante algoritmos avanzados, ofreciendo una amplia gama de estilos y funcionalidades. A continuación, algunas de las aplicaciones más populares en este campo, analizando su rendimiento para este ámbito específico.

ChatGPT4

ChatGPT-4¹ es un modelo de inteligencia artificial desarrollado por OpenAI, basado en la arquitectura GPT-4. Este modelo es notable por sus capacidades avanzadas de generación de texto e imágenes, permitiéndole crear representaciones visuales detalladas a partir de descripciones textuales. Esta característica es especialmente útil en campos como el diseño gráfico, la publicidad y la educación [5, 31].



Ilustración 5.37: Resultados de imágenes generadas con ChatGPT.

En las pruebas realizadas, se observó que los resultados generados por ChatGPT-4 eran en blanco y negro, por lo que no se asemejaba a una imagen térmica. Esto es evidente en los ejemplos mostrados en la Ilustración 5.37.

Runway ML

Runway ML² es una plataforma diseñada para facilitar la experimentación y gestión de modelos de aprendizaje automático. La plataforma proporciona una interfaz amigable que permite a los usuarios explorar, ejecutar y gestionar experimentos con modelos de aprendizaje automático. Esto es particularmente útil para artistas, diseñadores y otros profesionales

¹<https://chatgpt.com/>

²<https://runwayml.com/ai-tools/text-to-image/>

creativos que desean incorporar técnicas avanzadas de IA en sus proyectos sin tener que lidiar con la complejidad técnica subyacente [43].



Ilustración 5.38: Resultados de imágenes con Runway ML

Aunque Runway ML es una herramienta versátil y accesible para la experimentación con modelos de IA, no está específicamente optimizada para la generación de imágenes térmicas. Las imágenes generadas por Runway ML que se muestran en la Ilustración 5.38, pueden ser en blanco y negro o a color, pero no capturan las características distintivas de las imágenes térmicas, como las variaciones de intensidad que representan diferentes temperaturas.

Leonardo AI

Leonardo AI³ es una avanzada herramienta de creación de imágenes que utiliza inteligencia artificial generativa para producir contenido visual a partir de indicaciones textuales. Leonardo AI permite a los usuarios crear imágenes simplemente ingresando descripciones textuales. La plataforma convierte estas descripciones en representaciones visuales precisas y estilizadas. Los usuarios pueden ajustar el estilo y la creatividad de las imágenes generadas para obtener resultados personalizados. [28] [26]



Ilustración 5.39: Resultados de imágenes generadas con LeonardoAI

Los resultados de Leonardo AI en la Ilustración 5.39 muestran que, aunque también genera imágenes en blanco y negro, es más propenso a crear imágenes a color, lo que las aleja más de

³<https://app.leonardo.ai/>

las imágenes térmicas. A pesar de esto, su enfoque en la precisión técnica lo hace adecuado para aplicaciones que requieren imágenes detalladas y exactas.

Artbreeder

Artbreeder⁴ es una herramienta de creatividad asistida por inteligencia artificial que permite a los usuarios generar y modificar imágenes, mediante una interfaz intuitiva. Utiliza técnicas avanzadas de aprendizaje profundo, específicamente redes generativas adversariales (GANs), para facilitar la creación y combinación de imágenes de manera fluida y creativa. [25] [14]



Ilustración 5.40: Resultados de imágenes generadas con Artbreeder

En comparación con otras herramientas de generación de imágenes, Artbreeder se distingue por su capacidad de interpolación y combinación de rasgos. Mientras que herramientas como ChatGPT4 y Runway ML generan imágenes principalmente en blanco y negro, Artbreeder tiende a generar imágenes a color, lo que lo diferencia significativamente de las imágenes térmicas, que son el foco principal de algunas aplicaciones.

A pesar de usar diferentes *prompts* y ajustarse al entorno, se puede analizar en la Ilustración 5.40, los numerosos errores en la representación de las escalas de colores, con errores que van desde imágenes a color, hasta imágenes en blanco y negro. Por lo tanto, los colores asignados al entorno y al barco se alejan de las imágenes térmicas, lo que no lo hace confiable para un entrenamiento del modelo. Aunque los resultados generados son más rápidos que entrenar una GAN como en el caso de CycleGAN, por ejemplo, sus resultados son peores.

⁴<https://www.artbreeder.com/create/prompter?fromCreate=1>

5.9.2. GLIDE (Guided Language-to-Image Diffusion for Generation and Editing)

GLIDE es un modelo de difusión guiado por texto que genera imágenes realistas y coherentes a partir de descripciones textuales. Este modelo no solo crea imágenes desde cero, sino que también permite la edición de imágenes existentes basándose en nuevas instrucciones textuales. [34]



Ilustración 5.41: Resultados de barcos generadas mediante GLIDE [34]

A pesar de que las imágenes generadas por GLIDE, mostradas en la Ilustración 5.41, son realistas y de alta resolución, no se asemejan a una representación de imagen térmica, sino a imágenes en blanco y negro. Por lo tanto, se puede analizar que es una indicación compleja o no se encuentra reflejada en los datos de entrenamiento.

5.9.3. Text-guided image-to-image generation

El proceso involucró utilizar las imágenes descartadas en el dataset inicial por ser demasiado similares a las ya presentes en dicho dataset y transformarlas en nuevas versiones con

características específicas, como añadir barcos o modificar los existentes en las imágenes.

Para lograr una transformación efectiva, se realizaron pruebas con distintos valores de *strength* y *guidance-scale*. Cuando se usaron valores altos de *strength* y *guidance-scale*, las imágenes resultantes se alejaron considerablemente de la imagen original, produciendo colores y detalles que no parecían realistas. Con valores bajos, las imágenes generadas eran muy similares a las originales, con cambios mínimos que no cumplían completamente con las especificaciones del prompt. Se obtuvieron los mejores resultados con valores intermedios, especialmente cuando se inclinaban hacia la parte inferior del rango. Estos valores equilibraban la preservación de la estructura original de la imagen con la incorporación de los detalles especificados en el prompt.

A continuación se muestra un resultado obtenido utilizando este método con diferentes valores de *strength* y *guidance-scale*. En la Ilustración 5.42, se presentan varios ejemplos del mismo método aplicado a la misma imagen, con resultados no deseados.

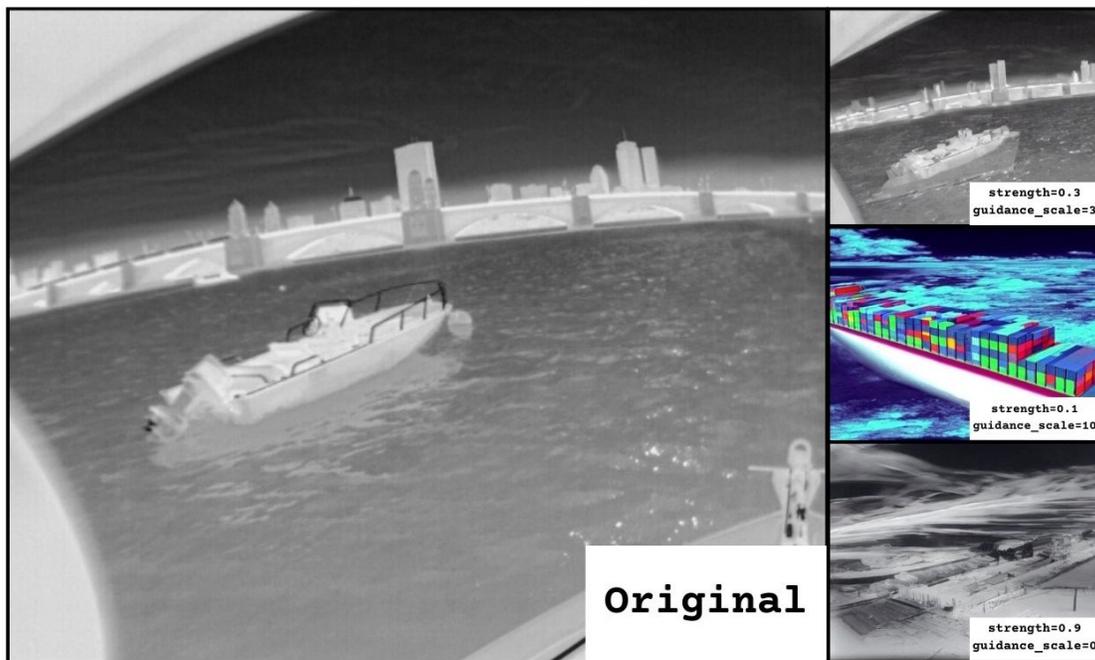


Ilustración 5.42: Resultados generados con Text-guided image-to-image generation [12] con distintos valores de *strength* y *guidance scale*. De arriba a abajo, se observa: un resultado similar a la imagen original, un resultado a color que no corresponde a una cámara térmica, y un resultado que no se asemeja al prompt proporcionado.

Al ajustar cuidadosamente los parámetros de *strength* y *guidance-scale*, se pudieron obtener imágenes que eran visualmente coherentes y realistas, cumpliendo con las especificaciones de los *prompts* textuales sin perder la esencia de las imágenes originales. La Ilustración 5.43 muestra ejemplos de estos ajustes, demostrando cómo esta metodología permitió enriquecer el conjunto de datos, ofreciendo una herramienta valiosa para la mejora y expansión de conjuntos de datos visuales.

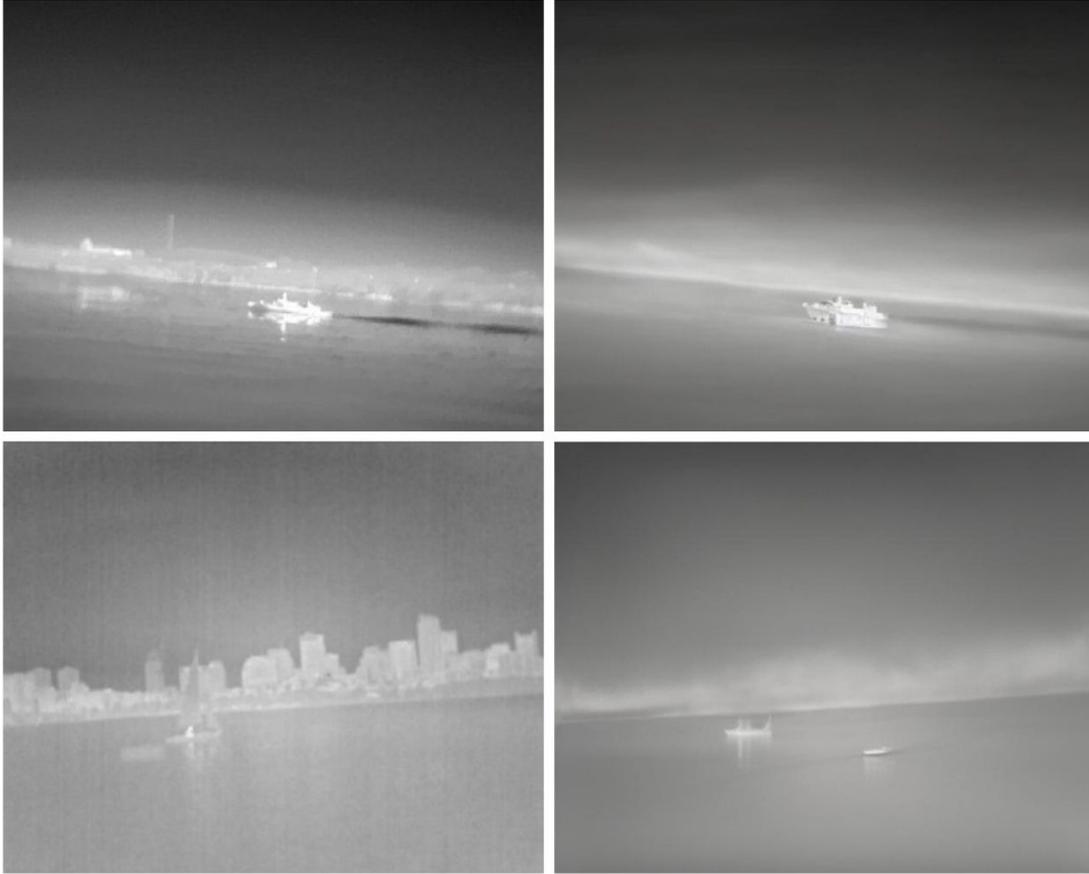


Ilustración 5.43: Imágenes de barcos generadas mediante Text-guided image-to-image generation [12]. Las imágenes situadas la izquierda son las originales, y las de la derecha el resultado del proceso

Tras la generación de diversas imágenes y su posterior incorporación al dataset inicial, el conjunto de datos ahora comprende un total de 515 imágenes. En la Ilustración 5.44, se analiza que el modelo identifica correctamente los barcos en el 74 % de los casos (verdaderos positivos). Sin embargo, el 26 % de falsos negativos, donde el modelo no detecta los barcos presentes, es un área de preocupación significativa. Esto indica que, aunque el modelo tiene una alta precisión, su capacidad para detectar todos los barcos aún necesita mejorar.

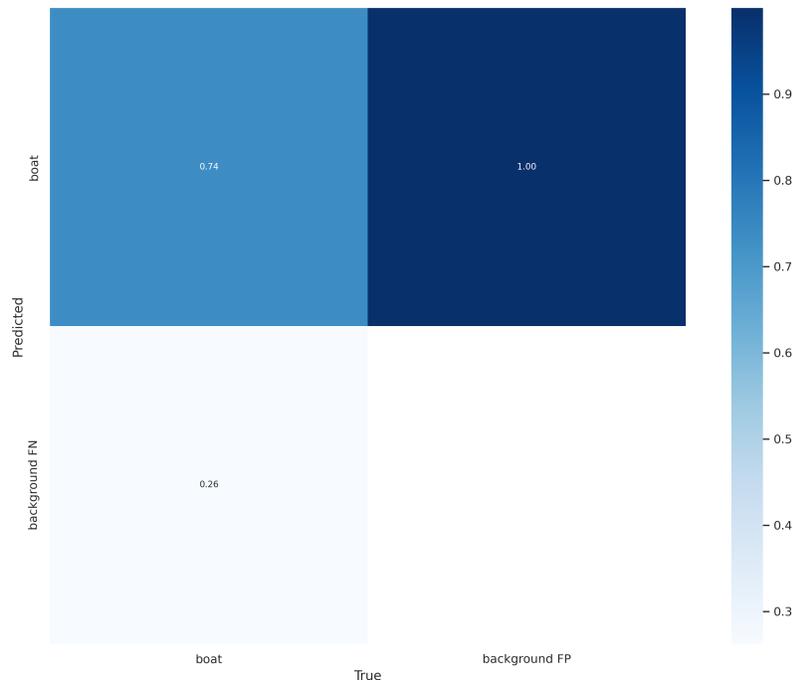


Ilustración 5.44: Matriz de confusión del entrenamiento del conjunto de datos inicial y generación de imágenes guiada por texto

Como se analiza en la Ilustración 5.45, inicialmente hay valores altos al principio del val Box, lo cual podría indicar una incertidumbre inicial del modelo en la detección de barcos en los datos de validación. Esto se estabiliza rápidamente, lo que sugiere que el modelo mejora su capacidad de distinguir entre barcos y fondo a lo largo del entrenamiento.

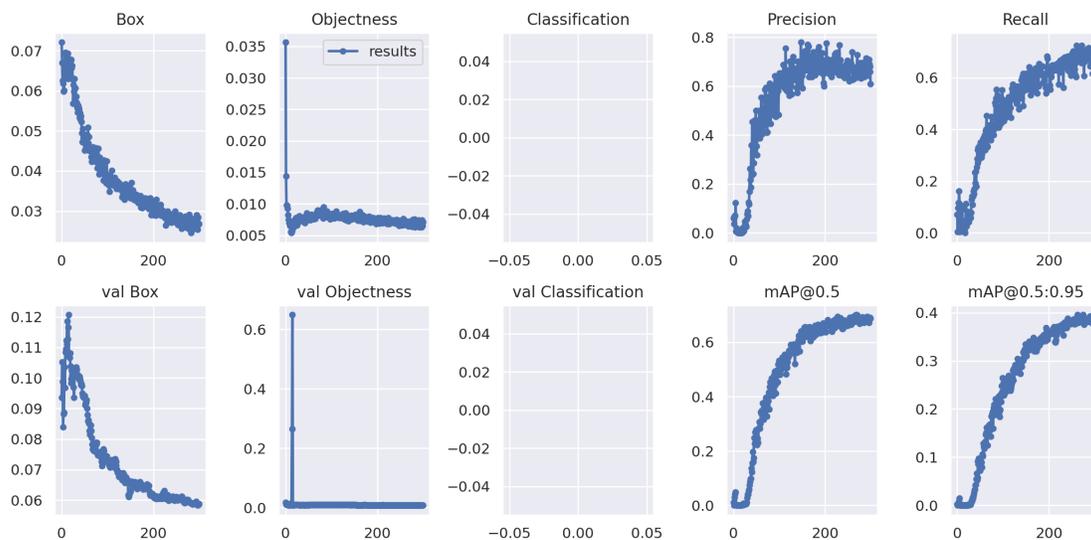


Ilustración 5.45: Resultados del entrenamiento del conjunto de datos inicial y generación de imágenes guiada por texto

Las métricas de precisión y recuperación muestran una mejora continua, con la precisión alcanzando valores altos, lo que implica que el modelo es muy preciso en sus predicciones, aunque con fluctuaciones significativas en ambos casos.

En general, el modelo de detección de barcos muestra un buen rendimiento en términos de precisión y recall. La alta precisión asegura que el modelo no etiqueta incorrectamente el fondo como barcos, y la mejora continua en las métricas clave sugiere que el modelo está aprendiendo de manera efectiva. Sin embargo, el 26% de falsos negativos es un área de preocupación significativa, ya que indica que el modelo omite algunos barcos, lo cual podría ser crítico dependiendo de la aplicación. La fluctuación inicial en la pérdida de objetividad en la validación destaca la necesidad de mejorar la estabilidad del modelo en datos no vistos.

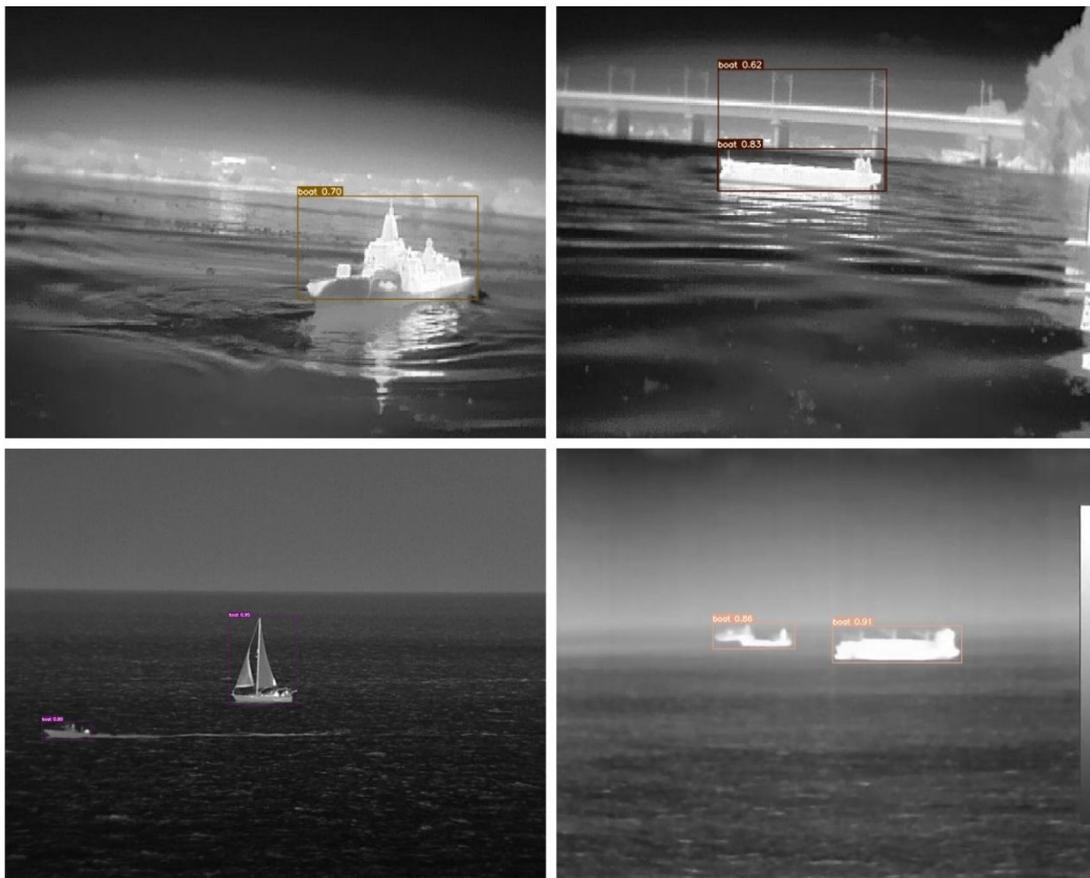


Ilustración 5.46: Inferencia del entrenamiento del conjunto de datos inicial y Generación de imagen a imagen guiada por texto

A pesar de estas áreas de mejora, el modelo se considera eficaz para la detección de barcos debido a su alta precisión y su capacidad de mejorar con el entrenamiento. Además, aunque el modelo muestra un buen desempeño, todavía presenta fallos en situaciones de oclusión y ocasionalmente genera detecciones falsas en estructuras de fondo que no son barcos, como se puede observar en la Ilustración 5.46.

5.10. Sprint 9: Dataset Combinado (Superposición de Recortes y Text-guided image-to-image generation)

Este dataset contiene un total de 701 imágenes. En la Ilustración 5.47, la matriz de confusión muestra que la precisión para la clase barcos es de 0.76, lo que significa que el modelo predice correctamente el 76 % de los botes presentes en las imágenes. Por otro lado, la proporción de falsos negativos es de 0.24, indicando que el 24 % de los botes no fueron detectados por el modelo. Este análisis subraya la efectividad del modelo en la detección de objetos, aunque hay margen de mejora en la reducción de falsos negativos.

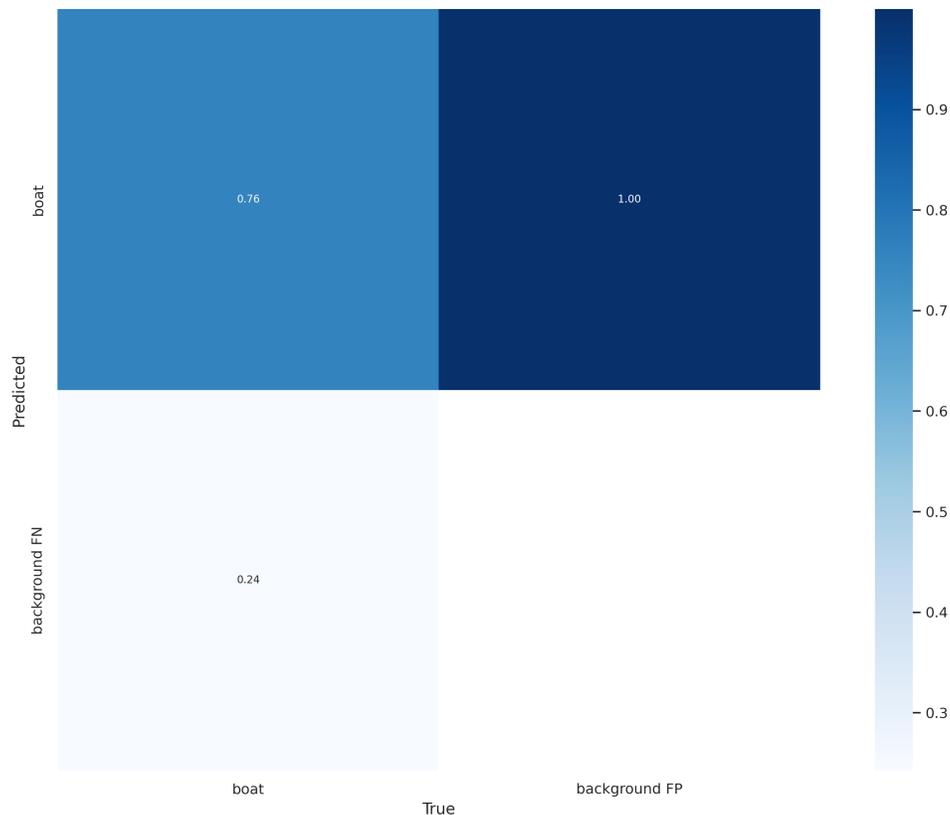


Ilustración 5.47: Matriz de confusión del entrenamiento del dataset inicial y Generación de imagen a imagen guiada por texto y Superposición de Recortes

Además, se han observado mejores valores en la matriz de confusión al evaluar el modelo en el dataset combinado, en comparación con los resultados obtenidos con los datasets individuales.

La Ilustración 5.48, muestra la evolución de las pérdidas durante el entrenamiento y la validación. La métrica de pérdida disminuye consistentemente hasta alcanzar aproximadamente 0.02 al final del entrenamiento, indicando una mejora en la precisión de la localización

de los objetos en las imágenes. La pérdida de Objectness también muestra una tendencia descendente, finalizando cerca de 0.005, lo que sugiere que el modelo mejora su capacidad para distinguir entre objetos y el fondo.

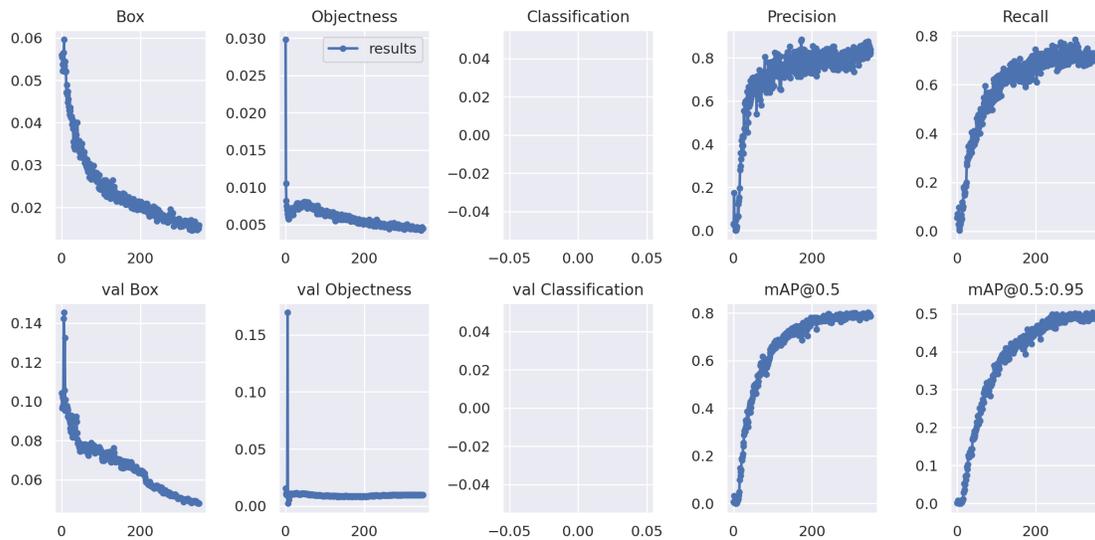


Ilustración 5.48: Resultados del entrenamiento del dataset inicial y Text-guided image-to-image generation y Superposición de Recortes

Los gráficos de precisión y recall muestran que ambos aumentan de manera constante y rápida con cierta dispersión, la cual no es significativa. La métrica mAP@0.5, que mide la precisión media promedio a un umbral de 0.5, se estabiliza cerca de 0.786. Además, la métrica mAP@0.5:0.95, que es más estricta al evaluar la precisión promedio en diferentes umbrales de IOU, también muestra una tendencia ascendente.

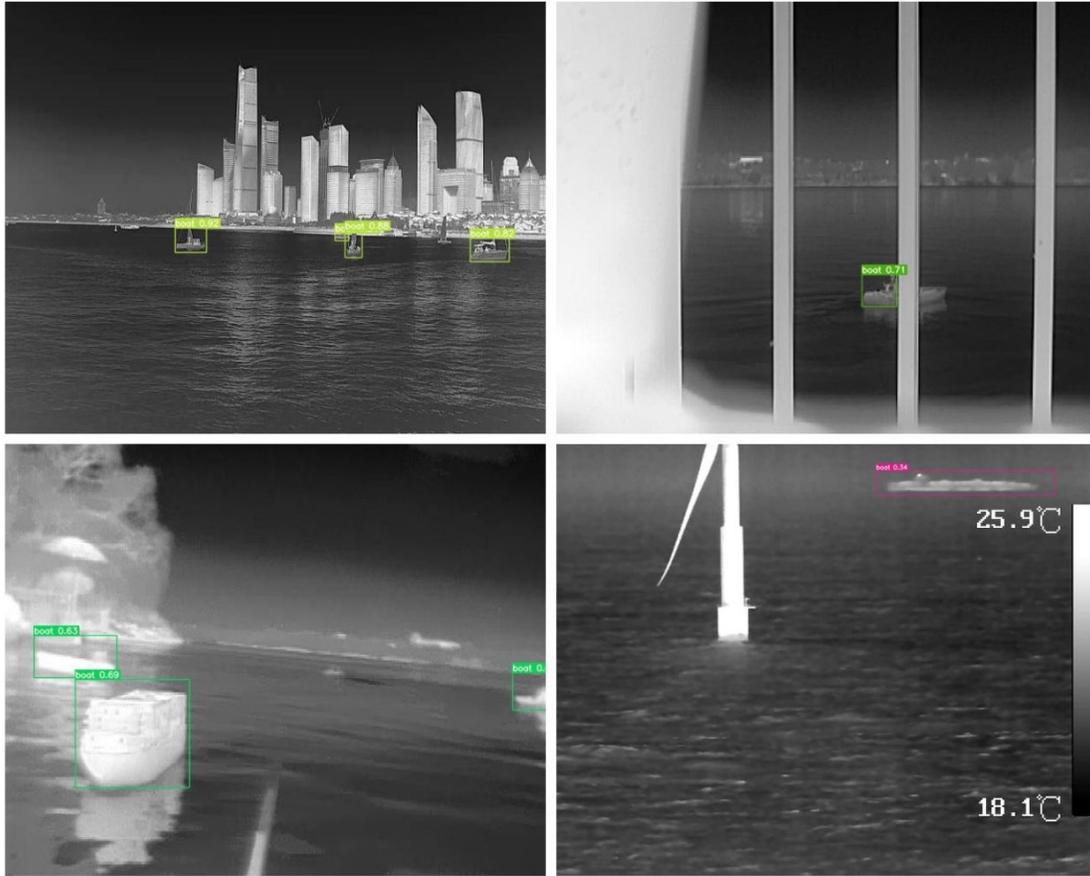


Ilustración 5.49: Inferencia del entrenamiento del dataset inicial y Generación de imagen a imagen guiada por texto y Superposición de Recortes

Los resultados del entrenamiento del modelo YOLOv7 con el dataset combinado de Superposición y Text-guided Image-to-Image Generation son muy prometedores. El modelo muestra una alta precisión y recall. Además, el análisis de la matriz de confusión confirma que el modelo es eficaz en la detección de objetos, aunque puede beneficiarse de mejoras adicionales para reducir la tasa de falsos negativos. Este rendimiento demuestra que la combinación de superposición de imágenes y generación guiada por texto proporciona un conjunto de datos robusto y efectivo para el entrenamiento de modelos de detección de objetos.

Modificación del Dataset

Es importante señalar que antes de la creación del conjunto de datos inicial, se realizó un proceso de limpieza del mismo. Al principio, el dataset contenía imágenes de baja calidad que empeoraban los resultados debido al pequeño tamaño del conjunto de datos. Ahora, con un conjunto de datos más grande, se han reincorporado dichas imágenes, incrementando el tamaño del dataset de 701 imágenes a 765.

En la Ilustración 5.50, la matriz de confusión indica que el modelo tiene un buen desempeño en la identificación de la categoría barco. Los resultados muestran que el modelo predijo correctamente el 73 % de los casos de bote, mientras que el 27 % fueron falsos negativos, es

decir, el modelo no logró identificar las embarcaciones en estos casos.

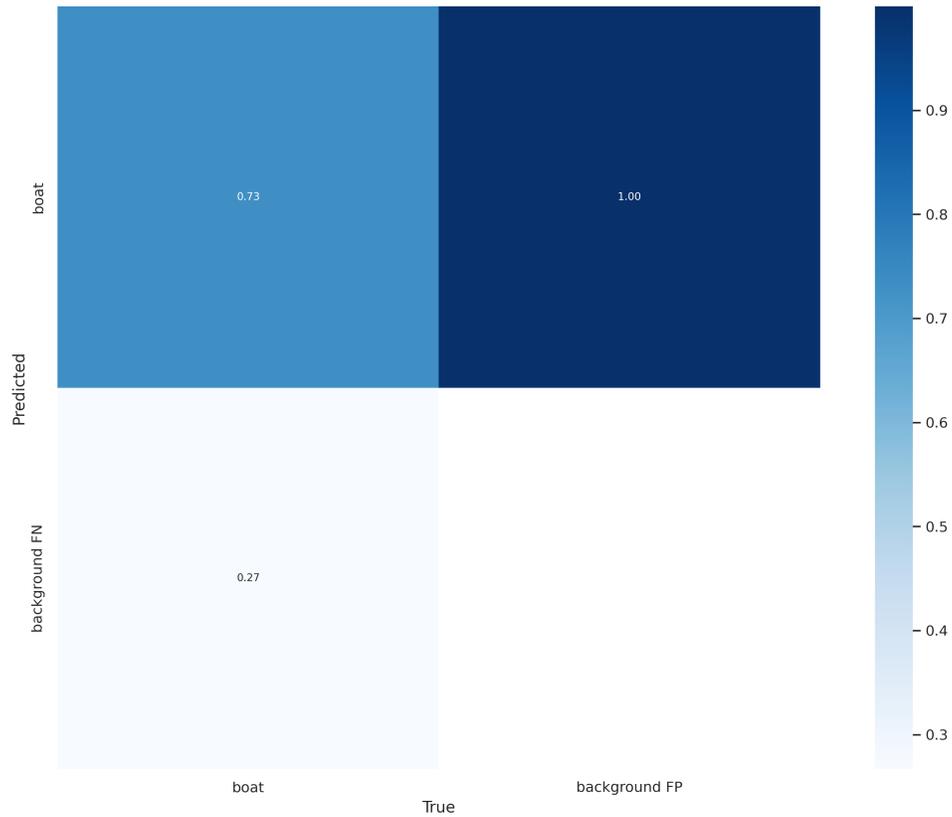


Ilustración 5.50: Matriz de confusión del entrenamiento del dataset inicial y Text-guided image-to-image generation y Superposición de Recortes e imágenes añadidas

En la Ilustración 5.51 se observa una disminución general en la pérdida de caja (Box) y objetividad (Objectness), lo que sugiere una mejora en la capacidad del modelo para predecir las posiciones de los objetos y distinguir objetos relevantes del fondo. Sin embargo, las fluctuaciones iniciales en la objetividad sugieren interferencias que podrían estar dificultando la estabilidad del aprendizaje.

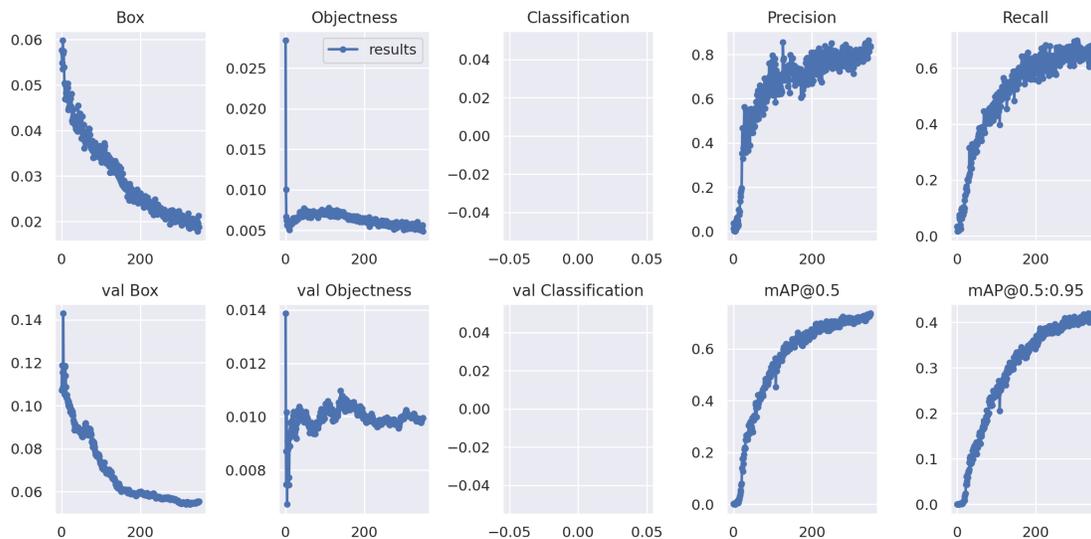


Ilustración 5.51: Resultados del entrenamiento del dataset inicial y la generación de imágenes de texto a imagen y superposición de recortes e imágenes añadidas

Las métricas de precisión y recuperación muestran una tendencia positiva, con incrementos notables a lo largo de las épocas, indicando mejoras en la capacidad del modelo para identificar y detectar barcos; sin embargo, la dispersión durante el entrenamiento es mayor que sin estas nuevas imágenes. Las métricas mAP@0.5 y mAP@0.5:0.95 también presentan una mejora constante, señalando una buena generalización en el conjunto de validación, a pesar de las interferencias.

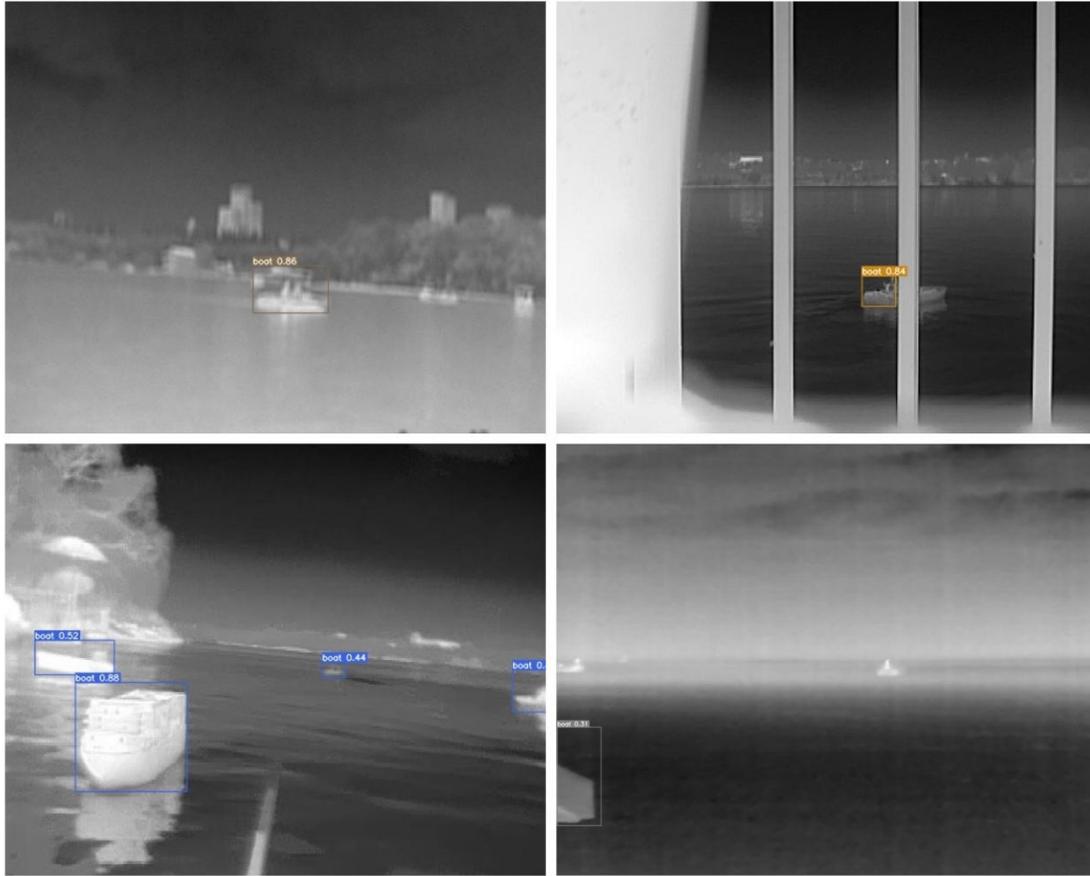


Ilustración 5.52: Inferencia del entrenamiento del dataset inicial y generación de imágenes guiadas por texto e imagen a imagen y superposición de recortes e imágenes adicionales

5.11. Sprint 10: Análisis Comparativo de Datasets

La Tabla 5.2 muestra el rendimiento de diferentes datasets utilizados para validar el modelo YOLOv7 en diversas condiciones. Los valores reflejan la cantidad de detecciones realizadas bajo cada condición específica. Dependiendo de la condición, es deseable valores altos o bajos, ya que reflejan una mayor cantidad de detecciones correctas o erróneas. Este análisis es esencial para garantizar la fiabilidad y eficiencia del sistema de detección, especialmente en condiciones variadas que son representativas del entorno real donde se implementará.

Para la evaluación, se consideraron varios datasets: Inicial con distribución Aleatoria, Inicial con distribución por Histogramas, CycleGAN, Superposición de Recortes, *Text-guided image-to-image generation*, Combinado y Combinado Modificado.

Tabla 5.2: Resultados de la Validación de YOLO con Diferentes Datasets

Condición \ Dataset	COCO	Inicial Aleatorio	Inicial Histograma	CycleGAN	Supersposición de Recortes	Text-guided image-to-image generation	Combinado	Combinado Modificado
Primer Plano	Muy baja	Muy baja	Muy baja	Muy baja	Baja	Muy baja	Baja	Baja
Oclusión	Muy baja	Moderada	Alta	Baja	Moderada	Moderada	Alta	Alta
Calima	Muy baja	Moderada	Baja	Moderada	Baja	Moderada	Moderada	Moderada
Parcial	Muy baja	Alta	Alta	Alta	Alta	Alta	Muy Alta	Alta
Objeto no barco parcial	Muy baja	Muy baja	Baja	Muy baja	Muy baja	Baja	Muy baja	Muy baja
Lejos	Baja	Muy Alta	Muy Alta	Alta	Alta	Muy Alta	Muy Alta	Alta
Cerca	Muy baja	Baja	Muy baja	Baja	Baja	Baja	Alta	Baja
Borroso	Muy baja	Muy baja	Muy baja	Muy baja	Muy baja	Muy baja	Muy baja	Muy baja
Falso positivo	Moderada	Baja	Moderada	Baja	Baja	Moderada	Baja	Baja

En general, se observa que algunos datasets tienen fortalezas claras dependiendo de la condición específica:

Primer Plano

Esta condición evalúa la capacidad del modelo para detectar barcos que ocupan la mayor parte de la imagen. A pesar de que la detección debe ser fácil, la mayoría de los datasets muestran un rendimiento muy bajo en esta área, indicando una detección mínima de barcos en primer plano. Aunque los datasets Superposición de Recortes, Combinado y Combinado Modificado muestran una mejora, su rendimiento aún es insuficiente para una detección confiable. Además, las cajas delimitadoras pueden no encerrar completamente al barco, como se muestra en la Ilustración 5.53.

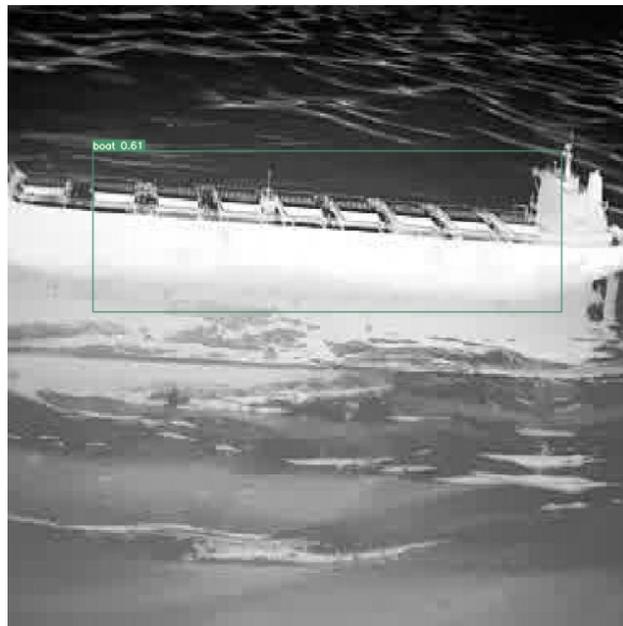


Ilustración 5.53: Barco detectado en primer plano con las cajas delimitadoras no ajustadas correctamente con el dataset Combinado Modificado

Cerca

Aquí se analiza la detección de barcos en planos medios, donde el dataset Combinado destaca con un alto rendimiento, capturando la mayoría de los barcos en esta condición específica.

Lejos

En los casos donde los barcos están situados a gran distancia, el rendimiento varía significativamente entre los datasets evaluados. COCO muestra un rendimiento bajo en esta condición, indicando dificultades para detectar barcos distantes. Por otro lado, datasets como CycleGAN, Superposición de Recortes y Combinado Modificado muestran un alto rendimiento, lo cual es muy favorable. Además, Inicial Aleatorio, Inicial Histograma, *Text-guided image-to-image generation* y Combinado muestran un rendimiento muy alto, lo cual es ideal para la detección de barcos a largas distancias.

Es importante considerar que a medida que los barcos se encuentran más lejos, su tamaño en la imagen disminuye considerablemente. Esto puede hacer que sea difícil distinguirlos claramente, incluso para un observador humano, ya que pueden confundirse con plataformas marítimas o boyas debido a la pérdida de detalles y la reducción de la forma distintiva del barco.

Oclusión

Esta condición evalúa la capacidad del modelo para detectar barcos parcialmente ocultos por otros barcos. COCO muestra un rendimiento insuficiente para detectar barcos ocluidos, no logrando detectar ninguno en estas condiciones. Por otro lado, datasets como Inicial Aleatorio, Superposición de Recortes y *Text-guided image-to-image generation* muestran un rendimiento moderado, detectando aproximadamente la mitad de los casos o únicamente el barco que está menos ocluido y más cercano a la cámara. En contraste, Inicial Histograma, Combinado y Combinado Modificado muestran un rendimiento más robusto, logrando analizar correctamente la mayoría de los barcos ocluidos. Es importante notar que algunos datasets pueden agrupar múltiples barcos ocluidos en un único objeto, tal y como se muestra en la Ilustración 5.54, lo cual afecta la precisión de la detección en estos casos específicos.



Ilustración 5.54: Barcos ocluidos que son detectados como un único barco con el dataset Text-guided image-to-image generation

Calima

Aunque las cámaras térmicas mejoran considerablemente la visibilidad de los barcos en condiciones de calima en comparación con las cámaras ópticas, los modelos aún enfrentan dificultades para detectar los barcos en estas imágenes. En la mayoría de los casos, los barcos no son detectados o la confianza en la detección es muy baja. Para los datasets Inicial Aleatorio, CycleGAN, *Text-guided image-to-image generation*, Combinado y Combinado Modificado, aproximadamente la mitad de los barcos son detectados en estas condiciones. Sin embargo, como muestra la Ilustración 5.55, los barcos detectados suelen tener una alta confianza en la detección, generalmente por encima del 0.90.



Ilustración 5.55: Barco detectado en la zona del Mirador con situación de calima con el dataset Combinado

Parcial

Esta condición evalúa la capacidad del modelo para detectar barcos que están parcialmente visibles en la imagen, ya sea porque están parcialmente ocultos por otros objetos en el entorno o porque están recortados por los márgenes de la imagen. La mayoría de los datasets muestran un rendimiento alto en esta área, detectando correctamente los barcos parcialmente visibles.

Objeto no barco parcial

En el siguiente caso se analiza la detección de objetos que no son barcos en entornos donde pueden estar parcialmente visibles, como muros en un mirador (consultar Ilustración 5.56). Al analizar los resultados, se observa que la mayoría de los datasets muestran un rendimiento muy bajo en esta área, lo cual es positivo ya que indica una baja incidencia de falsos positivos para estos casos específicos. Sin embargo, los datasets Inicial Histograma y Superposición de Recortes muestran un rendimiento mayor, lo que sugiere una ligera tendencia a generar más falsos positivos en comparación con otros datasets.



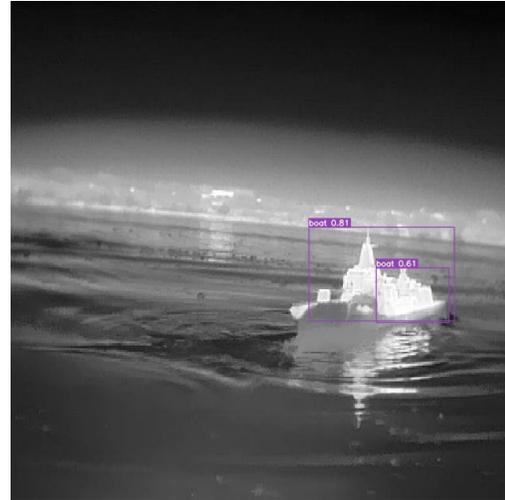
Ilustración 5.56: Muro detectado como barco zona del Mirador con el dataset Text-guided image-to-image generation

Falso positivo

Excluyendo los casos de *Objeto no barco parcial*, los falsos positivos, en su mayoría, se detectan como boyas altas. Los datasets COCO, Inicial Histograma y *Text-guided image-to-image generation* muestran un rendimiento moderado en esta área. En contraste, los datasets Inicial Aleatorio, CycleGAN, Superposición de Recortes, Combinado y Combinado Modificado muestran un bajo rendimiento, lo cual es favorable ya que indica una baja tasa de falsos positivos. Los casos de falsos positivos pueden variar desde la detección de montañas u orillas (véase Ilustración 5.57a) como barcos, hasta la detección conjunta del barco y su entorno, resultando en una caja delimitadora no correcta (véase Ilustración 5.57b).



(a) Falso positivo, ha detectado incorrectamente un objeto en el fondo de la imagen



(b) Falso positivo detectado en una parte del barco. La detección incorrecta ocurre dentro del objeto de interés

Ilustración 5.57: Ejemplos de falsos positivos en el dataset Inicial Histograma

Borroso

Aunque las imágenes borrosas pueden ser reconocibles como barcos para el ojo humano, sigue siendo un desafío significativo para todos los conjuntos de datos. En general, todos muestran un rendimiento muy bajo en la detección de barcos en condiciones borrosas, sin lograr detectar ningún barco en estas circunstancias.

5.12. Sprint 11: Evaluación de Modelos YOLO

5.12.1. Comparativa Dataset Combinado

Al realizar un análisis comparativo de los modelos YOLO para la detección de barcos utilizando el Dataset Combinado, se observan diferencias significativas en el desempeño de los modelos YOLOv7, YOLOv8 y YOLOv9. La evaluación se ha basado en diversas métricas, incluyendo la matriz de confusión, el F1 Score, la precisión, la relación precisión-recuperación (PR) y la recuperación. Los resultados detallados de esta evaluación se presentan en la Tabla 5.3.

Tabla 5.3: Resultados validación YOLO con el Dataset Combinado

Métrica \ Modelo	YOLOv7	YOLOv8	YOLOv9
Matriz de Confusión	0.86	0.77	0.73
F1 Score (F1)	0.83	0.80	0.75
Precisión (P)	0.942	0.931	0.885
Precision-Recall (PR)	0.762	0.822	0.795
Recuperación (R)	0.77	0.74	0.70

YOLOv7 se destaca como el modelo con mejor desempeño general en varias métricas clave. Con una puntuación de 0.86 en la matriz de confusión, YOLOv7 muestra una alta capacidad para clasificar correctamente las embarcaciones. Además, presenta el F1 Score más alto con un valor de 0.83, lo que indica un excelente equilibrio entre precisión y recuperación. En términos de precisión, YOLOv7 alcanza un valor de 0.942, confirmando su eficacia en la identificación correcta de los barcos.

Por otro lado, **YOLOv8** muestra un rendimiento muy competitivo, aunque ligeramente inferior en algunas métricas en comparación con YOLOv7. Su puntuación en la matriz de confusión es de 0.77 y su F1 Score es de 0.80. Sin embargo, YOLOv8 presenta una alta precisión de 0.931 y una relación precisión-recuperación (PR) de 0.822, lo que indica un rendimiento robusto en términos de equilibrio entre precisión y recuperación. Además, YOLOv8 muestra una recuperación de 0.74, que aunque menor que la de YOLOv7, sigue siendo significativa.

Finalmente, **YOLOv9** obtiene los resultados más bajos en la mayoría de las métricas evaluadas. Su puntuación en la matriz de confusión es de 0.73, y su F1 Score es de 0.75. La precisión de YOLOv9 es de 0.885, la más baja entre los tres modelos, y su relación precisión-recuperación (PR) es de 0.795. La recuperación también es la menor, con un valor de 0.70, lo que sugiere que YOLOv9, aunque efectivo, no alcanza el mismo nivel de equilibrio y eficacia que los otros dos modelos.

Después de realizar la inferencia en la Ilustración 5.58, se ha observado que YOLOv8 y YOLOv9 han solucionado los problemas de detección de barcos térmicos en condiciones de calima. En particular, YOLOv8 muestra una notable capacidad para detectar barcos superpuestos de manera individual, dependiendo del grado de superposición. En contraste, YOLOv7 tiende a identificar ambos barcos como una sola entidad, mientras que YOLOv9 únicamente detecta uno de los barcos superpuestos.

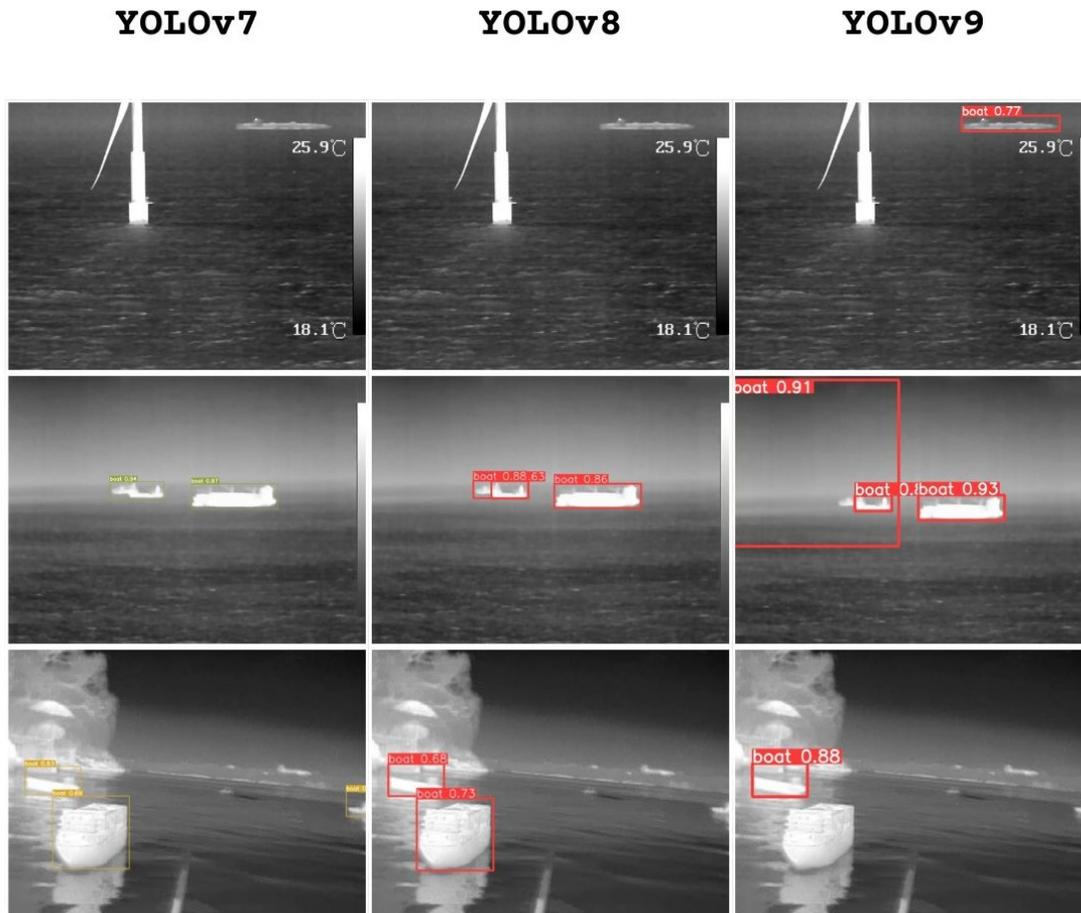


Ilustración 5.58: Inferencia del entrenamiento del dataset inicial y Text-guided image-to-image generation y Superposición de Recortes con YOLOv7, YOLOv8 y YOLOv9. La imagen superior contiene calima, en la segunda barcos superpuestos y la tercera es un caso general

Considerando las métricas evaluadas, YOLOv7 se destaca como el modelo con mejor desempeño general en términos de matriz de confusión, F1 Score, precisión y recuperación. Aunque YOLOv8 mostró una mejor relación precisión-recuperación (PR), su rendimiento en otras métricas clave fue ligeramente inferior al de YOLOv7.

La Tabla 5.4 muestra los resultados de la validación de tres versiones: YOLOv7, YOLOv8 y YOLOv9 en el dataset Combinado, evaluando el desempeño en diversas condiciones específicas. Los resultados se presentan cualitativamente (Muy baja, Baja, Moderada, Alta, Muy alta) con un código de colores para facilitar la visualización.

Tabla 5.4: Resultados de la Validación de diferentes YOLO con Dataset Combinado

YOLO \ Condición	YOLOv7	YOLOv8	YOLOv9
Primer Plano	Baja	Moderada	Baja
Oclusión	Alta	Muy Alta	Moderada
Calima	Moderada	Muy baja	Alta
Parcial	Muy Alta	Muy Alta	Muy Alta
Objeto no barco parcial	Muy baja	Baja	Muy baja
Lejos	Muy Alta	Alta	Alta
Cerca	Alta	Alta	Baja
Borroso	Muy baja	Muy baja	Muy baja
Falso positivo	Baja	Moderada	Alta

La evaluación cualitativa muestra que YOLOv8 destaca en la detección bajo condiciones de oclusión, identificando barcos en todas las imágenes con alta confianza. Sin embargo, su rendimiento es inferior en condiciones de calima y primer plano, donde no detecta ningún barco o los detecta con baja confianza. YOLOv7 y YOLOv9 muestran un rendimiento bajo en condiciones de primer plano, aunque YOLOv9 tiene un desempeño moderado en oclusión, solo detectando consistentemente el barco que ocluye.

En situaciones con objetos no barco parciales, tanto YOLOv7 como YOLOv9 presentan un rendimiento muy bajo, mientras que YOLOv8 detecta incorrectamente un muro. En condiciones de detección a distancia, YOLOv7 tiene un desempeño muy alto, superando a YOLOv8 y YOLOv9, que también muestran un alto rendimiento. Para detección cercana, YOLOv7 y YOLOv8 mantienen un alto desempeño, pero YOLOv9 tiene un rendimiento significativamente peor. En condiciones de imágenes borrosas, todas las versiones de YOLO presentan un rendimiento muy bajo, sin capacidad de detección efectiva.

5.12.2. Comparativa Dataset Combinado Modificado

Observando los resultados de los entrenamientos de cada modelo para el Dataset Combinado Modificado, presentados en la Tabla 5.5. En cuanto a la matriz de confusión, YOLOv7 presenta el mejor desempeño con un valor de 0.81, lo que indica una mayor precisión en las predicciones correctas y una menor cantidad de falsos positivos y negativos. Por otro lado, YOLOv9 obtiene el peor resultado con un valor de 0.74. Respecto al F1 Score, YOLOv8 destaca con un valor de 0.81, sugiriendo un equilibrio óptimo entre precisión y recuperación. En cambio, YOLOv7 tiene el desempeño más bajo con un valor de 0.71. En términos de precisión, YOLOv8 nuevamente lidera con un valor de 0.951, indicando una alta proporción de verdaderos positivos entre los ejemplos predichos como positivos. YOLOv9, por su parte, obtiene el valor más bajo con 0.879.

Tabla 5.5: Resultados de la validación de YOLO con el Dataset Combinado Modificado

Métrica \ Modelo	YOLOv7	YOLOv8	YOLOv9
Matriz de Confusión	0.81	0.78	0.74
F1 Score (F1)	0.71	0.81	0.79
Precisión (P)	0.892	0.951	0.879
Precision-Recall (PR)	0.579	0.841	0.832
Recuperación (R)	0.59	0.78	0.74

La métrica de Precision-Recall muestra que YOLOv8 logra el mejor resultado con 0.841, lo que evidencia un rendimiento sólido en términos de equilibrio entre precisión y recuperación. YOLOv7 tiene el desempeño más débil en esta métrica con un valor de 0.579. Finalmente, en la métrica de recuperación, YOLOv8 sobresale con un valor de 0.78, lo que indica que puede detectar una mayor proporción de verdaderos positivos entre todos los positivos reales. YOLOv7, sin embargo, presenta el peor rendimiento con un valor de 0.59.

De acuerdo con los resultados obtenidos y el análisis de las diferentes métricas, se puede concluir que YOLOv8 ofrece el mejor rendimiento general. Este modelo lidera en F1 Score, precisión, relación precisión-recuperación y recuperación, lo que lo convierte en la opción preferida para aplicaciones que requieren alta precisión y recuperación en la detección de objetos.

Aunque YOLOv7 muestra un buen desempeño en la métrica de la matriz de confusión, su rendimiento es inferior en las otras métricas evaluadas. Por otro lado, YOLOv9 presenta un rendimiento intermedio, mejor que YOLOv7 en algunas métricas, pero generalmente inferior a YOLOv8.

Después de realizar la inferencia situada en la Ilustración 5.59, se ha observado que YOLOv8 y YOLOv9 siguen detectando barcos térmicos en condiciones de calima. Con respecto a la superposición de las embarcaciones, YOLOv8 y YOLOv7 se mantienen con el mismo comportamiento que con el dataset sin ser modificado, mientras que YOLOv9 ahora detecta ambos barcos como uno, siguiendo el comportamiento de YOLOv7.

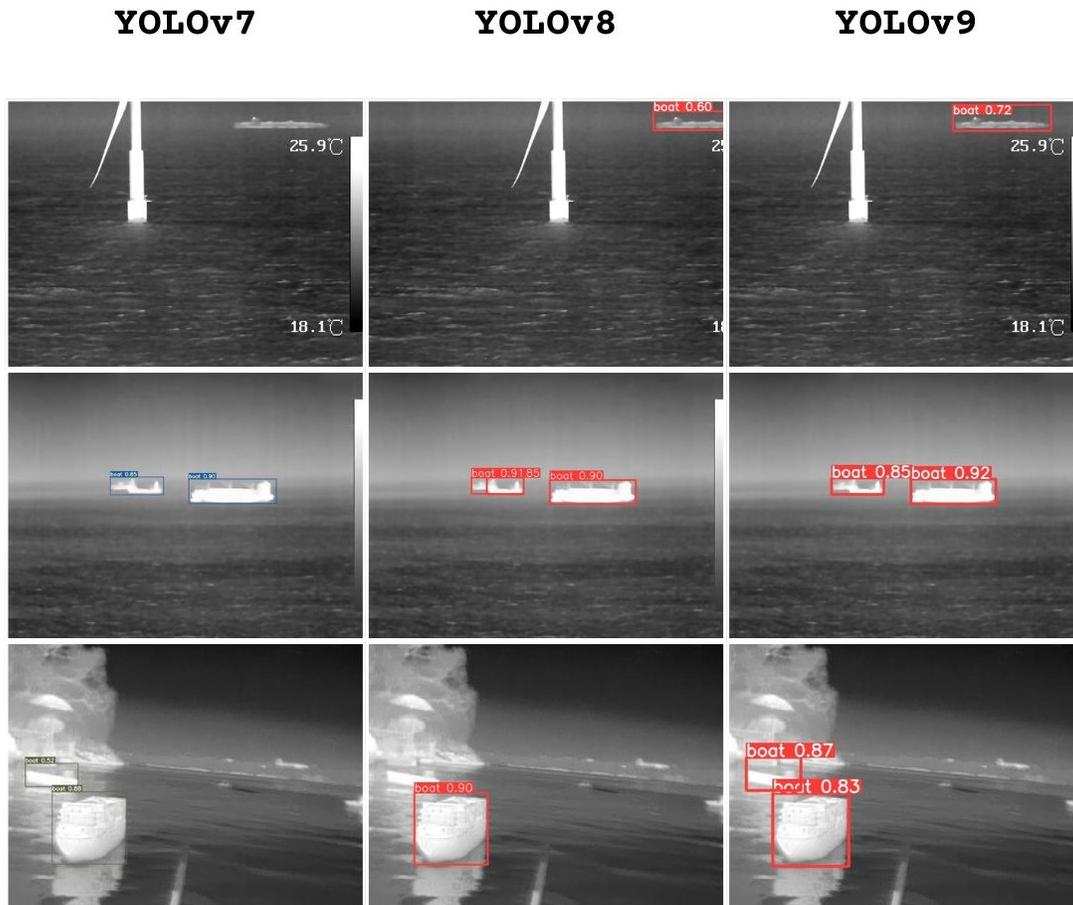


Ilustración 5.59: Inferencia del entrenamiento del dataset inicial y generación de imagen a imagen guiada por texto, con superposición de recortes modificada usando YOLOv7, YOLOv8 y YOLOv9. La imagen superior muestra calma, en la segunda se observan barcos superpuestos y en la tercera un caso general

Por otro lado, la Tabla 5.6 muestra el análisis de la validación de las versiones de YOLO utilizando el dataset Combinado Modificado en diferentes condiciones específicas.

Tabla 5.6: Resultados de la Validación de diferentes YOLO con Dataset Combinado Modificado

YOLO Condición	YOLOv7	YOLOv8	YOLOv9
Primer Plano	Baja	Moderada	Baja
Oclusión	Alta	Muy Alta	Alta
Calima	Moderada	Alta	Alta
Parcial	Alta	Muy Alta	Alta
Objeto no barco parcial	Muy baja	Muy baja	Muy baja
Lejos	Alta	Muy Alta	Alta
Cerca	Baja	Alta	Moderada
Borroso	Muy baja	Baja	Muy baja
Falso positivo	Baja	Moderada	Moderada

YOLOv8 destaca especialmente en condiciones de oclusión, detección a distancia y visibilidad parcial. Sin embargo, enfrenta problemas con falsos positivos y la detección de objetos borrosos, donde su rendimiento es limitado.

Por otro lado, la capacidad para manejar objetos borrosos de YOLOv9 es deficiente, y genera falsos positivos en situaciones desafiantes, lo que disminuye su efectividad general.

En comparación, YOLOv7 maneja adecuadamente la oclusión y la detección a distancia. Además, este modelo tiene dificultades significativas con la detección de objetos cercanos y borrosos, lo que afecta su eficacia en aplicaciones prácticas.

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusión Datasets

Después de analizar diversas condiciones, se han identificado las áreas de fortaleza y debilidad de cada dataset, lo que permite entender mejor su aplicación y áreas de mejora. Es evidente que todos los datasets necesitan mejorar en la capacidad de detectar barcos en primer plano. Tanto COCO como CycleGAN muestran desventajas significativas y no contribuyen efectivamente a la detección de barcos en imágenes térmicas, ya sea por falta de entrenamiento específico para estos entornos o por la baja calidad de las imágenes generadas.

Aunque los datasets Inicial Aleatorio e Histograma destacan en la detección de barcos lejanos y parcialmente visibles, existen opciones más adecuadas para diversas condiciones adversas.

Los datasets Superposición de Recortes y *Text-guided image-to-image generation*, aunque muestran un rendimiento moderado y consistente, no sobresalen en condiciones específicas.

Por último, los datasets Combinado y Combinado Modificado muestran un rendimiento superior en condiciones desafiantes como oclusión, visibilidad parcial y reducción de falsos positivos. Estos datasets se destacan por su robustez y adaptabilidad, siendo las opciones más confiables para aplicaciones en entornos adversos.

6.2. Conclusión Modelo YOLO

Para la detección a tiempo real de barcos en cámara térmica, YOLOv8 utilizando el dataset Combinado Modificado es la mejor opción. Este modelo proporciona un equilibrio superior entre precisión y recuperación, esencial para aplicaciones en tiempo real donde la rapidez y exactitud son críticas. Aunque YOLOv7 puede ser útil en situaciones específicas de detección a distancia, YOLOv8 se destaca como la solución más efectiva y versátil para este propósito.

YOLOv9, por su parte, necesita mejoras antes de poder ser considerado competitivo para aplicaciones en tiempo real.

6.3. Trabajo Futuro

Hay varias oportunidades de mejora y expansión futuras del proyecto que podrían mejorar significativamente la precisión y robustez del modelo de detección de objetos en imágenes marítimas. A continuación se detallan algunas de estas áreas claves en las que se puede enfocar:

6.3.1. Generación de Entornos

Un enfoque es el uso de Redes Generativas Antagónicas (GANs) para generar imágenes en condiciones adversas. Por ejemplo, estas condiciones pueden ser un fuerte oleaje, tormentas, o un iceberg. Al agregar estas imágenes generadas al entrenamiento, se espera mejorar no solo el desempeño del modelo, sino también su velocidad en respuestas extremadamente adversas. Al mismo tiempo, el campo de prueba validado se vuelve más diverso y grande, lo que reduce el sobreajuste.

6.3.2. Integración en el Sistema de la Empresa

Un paso crucial para llevar este proyecto a la práctica es la integración del modelo en el sistema operativo de la empresa. Esto incluye el desarrollo de un sistema de alertas que notifique a los operadores cuando se detecte un barco acercándose a la plataforma. Además, se podría implementar un sistema de detección que opere específicamente fuera de la zona de la plataforma, ya que esta área se detecta consistentemente con un alto umbral con la etiqueta barco, como se muestra en la Ilustración 6.1. Esta integración no solo automatizaría el proceso de monitoreo, sino que también aumentaría la seguridad y eficiencia operativa.

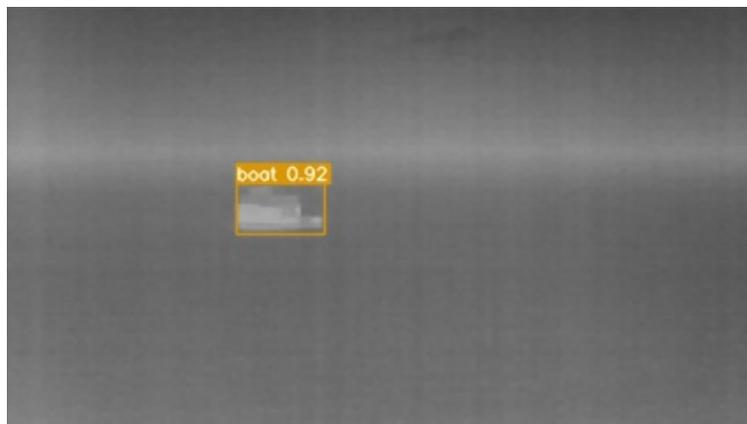


Ilustración 6.1: Plataforma en el entorno marítimo del Mirador, detectada como barco (FP)

6.3.3. Mejora de Imágenes

Otra área es la mejora de la calidad de las imágenes, ayudando a obtener detalles más claros y precisos de los objetos, especialmente en condiciones de baja visibilidad.

6.3.4. Exploración de Nuevas Distribuciones

Para mejorar aún más el rendimiento del modelo y su robustez en la detección de objetos, es crucial considerar métodos avanzados de distribución del dataset. En la actualidad, la distribución del dataset se realiza mediante los histogramas. Como propuesta a futuro, se pueden explorar técnicas más sofisticadas para la partición del dataset, tales como validación cruzada.

6.3.5. *Hard Negative*

Además, se puede considerar añadir *hard negatives* al dataset, como boyas u otros objetos que puedan ser confundidos con barcos. Esto puede ayudar al modelo a aprender a distinguir mejor entre objetos similares y mejorar su precisión general.

Bibliografía

- [1] Ahmad, M. O., Markkula, J., and Oivo, M. (2016). Insights into the perceived benefits of kanban in software companies: Practitioners' views.
- [2] Albarqi, A. A. and Qureshi, R. (2018). The proposed l-scrumban methodology to improve the efficiency of agile software development.
- [3] Arévalo, V. M., González, J., and Ambrosio, G. (2004). La librería de visión artificial opencv aplicación a la docencia e investigación.
- [4] Avrahami, O., Lischinski, D., and Fried, O. (2022). Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18208–18218.
- [5] Barash, Y., Klang, E., Konen, E., and Sorin, V. (2023). Chatgpt-4 assistance in optimizing emergency department radiology referrals and imaging selection. *Journal of the American College of Radiology*.
- [6] Bourou, A., Daupin, K., Dubreuil, V., Thonel, A. D., Lallemand-Mezger, V., and Genovesio, A. (2023). Unpaired image-to-image translation with limited data to reveal subtle phenotypes. *arXiv*.
- [7] Co., H. M. S. (2023). Thermal & optical bi-spectrum ptz network camera user manual. Hickvision.
- [8] Corona, E. and Pani, F. E. (2013). A review of lean-kanban approaches in the software development.
- [9] Dalhousie (2023). Marine_image_thermal dataset.
- [10] dilipprasad (n.d.). Singapore maritime dataset. dilipprasad.
- [11] Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., and Tian, Q. (2019). Centernet: Keypoint triplets for object detection. *arXiv.org*.
- [12] Face, H. (n.d.). Image-to-image.
- [13] FLIR, T. (2020). How do thermal cameras work? Teledyne FLIR.

- [14] Gokay, D., Simsar, E., Atici, E., Ahmetoglu, A., Yuksel, A. E., and Yanardag, P. (2018). Graph2pix: A graph-based image to image translation framework.
- [15] Goodfellow, I. J., Pouget-Abadie†, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio§, Y. (2014). Generative adversarial nets.
- [16] Handelman, G. S., Kok, H. K., Chandra, R. V., Razavi, A. H., Huang, S., Brooks, M., Lee, M. J., and Asadi, H. (2018). Peering into the black box of artificial intelligence: Evaluation metrics of machine learning methods.
- [17] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del R, J. F., Wiebe, M., Peterson, P., Gerard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with numpy.
- [18] Hikvision (n.d.). Thermal & optical bi-spectrum network stable ptz camera. Hickvision.
- [19] Hiranabe, K. (2008). Kanban applied to software development: from agile to lean. InfoQ.
- [20] IBM (2023). Bm maximo visual inspection version 8.6.0 and later, and saas.
- [21] IBM (n.d.a). ¿qué es pytorch? — ibm.
- [22] IBM (n.d.b). ¿qué es una red neuronal? IBM.
- [23] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*.
- [24] Kütük, Z. and Algan, G. (n.d.). Semantic segmentation for thermal images: A comparative survey.
- [25] Lazarus, G. and Grace, K. (n.d.). Minimally juxtapository tasks as a co-creative systems user study method.
- [26] Lee, N. (n.d.). Leonardo.ai help center.
- [27] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E., Fu, C., and Berg, A. C. (2016). Ssd: Single shot multibox detector. arXiv.org.
- [28] Ltd, L. I. P. (n.d.). Ai image generator create art, images and video. Leonardo AI.
- [29] Lu, S., Liu, Y., and Kong, A. W.-K. (2023). Tf-icon: Diffusion-based training-free cross-domain image composition.
- [30] Mason (2023). test dataset.
- [31] Mert, S., Stoerzer, P., Brauer, J., Fuchs, B., Haas-Lützenberger, E. M., Demmer, W., Giunta, R. E., and Nuernberger, T. (2024). Diagnostic power of chatgpt 4 in distal radius fracture detection through wrist radiographs.

- [32] Morandini, M., Coleti, T. A., Oliveira, E., and Corrêa, P. L. P. (2021). Considerations about the efficiency and sufficiency of the utilization of the scrum methodology: A survey for analyzing results for development teams.
- [33] Nguyen, T. X. B., Rosser, K., and Chahl, J. (2021). A review of modern thermal imaging sensor technology and applications for autonomous aerial navigation. MDPI.
- [34] Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. (2022). Glide: Towards photorealistic image generation and editing with text-guided diffusion models.
- [35] Nirgudkar, S., DeFilippo, M., Sacarny, M., Benjamin, M., and Robinette, P. (2023). Massmind: Massachusetts marine infrared dataset.
- [36] Niu, L., Cong, W., Liu, L., Hong, Y., Zhang, B., Liang, J., and Zhang, L. (2021). Making images real again: A comprehensive survey on deep image composition. *arXiv preprint arXiv:2106.14490*.
- [37] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.
- [38] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection.
- [39] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv.org.
- [40] Rhodes, A. D., Quinn, M. H., and Mitchell, M. (2016). Fast on-line kernel density estimation for active object localization. arXiv.
- [41] Schmidhuber, J. (2014). Deep learning in neural networks: An overview.
- [42] Terven, J. R. and Cordova-Esparza, D. M. (2024). A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas.
- [43] Tsay, J., Mummert, T., Bobroff, N., and Hirzel, M. (2018). Runway: machine learning model experiment management tool. ResearchGate.
- [44] Ultralytics (2024). <https://docs.ultralytics.com/es/guides/yolo-performance-metrics/>. Ultralytics.
- [45] Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (n.d.). Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors.
- [46] Wang, C.-Y., Yeh, I.-H., and Liao, H.-Y. M. (2024). Yolov9: Learning what you want to learn using programmable gradient information.
- [47] Wang, S. and Su, Z. (2019). Metamorphic testing for object detection systems.
- [48] Westlake, S. (2020). Irships. Cranfield Online Research Data (CORD).

- [49] Westlake, S. T., Volonakis, T. N., Jackman, J., James, D. B., and Sherriff, A. (2020). Deep learning for automatic target recognition with real and synthetic infrared maritime imagery. In *Artificial Intelligence and Machine Learning in Defense Applications II*, volume 11543, page 1154309. International Society for Optics and Photonics.
- [50] Zhai, S., Shang, D., Wang, S., and Dong, S. (2020). Df-ssd: An improved ssd object detection algorithm based on densenet and feature fusion. *IEEE Access*.
- [51] Zhang, P., Liu, Z., Hu, X., Sun, Y., Deng, X., Zhu, B., and Yang, Y. (2023). Constraints and recent solutions of optical camera communication for practical applications. *MDPI*.
- [52] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*.
- [53] Zou, Z., Chen, K., Shi, Z., Guo, Y., and Ye, J. (n.d.). Object detection in 20 years: A survey. *IEEE Journals & Magazine — IEEE Xplore*.