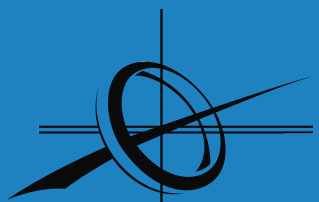
POLITÉCNICA



Universidad Politécnica de Madrid

ETSI SISTEMAS INFORMÁTICOS

Detección y clasificación de aeronaves en imágenes SAR mediante machine learning y generación sintética con modelos de difusión.

Proyecto Fin de Grado

Grado en Ciencia de Datos e Inteligencia Artificial

Autor:
Beatriz Noelia Vulpe

Tutor:
Francisco Serradilla García

28 junio 2025

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingeniería de Sistemas Informáticos

Detección y clasificación de aeronaves en imágenes SAR mediante machine learning y generación sintética con modelos de difusión.

Proyecto Fin de Grado

Grado en Ciencia de Datos e Inteligencia Artificial

Curso académico 2024-2025

Autor:
Beatriz Noelia Vulpe

Tutor:
Francisco Serradilla García

A mi madre y a mi hermana, por aguantarme. Una vez al día me tocaba mencionar algo sobre los avances del proyecto y, aunque no entendierais del todo de lo que estaba hablando, me dejasteis seguir con el runrún continuo, aunque estuvierais ya cansadas de mí, seguro.

A Félix, por dedicar su tiempo e interesarse lo suficiente en este proyecto como para leérselo.

A mis amigos, que me acompañaron en este proceso y me ayudaron a desconectar de vez en cuando. No muchos de vosotros leeréis esto, pero se os quiere.

A mi tutor, por confiar en mí para llevar el proyecto y guiarme en este inicio de mi etapa profesional como graduada.

Y, por supuesto, a todos los profesores que despertaron mi curiosidad y me permitieron realizar proyectos que me ayudaron a entender qué quiero seguir haciendo en mi futuro profesional.

Gracias a todos vosotros me despido de esta etapa y doy comienzo a una nueva.

Resumen

Este trabajo de fin de grado presenta un sistema automático basado en aprendizaje profundo (YOLOv8), para la detección y clasificación de aeronaves en imágenes de Radar de Apertura Sintética (SAR). Financiado parcialmente por la Agencia Espacial Europea, en colaboración con el INSIA y HISDESAT Servicios Estratégicos S.A., este proyecto aborda un desafío crucial en la vigilancia aérea y la seguridad.

Se ha optimizado YOLOv8, logrando una precisión del 95.86 % y un recall del 94.93 % en la detección, superando a otros modelos de estado del arte. Por otro lado, el modelo demostró una notable robustez en la clasificación multiclase, alcanzando un 94.34 % de precisión. Sin embargo, la distribución desequilibrada de los datos afectó la precisión de detección en áreas sin aviones y causó clasificaciones interclase incorrectas al aumentar la granularidad.

Para mitigar este desequilibrio y la escasez de datos, se ha explorado la generación de imágenes SAR sintéticas utilizando modelos de difusión. Esta aproximación de IA generativa ha mostrado resultados muy prometedores en la creación de firmas radar, demostrando su viabilidad para futuras ampliaciones del proyecto.

El sistema, con su arquitectura modular, ya ha sido implementado y presentado a HISDESAT para su uso interno. Los principales hallazgos están en proceso de publicación.

Palabras clave: *Detección de aeronaves, Visión por computador, Aprendizaje profundo, Clasificación multiclase, Detección de objetos, Imagen SAR, Imagen de radar de apertura sintética, YOLO, IA generativa, Generación de imágenes sintéticas, FLUX, Stable Diffusion*

Abstract

The research presented in this paper focuses on an automated system based on deep learning(YOLOv8) for the detection and classification of aircraft in Synthetic Aperture Radar (SAR) imagery. Partially funded by the European Space Agency, in collaboration with INSIA and HISDESAT Servicios Estratégicos S.A., this project addresses a crucial challenge in aerial surveillance and security.

YOLOv8 was optimized, achieving a detection precision of 95.86 % and a recall of 94.93 %, outperforming other state-of-the-art models. Furthermore, the model demonstrated notable robustness in multiclass classification, reaching 94.34 % precision. However, an unbalanced data distribution affected detection precision in areas without aircraft and led to incorrect detections and interclass classifications when granularity was increased.

To mitigate this imbalance and data scarcity, we explored the generation of synthetic SAR images using diffusion models. This generative AI approach has shown very promising results in creating radar signatures, demonstrating its viability for future research.

The system, with its modular architecture, has already been implemented and presented to HISDESAT for in-house use. The main findings are currently in the process of publication.

Keywords: *Aircraft Detection, Computer Vision, Deep Learning, Multi-label Classification, Object Detection, SAR Imaging, Synthetic Aperture Radar Imaging, YOLO, Generative AI, Synthetic Image Generation, FLUX, Stable Diffusion*

Índice

Índice de tablas

Índice de figuras

Capítulo 1

Introducción

En los últimos años, los avances en tecnologías de teledetección han transformado significativamente ámbitos tan diversos como la monitorización ambiental, la defensa y la seguridad. Entre estas tecnologías, una de las más innovadoras es el Radar de Apertura Sintética (SAR), que ha revolucionado la obtención de imágenes de alta resolución en condiciones complejas y desafiantes.

A diferencia de los sensores ópticos tradicionales, que dependen pasivamente de la luz solar reflejada, el SAR adopta un enfoque activo: emite pulsos electromagnéticos y capta las señales de retorno. Esta capacidad de sondear el entorno de manera independiente de las condiciones de iluminación lo convierte en una herramienta especialmente eficaz en escenarios donde los sensores ópticos fallan, como en presencia de nubes, humo o durante la noche [1].

Entre sus múltiples usos, la detección de aeronaves ha despertado un interés creciente en la investigación. Esta tecnología resulta especialmente valiosa para la vigilancia continua de aeropuertos y del espacio aéreo circundante, ya que permite identificar rápidamente posibles amenazas y optimizar las operaciones terrestres.

No obstante, a pesar de sus capacidades avanzadas, el uso de SAR en la detección de aeronaves presenta importantes desafíos prácticos. El principal obstáculo radica en cómo se representan estos vehículos en las imágenes SAR: sus superficies lisas generan patrones irregulares de puntos de dispersión brillantes, en lugar de formas coherentes, lo que puede fragmentar una sola aeronave en múltiples firmas (patrones característicos de reflexión) aisladas. Esta fragmentación se complica aún más por la variabilidad en el comportamiento de dispersión de los distintos componentes del vehículo aéreo y su interacción con las señales de radar. A ello se suma el entorno aeroportuario, que puede producir retornos de radar similares a los de las propias aeronaves [2]. Además, las señales reflejadas por radar de un mismo avión pueden variar drásticamente según el ángulo de observación, lo que dificulta su detección y aumenta el riesgo de errores de clasificación. Estos retos son especialmente críticos en sistemas de detección automatizados, donde las características intrínsecas del SAR (como el ruido coherente o speckle, el bajo contraste o la compleja interferencia de fondo) representan un desafío incluso para modelos avanzados de aprendizaje profundo. Si bien incrementar la complejidad del modelo puede mejorar la precisión global [3], esto suele comprometer la velocidad de procesamiento.

Ante el volumen masivo de datos que se generan diariamente, surge una pregunta fundamental: ¿cómo podemos procesar de manera eficiente y precisa esta información, superando las complejidades inherentes de las imágenes SAR?

El presente trabajo aborda directamente esta cuestión mediante el desarrollo de un sistema automático e integral, utilizando técnicas avanzadas de visión por computador y aprendizaje profundo. El proyecto no solo implementa meto-

dologías consolidadas, como los modelos de detección de objetos de la familia YOLO (You Only Look Once), sino que también explora una de las nuevas fronteras de la inteligencia artificial: el uso de modelos generativos de difusión para crear datos sintéticos. Esta línea busca mitigar uno de los principales cuellos de botella del campo: la escasez de datos etiquetados de alta calidad.

Esta investigación, que ha contado con financiación parcial del proyecto ATR4PAZ de la Agencia Espacial Europea (ESA), se ha desarrollado mediante una colaboración entre el Instituto Universitario de Investigación del Automóvil (INSIA) e HISDESAT Servicios Estratégicos, S.A., en el marco de unas prácticas curriculares.

1.1. Motivación

La necesidad de este sistema responde a una demanda directa del sector aeroespacial y de defensa. Organizaciones y empresas como HISDESAT, colaboradora en este proyecto, gestionan flujos constantes de datos satelitales que requieren un análisis rápido y fiable. La automatización no solo permite reducir la carga de trabajo de los analistas humanos y minimizar errores, sino que también acelera los tiempos de respuesta ante eventos críticos. Además, su diseño modular facilitará su uso independiente dentro de la empresa, permitiendo la incorporación continua de nuevos datos de entrenamiento y la reevaluación de su rendimiento a lo largo del tiempo.

Desde el punto de vista teórico, el proyecto se fundamenta en el éxito demostrado de las Redes Neuronales Convolucionales (CNNs) en el campo de la visión por computador. Entre las diversas arquitecturas de aprendizaje profundo, la familia de modelos YOLO ha demostrado ser especialmente prometedora en la detección de objetos en imágenes SAR. Análisis comparativos entre modelos de detección modernos, como YOLOv8, RTMDet y arquitecturas basadas en transformers como DETR, han evidenciado consistentemente el rendimiento superior de YOLO, particularmente en aplicaciones de monitorización en tiempo real donde la velocidad y precisión son cruciales [4].

Este trabajo se construye sobre las exploraciones del TFG «Detección de Aviones Mediante Inteligencia Artificial en Imágenes Satelitales de Radar», de Pablo Alonso López, que ya destacaba el prometedor desempeño inicial de YOLO frente a otras arquitecturas en el conjunto de datos proporcionado por la empresa colaboradora. El presente proyecto se basa en el rendimiento consolidado de YOLOv8 tanto en imágenes ópticas como SAR, eligiéndolo por su arquitectura bien establecida y el amplio conocimiento disponible sobre su comportamiento en distintos escenarios.

A pesar de estos avances, persisten dos brechas clave que se busca abordar. En primer lugar, la literatura actual carece de una evaluación sistemática y comparativa de modelos de detección aplicados específicamente a conjuntos de datos personalizados que presentan desafíos únicos como variaciones en la reflectividad del radar y la apariencia del objeto. En segundo lugar, y más importante aún, la escasez de grandes volúmenes de datos SAR etiquetados de alta calidad sigue siendo el principal obstáculo para el desarrollo de modelos de inteligencia artificial robustos y generalizables. Este TFG plantea una solución mediante el

desarrollo experimental de un sistema basado en modelos de difusión para la generación de datos sintéticos, una técnica emergente cuya aplicación en este contexto ha sido escasamente explorada.

1.2. Objetivos

El objetivo principal de este proyecto es desarrollar y evaluar un sistema automático basado en aprendizaje profundo para la detección y clasificación precisa de aeronaves en imágenes de Radar de Apertura Sintética.

Como objetivo secundario exploratorio, se investigará el uso de datos sintéticos generados mediante modelos de difusión para explorar su potencial futuro en la mejora de la robustez y rendimiento del sistema. Dados los requisitos del proyecto para su aplicación dentro de la empresa, el sistema se desarrollará con una arquitectura modular que permita el despliegue independiente de sus componentes.

Para lograr estos objetivos generales, se establecen los siguientes objetivos específicos:

- **Evaluar sistemáticamente modelos de detección**: Realizar una evaluación comparativa exhaustiva de diversas configuraciones de modelos de la familia YOLO, variando parámetros clave como el tamaño de la imagen de entrada, la tasa de aprendizaje y las estrategias de preprocesamiento del conjunto de datos. Esto permitirá identificar la arquitectura más eficiente, optimizada para nuestro conjunto de datos específico.
- **Desarrollar un sistema de clasificación**: Implementar capacidades de clasificación de aeronaves en dos niveles de granularidad (5 clases básicas y 10 clases extendidas) para analizar el compromiso entre complejidad computacional y precisión de identificación según los requerimientos operativos específicos.
- **Explorar la generación sintética de datos SAR**: Desarrollar un prototipo de modelo de difusión para la generación de imágenes SAR sintéticas de aeronaves como prueba de concepto, evaluando la calidad visual y diversidad de las muestras generadas para establecer la viabilidad de esta aproximación en futuros desarrollos y posibles estrategias de aumento de datos.

1.3. Estructura del documento

Este trabajo se organiza en cinco capítulos. Tras una introducción inicial, el segundo capítulo está dedicado al Estado de la Cuestión. En él se abordan los fundamentos teóricos y técnicos que sustentan el proyecto, comenzando con una revisión de la tecnología de radar de apertura sintética (SAR), seguida por un análisis de las principales técnicas de detección de objetos y una descripción de las arquitecturas de aprendizaje profundo más relevantes. Asimismo, se incluye una sección centrada en la generación de datos sintéticos, que examina tanto los conjuntos de datos públicos disponibles como los modelos generativos utilizados

en la literatura.

El tercer capítulo desarrolla la Metodología empleada. Se describe el conjunto de datos utilizado en los experimentos, así como el proceso seguido para su preparación y anotación. Seguidamente, se detallan los distintos experimentos realizados, comenzando por los relacionados con la detección y clasificación de aeronaves mediante modelos YOLO. En este apartado se explican la configuración experimental, las métricas de evaluación utilizadas, las variantes del modelo aplicadas y los enfoques empleados tanto para la detección binaria como para la clasificación multiclase. Posteriormente, se presenta la metodología seguida para la generación de imágenes SAR sintéticas, incluyendo el entrenamiento de LoRAs sobre los modelos Flux 1 y Stable Diffusion 3.5, y las distintas pruebas de generación realizadas a partir de texto, imagen y condiciones multimodales.

El capítulo cuarto recoge los Resultados obtenidos en las distintas fases del proyecto.

Finalmente, el documento cierra con las Conclusiones, donde se sintetizan los principales aportes del trabajo, se reflexiona sobre su posible impacto social y medioambiental, y se proponen líneas de investigación futuras que puedan ampliar o perfeccionar los resultados aquí obtenidos.

Capítulo 2

Estado de la cuestión

2.1. Fundamentos teóricos del SAR

El radar es una tecnología fundamental que emplea ondas electromagnéticas para la detección, localización y caracterización precisa de objetos a distancia. Su principio operativo es simple pero efectivo: transmite pulsos de energía electromagnética hacia un objetivo y analiza la señal reflejada. La ecuación del radar cuantifica la potencia recibida (P_r) por el sistema y es crucial para entender la interacción de las ondas con el entorno. Esta se expresa como:

$$P_r = \frac{P_t \times G_t \times G_r \times \lambda^2 \times \sigma}{(4\pi)^3 \times R^4} \tag{2.1}$$

Donde P_t es la potencia transmitida, G_t y G_r son las ganancias de las antenas transmisora y receptora respectivamente, λ representa la longitud de onda, σ es la sección transversal radar (RCS) del objetivo (una medida de su reflectividad) y R es la distancia al objetivo. Esta ecuación revela la dependencia de la potencia recibida respecto a la inversa de la cuarta potencia de la distancia, lo que implica una rápida atenuación de la señal con el aumento de la distancia.

Los radares convencionales tienen limitaciones significativas debido al tamaño físico de su antena y la distancia al objetivo. Para superar estas restricciones, el Radar de Apertura Sintética aprovecha el movimiento de la plataforma que lo transporta, ya sea un satélite o una aeronave. Al procesar múltiples ecos de radar recibidos mientras la plataforma se desplaza, el SAR es capaz de sintetizar una antena virtual de gran tamaño. Esto le permite lograr resoluciones espaciales extremadamente altas, independientemente de la distancia a la que se encuentre el objetivo.

El SAR opera fundamentalmente como un dispositivo de medición de distancia. El sensor emite pulsos de microondas hacia la superficie terrestre y mide el tiempo que tardan estas señales en regresar. Dado que todos los radares de imagen SAR son de mirada lateral (es decir, el haz radar incide oblicuamente sobre la superficie, formando un ángulo respecto a la vertical, y no apunta directamente hacia abajo como en los sensores de mirada nadiral), el tiempo que tarda la señal en alcanzar y regresar desde distintos puntos difiere, lo que posibilita su distinción espacial.

2.1.1. Arquitectura y geometría

Un sistema SAR consta esencialmente de un transmisor que genera pulsos de microondas, un receptor para captar los ecos, una antena que enfoca el haz y un sistema de almacenamiento y procesamiento de datos. En el caso de los satélites, el procesamiento de datos suele realizarse en estaciones terrenas.

La geometría del SAR es particular y se define por dos direcciones principales: la dirección de alcance (*range*), perpendicular a la trayectoria de vuelo de la plataforma, y la dirección azimutal, paralela a dicha trayectoria. El rango en inclinación (*slant range*) es la distancia directa entre el sensor y un punto en la superficie desde donde la señal es retrodispersada. A medida que la plataforma avanza, la grabación y procesamiento de las señales reflejadas en ambas dimensiones (rango y azimut) permiten construir una imagen bidimensional de la superficie. Otro parámetro esencial es el ángulo de incidencia, definido como el ángulo entre la vertical local y la dirección del haz radar. Este ángulo influye significativamente en la señal retrodispersada y, por tanto, en la apariencia de los objetos en la imagen SAR. Estos elementos se muestran en la Figura 2.1.

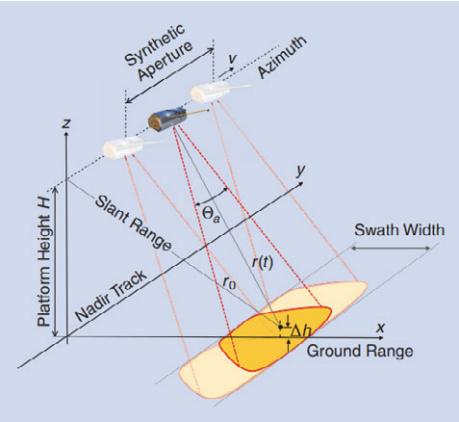


Figura 2.1: Geometría básica de un sistema de radar de apertura sintética. Se representan la trayectoria de la plataforma, el ángulo de visión θ_a, el rango inclinado (*slant range*), el rango en tierra (*ground range*) y la anchura de franja (*swath width*), junto con el movimiento del satélite y la apertura sintética generada [5].

Una imagen SAR es, por tanto, un mapa de reflectividad del área iluminada. La intensidad de la señal reflejada permite discriminar entre diferentes objetos del terreno. Aquellos objetos que devuelven una mayor cantidad de señal al radar se identifican como puntos brillantes en la imagen, indicando una alta retrodispersión. Por el contrario, las superficies más lisas o planas, que tienden a devolver menos señal al radar, aparecen como zonas oscuras. La Figura 2.2 presenta algunos ejemplos de imágenes.

Figura 2.2: Ejemplos representativos de imágenes SAR, incluyendo infraestructuras críticas y áreas urbanas [6].

2.1.2. Bandas de frecuencia y longitudes de onda

Los sistemas SAR operan en diversas bandas de frecuencia. Las principales bandas incluyen:

- **Banda L** (1-2 GHz, $\lambda \approx$ 15-30 cm): permite penetración profunda a través de vegetación y suelo, resultando ideal para estudios de biomasa forestal y contenido de humedad del terreno.
- **Banda S** (2-4 GHz, $\lambda \approx$ 7.5-15 cm): se destina principalmente a aplicaciones meteorológicas y monitoreo ambiental.
- **Banda C** (4-8 GHz, $\lambda \approx$ 3.75-7.5 cm): ampliamente utilizada en monitoreo oceánico, cartografía y estudios de superficie terrestre, ofreciendo un equilibrio entre capacidad de penetración y resolución espacial.
- **Banda X** (8-12 GHz, $\lambda \approx$ 2.5-3.75 cm): proporciona alta resolución con particular sensibilidad hacia estructuras de pequeño tamaño. Constituye la banda preferida para aplicaciones urbanas.
- **Banda Ku** (12-18 GHz, $\lambda \approx$ 1.7-2.5 cm): ofrece la mayor resolución disponible.

2.1.3. Resolución

La resolución es un parámetro crítico en cualquier sistema de imagen, y en SAR se distingue entre la resolución en alcance y la resolución azimutal. La resolución en alcance (ρ_r) se determina por el ancho de banda (B) del pulso transmitido:

$$\rho_r = \frac{c}{2B} \tag{2.2}$$

Donde c es la velocidad de la luz. Un mayor ancho de banda resulta en una mejor resolución en alcance, permitiendo diferenciar objetos que se encuentran a distancias similares pero ligeramente desplazados en la dirección del haz. Por otro lado, la resolución azimutal (ρ_a) es notablemente independiente de la distancia al objetivo, lo que es una ventaja significativa sobre los radares convencionales. Teóricamente, la resolución azimutal se define como:

$$\rho_a = \frac{L_a}{2} \tag{2.3}$$

Donde L_a es la longitud física de la antena. Esta capacidad de lograr una resolución azimutal constante a cualquier distancia es una de las principales fortalezas del SAR.

2.1.4. Modos de operación

Los sistemas SAR pueden operar en diversos modos:

- **Modo StripMap**: el modo de operación más básico, donde el haz radar se mantiene fijo en una dirección relativa a la trayectoria de vuelo. Proporciona una cobertura continua con resolución y ancho de franja constantes.
- **Modo ScanSAR**: para lograr una mayor cobertura terrestre a expensas de la resolución azimutal, el modo ScanSAR utiliza múltiples haces que barren la superficie de forma secuencial.
- **Modos Spotlight**: el haz se orienta continuamente hacia un punto de interés durante un tiempo prolongado.

  El modo HR Spotlight es una versión de mayor resolución, mientras que el modo Staring Spotlight maximiza el tiempo de iluminación del objetivo ajustando dinámicamente el punto de rotación virtual del haz. Este último modo es fundamental para lograr una resolución azimutal excepcional.

2.1.5. Polarización

La polarización de las ondas electromagnéticas define la orientación de su campo eléctrico y constituye una herramienta fundamental para extraer información detallada sobre las propiedades físicas de los objetos observados. Las configuraciones se clasifican según la orientación de la onda transmitida y recibida, utilizando una nomenclatura de dos letras donde la primera representa la polarización de la señal transmitida y la segunda la de la señal recibida (H para Horizontal, V para Vertical).

Las co-polarizaciones (HH, VV) son sensibles a la rugosidad superficial y las características estructurales de los objetos. Por su parte, las polarizaciones cruzadas (HV, VH) ofrecen información crucial sobre la dispersión volumétrica, revelando detalles que permanecen ocultos cuando se utilizan únicamente co-polarizaciones. El análisis de datos multi-polarización o polarimétrico representa una técnica avanzada que permite clasificar diferentes tipos de terreno y objetos con mayor precisión y detalle, maximizando la capacidad de interpretación de los datos SAR.

2.1.6. Procesamiento de la señal

Los sistemas SAR capturan tanto la intensidad como la fase de las señales reflejadas, generando datos complejos. Esta información dual permite aplicar técnicas de análisis que van más allá de las capacidades de los datos de intensidad convencionales. Los productos SAR se derivan del procesamiento de estos datos, siendo los más comunes:

- **Single Look Complex** (SLC): son los datos brutos del SAR, que incluyen la amplitud y la fase de la señal reflejada.
- **Ground Range Detected** (GRD): para una visualización más sencilla, los datos se procesan para corregir distorsiones geométricas y se proyectan sobre un plano de terreno. En este proceso, la información de fase generalmente se descarta.
- **Datos Multi-look**: para reducir el inherente ruido speckle, se promedian múltiples vistas de la misma área. Esto mejora la calidad visual de la imagen a cambio de una resolución más baja.

2.2. Detección de objetos

La detección de objetos ha experimentado una evolución significativa. Los enfoques clásicos, como los detectores CFAR (Constant False Alarm Rate), se basaban en análisis estadísticos del ruido de fondo para establecer umbrales de detección adaptativos. Aunque estos métodos ofrecían robustez en condiciones controladas, su capacidad para manejar entornos complejos y objetos con características variables era limitada [7] [8].

La irrupción del Deep Learning ha revolucionado este campo al permitir que los sistemas aprendan representaciones complejas directamente de los datos. En particular, las redes neuronales convolucionales (CNN) han demostrado una capacidad excepcional para extraer características relevantes de las imágenes SAR, superando las limitaciones de los métodos tradicionales. Esto se debe a su habilidad para adaptarse de forma superior a la variabilidad inherente de las características espectrales de los objetos y las condiciones ambientales [9].

2.3. Arquitecturas basadas en Deep Learning

Dentro de este contexto, las arquitecturas de detección de objetos se clasifican en dos enfoques principales: de una sola etapa y de dos etapas.

2.3.1. Detectores de dos etapas

Los modelos de detección de dos etapas operan de manera secuencial, imitando el procesamiento visual humano. En una primera fase, estos sistemas realizan pruebas preliminares para identificar todas las muestras que podrían contener objetos y generar regiones de interés (Regions of Interest, RoIs). Posteriormente, en una segunda etapa, estas regiones son clasificadas y su ubicación es ajustada para producir la caja de coordenadas (bounding box) final. A continuación, se describe la evolución de los modelos más reconocidos de esta familia.

R-CNN

Ante las limitaciones de los enfoques clásicos, Girshick et al. [10] presentaron R-CNN (Regions with Convolutional Neural Networks), la primera arquitectura que aplicó con éxito las redes neuronales convolucionales (CNN) a la detección de objetos. Su procedimiento consistía en tres etapas: primero, generar regiones de interés mediante un algoritmo externo como Selective Search; segundo, extraer características de cada RoI redimensionada usando una CNN; y tercero, clasificar estas características con una Máquina de Soporte Vectorial (SVM).

A pesar de su carácter pionero, R-CNN presentaba inconvenientes significativos: un complejo proceso de entrenamiento dividido en múltiples etapas (multi-pipeline), un alto coste computacional y de almacenamiento, y una notable lentitud tanto en el entrenamiento como en la inferencia.

Fast R-CNN

Para solucionar las ineficiencias de R-CNN, Girshick [11] desarrolló Fast R-CNN. La innovación fundamental de esta arquitectura fue procesar la imagen completa con una CNN una sola vez para generar un mapa de características compartido, a partir del cual se extraen las regiones de interés. Este enfoque eliminó la necesidad de pasar cada RoI por la CNN de forma independiente, reduciendo drásticamente la redundancia computacional.

Posteriormente, una capa de RoI Pooling extraía un vector de características de tamaño fijo para cada región, que era finalmente procesado por una red para la clasificación (usando Softmax) y el ajuste del bounding box.

Faster R-CNN

La principal limitación de Fast R-CNN era su dependencia de algoritmos externos lentos para la generación de RoIs. Para resolverlo, Ren et al. [12] introdujeron Faster R-CNN, que integra esta generación directamente en la red neuronal a través de la Region Proposal Network (RPN).

La RPN opera sobre el mapa de características convolucionales y predice un conjunto de regiones de interés con una puntuación de objectness score, que define la probabilidad de que esa región contenga un objeto. Al unificar la RPN con el detector Fast R-CNN, se logró por primera vez una arquitectura de detección de objetos verdaderamente end-to-end, mejorando sustancialmente tanto la velocidad como la precisión.

R-FCN

Aunque Faster R-CNN fue un gran avance, gran parte del cálculo seguía realizándose de forma independiente para cada RoI después de la capa de RoI Pooling. Para optimizar aún más la eficiencia, Dai et al. [13] propusieron las Region-based Fully Convolutional Networks (R-FCN). Esta arquitectura introduce un concepto clave: los mapas de puntuación sensibles a la posición (position-sensitive score maps). Estos mapas permiten codificar explícitamente la información espacial dentro de las características extraídas, lo que facilita que la red identifique partes específicas de los objetos. Gracias a esta estrategia, todas las capas convolucionales pueden ser compartidas en toda la red, lo que reduce significativamente la carga computacional.

Mask R-CNN

Para extender la detección de objetos a la segmentación de instancias (predecir un contorno a nivel de píxel para cada objeto), He et al. [14] desarrollaron Mask R-CNN. Esta arquitectura amplía Faster R-CNN añadiendo una rama paralela que predice una máscara de segmentación para cada RoI. Para ello, sustituye la capa de RoI Pooling por RoIAlign, que resuelve problemas de cuantización espacial y mejora significativamente la precisión de la máscara. Mask R-CNN se convirtió en un estándar ampliamente adoptado para tareas de segmentación de instancias debido a su alta precisión y flexibilidad.

A pesar de los avances en precisión y velocidad logrados por los detectores de dos etapas, su complejidad computacional seguía siendo una barrera importante para aplicaciones en tiempo real. Esta limitación impulsó el desarrollo de detectores de una sola etapa.

2.3.2. Detectores de una etapa

Las redes de detección de una sola etapa adoptan un enfoque más directo, realizando simultáneamente la localización y clasificación de objetos. Aunque este proceso unificado puede sacrificar cierta precisión en escenarios complejos, ofrece un rendimiento considerablemente más rápido [15].

La arquitectura YOLO [16], introducida en 2015, revolucionó este campo al procesar imágenes completas en una sola pasada para detectar múltiples categorías de objetos de forma simultánea. Esta eficiencia, combinada con su capacidad para mantener un alto rendimiento operando a velocidades apropiadas para el monitoreo en tiempo real, lo convierte en una herramienta particularmente relevante para el análisis de imágenes SAR.

Evolución de la familia YOLO

La arquitectura YOLO ha experimentado una notable evolución a lo largo de sus iteraciones. YOLOv2 perfeccionó el rendimiento con Batch Normalization y la optimización para resoluciones más altas. YOLOv3 avanzó con la introducción de skip connections y la detección multiescala, lo que incrementó notablemente la precisión. YOLOv4 reestructuró el modelo integrando componentes avanzados como el backbone CSPDarknet, el Spatial Pyramid Pooling (SPP), la Path Aggregation Network (PANet) y la función de activación Mish. YOLOv5 refinó

aún más la arquitectura, ofreciendo múltiples tamaños de modelo para equilibrar velocidad y precisión, y optimizó la detección basada en anclajes para una mayor eficiencia en el entrenamiento.

El cambio más significativo llegó con YOLOv8, que adoptó un enfoque completamente «anchor-free» (sin anclajes) para la detección. Esto significa que, en lugar de predecir ajustes a partir de cajas predefinidas, YOLOv8 predice directamente el centro y las dimensiones de los objetos. Esta innovación, junto con el uso de la Neural Architecture Search (NAS) para un diseño de red optimizado, ha simplificado el post-procesamiento y mejorado la eficiencia.

Como se muestra en la Figura 2.3, el modelo YOLOv8 se estructura en tres bloques principales:

- **Backbone**: actúa como el extractor de características principal. Basado en CSPDarknet, esta sección (ver área izquierda de la Figura 2.3) se encarga de la extracción robusta de características de la imagen de entrada. Utiliza Bloques Convolucionales (que combinan capas Conv2d, BatchNorm2d y la función de activación SiLU) y Bloques C2f. Los Bloques C2f dividen las características, pasando una parte a través de Bloques Bottleneck (que incorporan skip connections para un flujo de gradientes eficiente) y otra parte directamente a la concatenación. Al final del Backbone, el Bloque SPPF (Spatial Pyramid Pooling Fast) procesa las características a través de múltiples capas MaxPool2d para capturar información a diferentes escalas de manera eficiente, permitiendo que el modelo maneje objetos de distintos tamaños sin perder información espacial.

- **Neck**: funciona como una red de mejora de características (ver zona central de la Figura 2.3), empleando una estructura PANet modificada. Su función principal es fusionar de manera efectiva las características multiescala extraídas por el Backbone, permitiendo un flujo de información bidireccional entre los niveles superiores e inferiores. Este procesamiento bidireccional es crucial, ya que mejora tanto el detalle fino como la comprensión contextual de los objetos. El Neck utiliza capas de upsample para aumentar el tamaño de los mapas de características y Bloques Concat para combinar la información de diferentes resoluciones.

- **Detection Head**: sección final (a la derecha de la Figura 2.3) encargada de predecir los objetos detectados. A diferencia de las versiones anteriores, YOLOv8 utiliza un enfoque de basado en distribución para la predicción de las cajas finales (Distribution-based Bounding Box Prediction). Esto, junto con una función de pérdida que incorpora una estrategia basada en distribución, mejora significativamente la precisión de la localización de los objetos. El Detection Head tiene múltiples ramas, cada una especializada en la detección de objetos de un tamaño específico, recibiendo entradas del Neck.

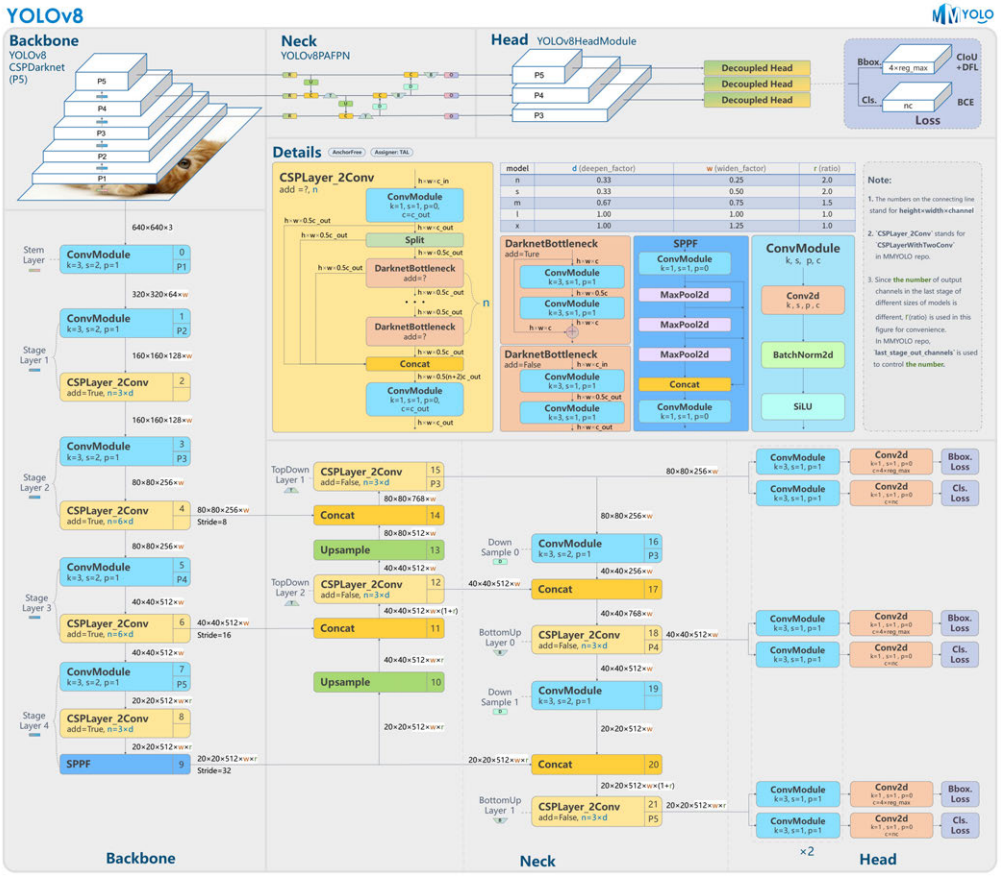


Figura 2.3: Arquitectura de YOLOV8 [17].

Dentro de la arquitectura de YOLOv8, la Non-Maximum Suppression (NMS) y la Intersection over Union (IoU) son conceptos fundamentales que se utilizan en el post-procesamiento de las detecciones del Detection Head. Aunque YOLOv8 es «anchor-free», sigue generando múltiples predicciones para los objetos.

- **IoU** (Intersection over Union): Esta métrica se utiliza para medir el solapamiento entre la caja delimitadora predicha por el modelo y la caja delimitadora real (ground truth) de un objeto. Un valor de IoU alto indica un buen solapamiento. Dentro de YOLOv8, la IoU se emplea para determinar qué predicciones son válidas y cuáles de ellas se solapan demasiado.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{2.4}$$

- **NMS** (Non-Maximum Suppression): Después de que el Detection Head genera múltiples cajas delimitadoras con sus respectivas puntuaciones de confianza para un mismo objeto (o para objetos que se solapan), la NMS se utiliza para filtrar estas predicciones redundantes. El proceso de NMS funciona seleccionando la caja con la mayor puntuación de confianza y luego eliminando cualquier otra caja que se solape significativamente con

ella (por encima de un umbral de IoU predefinido). Este proceso se repite hasta que no queden más cajas superpuestas, dejando solo la mejor predicción para cada objeto.

Basándose en los fundamentos de YOLOv8, las versiones posteriores introdujeron refinamientos dirigidos a mejorar el flujo de gradientes, la fusión de características y la eficiencia computacional. YOLOv9 incorporó Programmable Gradient Information (PGI) y Generalized Efficient Layer Aggregation Networks (GELAN). YOLOv10 eliminó la Non-Maximum Suppression (NMS) mediante una estrategia de asignación dual e implementó el Spatial-Channel Decoupled Downsampling para una arquitectura más refinada. YOLOv11 optimizó la atención espacial con los bloques C3k2 y el módulo C2PSA, mejorando la detección de objetos pequeños y superpuestos. YOLOv12 adoptó un enfoque centrado en la atención con Area Attention (A^2) para la agregación de características y Residual Efficient Layer Aggregation Networks (R-ELAN) para la estabilidad en la optimización. Sin embargo, a pesar de estas innovaciones, las evaluaciones comparativas han revelado que la complejidad añadida en estas últimas versiones a menudo no se traduce en ganancias de rendimiento sustanciales [18].

A pesar de los avances continuos, YOLOv8 sigue consolidándose como la arquitectura de referencia gracias a su rendimiento robusto y bien documentado y su amplia adopción en el campo. A continuación, se presentan distintos enfoques propuestos por la comunidad científica para adaptar la detección de objetos a las características particulares de las imágenes SAR.

La arquitectura de YOLO en el procesamiento de imágenes SAR

En 2024, Guo y Xu desarrollaron SAR-NTV-YOLOv8 [19]. Su propuesta incorporó la técnica de variación total no convexa (*non-convex total variation*) para eliminar el ruido speckle característico de las imágenes SAR, manteniendo al mismo tiempo los detalles importantes de la imagen. Además, incluyeron características especializadas para detectar objetivos pequeños y mecanismos de atención.

Durante 2024 se produjeron varios avances importantes de forma paralela. Fang y Wang propusieron la metodología FCCS-YOLO [20], que mejora YOLOv8 mediante aprendizaje contrastivo (*contrastive learning*). Por otro lado, una evaluación exhaustiva de diferentes variantes de YOLOv8 en escenarios aeroportuarios demostró el rendimiento superior de la configuración YOLOv8nx [21]. Adicionalmente, un estudio comparativo a gran escala utilizando los conjuntos de datos HRPlanesV2 y GDIT evaluó múltiples arquitecturas de detección, incluyendo YOLO v5/v8, Faster R-CNN, CenterNet, RetinaNet, RTMDet y DETR. Los resultados mostraron que las variantes YOLO exhibieron un rendimiento particularmente destacado en la detección de objetos aéreos [4].

La evolución de estos enfoques se basa en trabajo pionero anterior. En 2021, Guo, Wang y Xu presentaron su Red Piramidal de Atención Mejorada por Dispersión (Scattering-Enhanced Attention Pyramid Network) [22], que combina el realce de la información de dispersión con una red piramidal de atención (*attention pyramid network*). Esta propuesta logró resultados destacados en los conjuntos de datos Gaofen-3 y TerraSAR-X. Los antecedentes se remontan a 2018, cuando He y sus colaboradores fueron pioneros en el desarrollo de una

Red Paralela Multicapa basada en Componentes (Component-based Multilayer Parallel Network), demostrando mayor precisión en datos TerraSAR-X [23]. Posteriormente, en 2022 surgió el marco PFFADN [24], que se centra en la fusión de características (peak feature fusion) para mejorar la detección de objetivos.

La aparición de YOLOv8 como base para la detección de aeronaves en imágenes de Radar de Apertura Sintética se fundamenta en sus innovaciones arquitectónicas y sus métricas de rendimiento probadas.

Este proyecto se centra en evaluar el rendimiento de YOLOv8, ajustado a un conjunto de datos personalizado, por varias razones estratégicas. Este enfoque no solo permite establecer métricas de rendimiento fundamentales y aprovechar la eficiencia computacional del modelo, sino que también permite evaluar su adaptabilidad a condiciones SAR específicas. Adicionalmente, esta metodología proporciona información sobre posibles oportunidades de mejora, al tiempo que pone a prueba sus capacidades de generalización más allá de los conjuntos de datos estándar. En última instancia, esta investigación busca aportar una comprensión práctica de los sistemas de detección de vanguardia en aplicaciones SAR.

2.4. Generación de datos sintéticos

La generación de datos sintéticos se ha vuelto esencial para el avance de la inteligencia artificial (IA) y el machine learning (ML). Esta necesidad surge de la dificultad para obtener grandes volúmenes de datos del mundo real que sean diversos, representativos y etiquetados. En el ámbito de las imágenes SAR, esta problemática se intensifica debido a que la adquisición de datos es costosa, mientras que el etiquetado manual resulta laborioso y propenso a errores. Estas limitaciones restringen significativamente el tamaño y la diversidad de los conjuntos de datos disponibles.

Para comprender mejor las limitaciones actuales en el dominio de este proyecto, resulta fundamental examinar los datasets públicos existentes en el área de detección y clasificación de objetos SAR.

2.4.1. Datasets públicos

El dataset MSTAR [25] (Moving and Stationary Target Acquisition and Recognition), propuesto en los años 90, ha sido durante décadas el pilar fundamental en la investigación del Reconocimiento Automático de Objetivos (ATR) en SAR. Contiene 8,688 imágenes SAR de 7 tipos de vehículos terrestres y un objetivo de calibración, adquiridas con un sensor de banda X en modo spotlight con resolución espacial de 0.3 metros. Las imágenes se obtuvieron exclusivamente con polarización HH, presentando objetivos centrados en escenas de pasto bajo ángulos específicos (15°, 17°, 30° y 45°).

A pesar de su importancia histórica, MSTAR presenta limitaciones significativas. Su tamaño reducido, la falta de diversidad de muestras y las condiciones de adquisición idealizadas limitan la capacidad de generalización de los modelos a entornos realistas. Esta limitada diversificación de los datos se manifiesta claramente a nivel de píxel, como puede apreciarse en las muestras representativas

mostradas en la Figura 2.4, donde se evidencia la homogeneidad en las características espectrales y texturales de las imágenes del dataset.

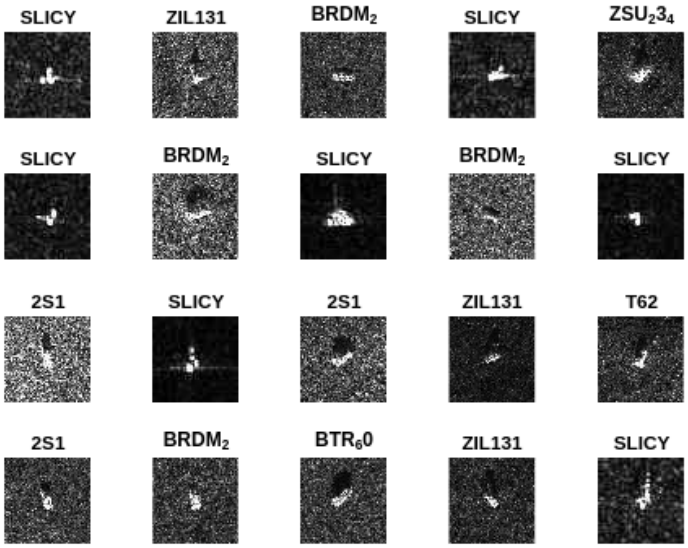


Figura 2.4: Ejemplos de imágenes del conjunto de datos MSTAR. Cada imagen SAR muestra un objetivo terrestre diferente y está etiquetada con su clase correspondiente (e.g., SLICY, ZIL131, BRDM2)[26].

Actualmente, un número mayoritario de los estudios SAR ATR siguen dependiendo de este dataset, evidenciando la escasez de alternativas disponibles [27].

Otros datasets existentes, como SAR-AIRcraft, Air-SARShip, SSDD y HRSID, presentan limitaciones similares al centrarse en un único tipo de objeto sobre fondos simplificados.

La introducción de SARDet-100K [28] en 2024 representa un avance significativo hacia la superación de estas limitaciones. Este dataset de referencia a gran escala fue creado mediante la recopilación y estandarización de 10 datasets SAR existentes. SARDet-100K alcanza una escala comparable al reconocido dataset COCO, abarcando aproximadamente 117,000 imágenes y 246,000 instancias distribuidas en seis categorías: Aeronaves, Buques, Coches, Puentes, Tanques y Puertos. Su diversidad se refuerza mediante imágenes de múltiples plataformas (GF-3, Sentinel-1B, TerraSAR-X, TanDEM-X, RadarSat-2, HISEA-1) y datos SAR aerotransportados. La resolución espacial varía jerárquicamente de 0.1 a 25 metros, incluyendo diversas bandas de frecuencia (C, X, Ka, Ku) y modos de polarización (HH, HV, VH, VV). Muestras de este dataset se presentan en la Figura 2.5.

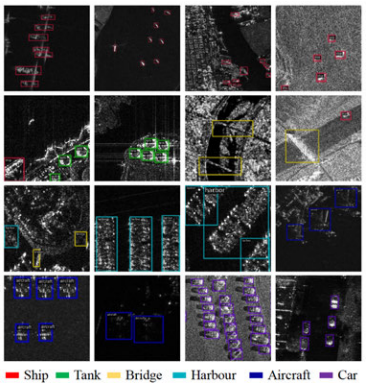


Figura 2.5: Ejemplos de imágenes del conjunto de datos SARDet100k. Cada clase queda anotada según color (rojo: barco, verde: tanque, amarillo: puente, cian: puerto, azul: aeronave, morado: coche) [28]

.

ATRNet-STAR [29] emerge como otra alternativa moderna al obsoleto MSTAR. Con más de 190,000 muestras anotadas de 40 categorías de vehículos, supera significativamente la escala de sus predecesores. Las imágenes fueron recolectadas bajo condiciones variadas en 5 escenas realistas, con alta resolución (0.12-0.15 metros) y soporte para cuadripolarización (HH, HV, VH, VV) en bandas X y Ku. Los objetos aparecen ubicados aleatoriamente y no centrados, simulando escenarios de detección más realistas.

Con todo esto, se busca evidenciar la limitada disponibilidad de datasets en este dominio específico. Aunque se han registrado nuevas contribuciones en los últimos dos años, la escasez persiste de manera notable. Esta limitación se acentúa considerablemente al enfocar el análisis en la aplicación concreta de este proyecto: la detección de aeronaves en entornos aeroportuarios y bases aéreas, donde los conjuntos de datos disponibles son aún más restringidos y presentan fondos de menor complejidad. La Figura 2.6 ilustra la evolución temporal de los datasets con imágenes SAR, reflejando esta problemática de disponibilidad de datos especializados.

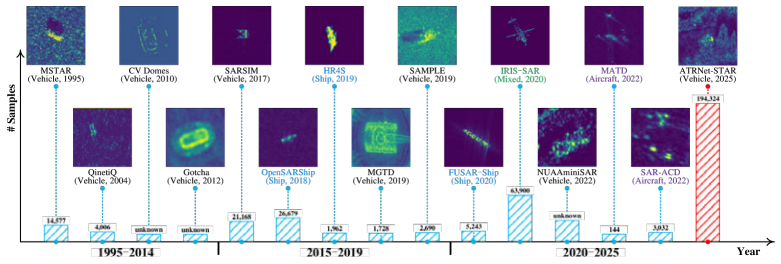


Figura 2.6: Comparativa y evolución de diversos conjuntos de datos de imágenes SAR para tareas de reconocimiento automático de objetivos, abarcando desde 1995 hasta 2025. La figura ilustra la progresión en la complejidad y el tamaño de los datasets a lo largo del tiempo, mostrando ejemplos visuales de las imágenes y el número de muestras para cada conjunto [29].

2.4.2. Modelos generativos

Los modelos de inteligencia artificial generativa han captado una atención significativa en los últimos años, consolidándose como herramientas esenciales para la síntesis de imágenes y videos.

A diferencia de los modelos discriminativos, que se enfocan en la clasificación o predicción de resultados a partir de datos existentes, los modelos generativos se centran en la creación. Los tres tipos principales de modelos generativos son las Redes Generativas Antagónicas (GANs), los Autoencoders Variacionales (VAEs) y los Modelos de Difusión.

Redes Generativas Antagónicas

Las Redes Generativas Antagónicas (GANs) operan bajo un principio de juego adversarial que involucra dos redes neuronales enfrentadas: un generador y un discriminador. El generador tiene como objetivo crear datos sintéticos, en este caso imágenes, a partir de un vector de ruido aleatorio, intentando que resulten indistinguibles de las muestras reales. Simultáneamente, el discriminador se entrena para clasificar si una imagen de entrada es auténtica o ha sido generada, buscando identificar con precisión el origen de cada muestra. Esta dinámica competitiva impulsa al generador a mejorar continuamente la calidad de sus salidas, mientras el discriminador afina su capacidad de detección.

La principal fortaleza de las GANs radica en su notable capacidad para generar datos de alta fidelidad visual, superando a menudo a otros modelos generativos. Una ventaja adicional es que, en su configuración básica, no requieren datos etiquetados para su entrenamiento.

No obstante, esta naturaleza adversarial también introduce desafíos significativos. El entrenamiento de GANs suele ser inestable y difícil de controlar, debido a la necesidad de mantener un equilibrio delicado entre la eficacia del generador y la del discriminador. Entre los problemas más comunes se encuentra el mode collapse, en el que el generador aprende a producir un conjunto limitado de muestras que engañan al discriminador, sin capturar la diversidad completa de la distribución de datos reales. Además, el entrenamiento efectivo de GANs tiende a requerir una gran cantidad de recursos computacionales.

Diversas arquitecturas derivadas han sido desarrolladas para abordar las particularidades de las imágenes de Radar de Apertura Sintética. Las Conditional GAN (cGAN), como Pix2Pix, se emplean en escenarios con imágenes emparejadas, permitiendo una traducción supervisada y controlada entre dominios [30]. Cuando no se dispone de pares exactos, CycleGAN es una alternativa eficaz al incorporar pérdidas cíclicas que preservan la coherencia estructural durante la conversión entre imágenes [31]. Por su parte, las Deep Convolutional GAN (DCGAN) [32] introducen capas convolucionales profundas en ambas redes, lo que contribuye a una mayor estabilidad y calidad visual.

En contextos más específicos, existen arquitecturas como DH-GAN [33], que incorpora discriminadores duales y mecanismos para reforzar detalles de alta frecuencia, mejorando la representación del ruido característico en imágenes SAR. En tareas de traducción entre dominios óptico-SAR, TSGAN [34] utiliza codificadores siameses que permiten capturar correspondencias espaciales y

semánticas entre ambas modalidades.

A pesar de estas capacidades, las GANs adaptadas al dominio SAR presentan limitaciones importantes, como la necesidad de conjuntos emparejados o largos procesos de preprocesamiento y un elevado coste computacional para lograr entrenamientos estables.

Por tanto, en un escenario de exploración inicial de imágenes SAR sintéticas, con recursos computacionales limitados y sin disponibilidad de datos emparejados, las GANs no representan una opción viable como punto de partida.

Autoencoders Variacionales

Los Autoencoders Variacionales (VAEs) adoptan una arquitectura de codificador - decodificador para la generación de datos. En este enfoque, un codificador (encoder) transforma los datos de entrada en un espacio latente, que representa una versión comprimida y probabilística de la distribución de los datos originales. Posteriormente, un decodificador (decoder) utiliza esta representación latente para reconstruir los datos. Una característica distintiva de los VAEs es su capacidad para mapear una misma imagen a múltiples valores dentro de una distribución de probabilidad, promoviendo así una alta diversidad en las muestras generadas.

Una de las principales ventajas de los VAEs es precisamente esa diversidad, ya que su entrenamiento está diseñado para capturar la distribución completa de los datos. Además, en comparación con las GANs y los modelos de difusión, los VAEs suelen ser más simples y menos costosos computacionalmente de entrenar.

Sin embargo, los VAEs presentan una limitación importante: suelen generar imágenes de baja fidelidad o con cierto grado de borrosidad. Este efecto se debe, en parte, a que sus funciones de pérdida están basadas en píxeles y al proceso de promediado inherente en el espacio latente durante la reconstrucción.

Dada la complejidad intrínseca de las imágenes SAR, los VAEs han encontrado aplicaciones principalmente en tareas como la extracción de características, la reconstrucción tridimensional y la eliminación de ruido.

Modelos de Difusión

Los Modelos de Difusión representan una de las innovaciones más recientes y prometedoras en el campo de la IA generativa para la síntesis de imágenes. Su enfoque se basa en un proceso de dos etapas para generar nuevos datos. Primero, se añade ruido gaussiano de manera incremental a un conjunto de datos hasta que las imágenes originales se transforman en ruido puro. Luego, el modelo aprende a revertir este proceso, eliminando gradualmente el ruido paso a paso para recuperar los datos originales o generar nuevas muestras que se asemejan a la distribución de los datos de entrenamiento.

La principal ventaja de estos modelos es su capacidad para generar muestras de muy alta fidelidad y gran diversidad, superando a menudo las limitaciones de las GANs en términos de estabilidad de entrenamiento y mode collapse. Ofrecen un control más preciso sobre el proceso de generación, lo que se traduce en una mayor versatilidad en las aplicaciones. Esta combinación de alta calidad y estabilidad los ha posicionado como el enfoque de vanguardia en el modelado generativo actual. Sin embargo, los Modelos de Difusión no están exentos de desafíos. Son computacionalmente intensivos y, por lo general, más lentos en la inferencia en comparación con las GANs y los VAEs, ya que requieren un gran número de iteraciones para el proceso de denoising.

En el ámbito de la teledetección, los modelos de difusión han comenzado a aplicarse con éxito a tareas complejas como la traducción de imágenes SAR a ópticas y la eliminación de nubes.

1. **Traducción de imágenes SAR a ópticas**

   Varios trabajos recientes han explorado el uso de modelos de difusión para la traducción de imágenes SAR a ópticas, con el fin de superar las limitaciones de enfoques tradicionales basados en GANs.

   Uno de los avances iniciales fue propuesto por Bai et al. (2024) [35], quienes desarrollaron un modelo de difusión condicionado con supervisión de color para guiar el proceso de traducción. En la misma línea, Bai et al. (2024) [36] propusieron una estrategia para acelerar el proceso de generación mediante distilación de consistencia adversarial (adversarial consistency distillation), alcanzando velocidades hasta 131 veces superiores sin sacrificar precisión ni introducir artefactos visuales.

   Otro enfoque destacable es C-DiffSET [37], basado en modelos de difusión latente (LDMs) y guiado por un mecanismo de confianza que penaliza regiones poco fiables, reforzando la calidad semántica y la integridad estructural de las imágenes generadas. Por su parte, Aydin et al. (2025) [38] exploraron la variante de cold diffusion para traducir imágenes SAR a RGB. Aunque esta técnica produce imágenes con menor fidelidad visual, resultó competitiva en tareas automáticas de clasificación del uso del suelo.

   Un modelo particularmente innovador fue propuesto por Kim et al. (2025) [39], introduciendo un marco basado en puentes brownianos condicionales (conditional brownian bridge diffusion), orientado a imágenes SAR de muy alta resolución (0.5m). Este método define una interpolación probabilística entre los dominios SAR y óptico, guiada por información semántica contextual, logrando preservar detalles finos en entornos urbanos complejos. De manera complementaria, Shi et al. (2024) [40] desarrollaron un modelo que integra atención multiescala y data augmentation adaptativo, permitiendo una traducción más precisa sin pérdida de estructura ni desenfoque.

2. **Eliminación de nubes en imágenes ópticas mediante datos SAR**

   Además de la traducción intermodal, los modelos de difusión también han demostrado gran eficacia en la tarea de eliminación de nubes en imágenes ópticas, al combinar datos ópticos con información proveniente de sensores SAR.

   Uno de los enfoques más innovadores es DC4CR (Diffusion Control for Cloud Removal) [41], propuesto por Yu et al. (2025). Este marco aprovecha modelos de difusión texto-a-imagen como Stable Diffusion, adaptados con técnicas como LoRA[1] y control mediante prompts, lo que permite la eliminación selectiva de nubes finas y gruesas sin necesidad de máscaras pregeneradas.

   Complementariamente, Hu et al. (2025) propusieron Diffusion Bridges for Cloud Removal (DB-CR) [42], un modelo de puente de difusión multimodal que incorpora dos ramas paralelas (SAR y óptico) y bloques de atención transmodal. Esta arquitectura permite fusionar eficazmente las características complementarias de ambas modalidades, mejorando la estabilidad del entrenamiento y la calidad de las reconstrucciones.

   En la misma línea, Zhang et al. (2025) introdujeron DMDiff [43], un modelo de difusión multirama condicionado, con políticas de atención adaptativa y entrenamiento progresivo, que mejora la consistencia espectral y espacial durante la eliminación de nubes.

La generación directa de imágenes SAR a partir de texto o prompts es aún un área emergente. Sin embargo, técnicas como la adaptación LoRA aplicada a modelos preentrenados de difusión texto-imagen, como Stable Diffusion y Flux, abren la puerta a explorar flujos de trabajo multimodales y generación sintética controlada, lo que puede revolucionar la creación de datasets sintéticos para entrenamiento y evaluación.

Dado este panorama, el presente proyecto se centra en una aproximación exploratoria al entrenamiento y adaptación de modelos LoRA sobre arquitecturas de difusión como Stable Diffusion y Flux, evaluando su potencial para la generación sintética de imágenes SAR y la integración en workflows multimodales (img2img y text2img). Esta línea inicial busca sentar las bases para futuras investigaciones que utilicen estas imágenes generadas para mejorar el entrenamiento de detectores basados en YOLO y otros modelos discriminativos.

[1] LoRA (Low-Rank Adaptation) permite adaptar modelos grandes añadiendo matrices entrenables de bajo rango, reduciendo costes sin modificar todos los parámetros.

Capítulo 3

Metodología

La metodología empleada en este proyecto involucra tres fases. La primera etapa se centra en la detección de aeronaves y constituye una de las fases más detalladas del proyecto. Se llevó a cabo una experimentación extensiva para identificar las combinaciones más efectivas de learning rates, tamaños de imagen de entrada, arquitecturas y otros hiperparámetros, con el objetivo de ajustar el modelo a las características particulares del dataset.

La segunda etapa se enfoca en la clasificación multiclase de aeronaves, utilizando como punto de partida las configuraciones optimizadas de la fase anterior. Se llevaron a cabo nuevos experimentos para evaluar la aplicabilidad de estas configuraciones a la tarea de clasificación, permitiendo evaluar la versatilidad del modelo para abordar diferentes dominios de problemas.

La tercera fase constituye una línea de trabajo completamente independiente, tanto en objetivos como en entorno técnico. El proyecto se transforma desde un enfoque de análisis supervisado hacia la generación sintética de imágenes SAR, reestructurando el conjunto de datos original y adaptándolo a los requerimientos de los modelos generativos.

Se estableció un nuevo entorno de entrenamiento basado en modelos de difusión entrenados mediante LoRAs (Low-Rank Adapters). Se llevaron a cabo múltiples pruebas experimentales para explorar diversas capacidades: generación de texto a imagen (text2img), transformación guiada de imagen a imagen (img2img), y generación multimodal mediante controladores como ControlNet e IPAdapter.

La preparación de los datos implicó un rediseño completo del pipeline de entrenamiento, experimentando con variantes enriquecidas y simplificadas del dataset original. El objetivo fue explorar el nivel de realismo inicial alcanzable y analizar su utilidad futura como datos de entrenamiento para tareas discriminativas.

3.1. Datos

3.1.1. Especificaciones técnicas de las imágenes utilizadas

El dataset empleado en este estudio es una colección propietaria, preparada y etiquetada para tareas de detección de aeronaves. Fue proporcionado por HISDESAT Servicios Estratégicos, S.A., y capturado por el satélite PAZ [44].

El satélite PAZ orbita a una altitud de 514 kilómetros, completando 15 órbitas diarias a una velocidad aproximada de 7 km/s. Gracias a su órbita cuasi-polar ligeramente inclinada, PAZ ofrece cobertura global con un tiempo de revisita medio de 24 horas. Es capaz de capturar más de 100 imágenes diarias con una resolución de hasta 25 cm. Con una capacidad de cobertura superior a 300,000 kilómetros cuadrados por día, el satélite resulta fundamental para aplicaciones como el control fronterizo, el monitoreo ambiental, el desarrollo de infraestructuras, la gestión de desastres y la cartografía de alta resolución.

PAZ está equipado con un sistema SAR de banda X, que opera a una longitud de onda de 3 cm. Este sistema incorpora una antena activa tipo phased array, que permite dirigir electrónicamente el haz de radiofrecuencia sin necesidad de movimiento mecánico. Su sistema SAR soporta múltiples modos de adquisición, incluyendo Spotlight, HR Spotlight, Staring Spotlight, StripMap, ScanSAR y Wide ScanSAR, con opciones de polarización HH, HV, VV y VH. Opera con un ancho de banda de hasta 300 MHz, lo que permite la obtención de imágenes de muy alta resolución (VHR).

Para este estudio, se utilizó el modo Staring Spotlight SSC (Single Look Slant Range Complex product), que optimiza el tiempo de iluminación en azimut ajustando el punto de rotación virtual al centro del haz. Esto permite mejorar significativamente la resolución azimutal, generando imágenes con un nivel de detalle excepcional.

La resolución muy alta de este dataset es crucial para la detección de objetivos pequeños y estructuralmente complejos, un nivel de detalle que no es alcanzable con sensores como Sentinel-1, cuya resolución es del orden de los 10 metros. Las principales características operacionales del satélite PAZ se resumen en la Tabla 3.1.

Tabla 3.1: Características del modo Staring Spotlight

Parámetro	**Valor**
Proyección en tierra (ground range)	9 – 4.6 km
Longitud nominal del producto	2.7 – 3.6 km
Rango de ángulos de incidencia a pleno rendimiento	20° – 45°
Rango de ángulos de incidencia accesibles	15° – 60°
Resolución (range × azimuth)	0.60 × 0.26 m
Polarizaciones	HH, VV, HV, VH

3.1.2. Preprocesamiento de imágenes en bruto

Se aplicaron varios pasos de preprocesamiento para asegurar la consistencia espacial y reducir la carga computacional:

- **Ajuste de píxel** (Pixel Aspect Ratio): La resolución en azimut era aproximadamente tres veces mayor que la resolución en rango. Para corregir esta discrepancia, los valores azimutales fueron promediados en bloques de 2 a 3 píxeles, obteniendo así un aspecto de píxel cercano al cuadrado, que representa de forma más fiel la realidad sobre el terreno.
- **Mejora radiométrica**: Los valores radiométricos originales de 16 bits fueron reescalados a un rango de 8 bits, con el objetivo de mejorar la eficiencia computacional sin comprometer significativamente la interpretabilidad visual [45].

3.1.3. Descripción del dataset

El conjunto de datos está compuesto por un total de 132 imágenes SAR de dimensiones variables, adquiridas con una resolución espacial uniforme de 96 píxeles por pulgada (ppi) y una profundidad radiométrica de 8 bits por píxel. Estas imágenes fueron capturadas por el satélite PAZ sobre diversas bases aéreas en distintos momentos temporales, reflejando así una amplia variedad de características espaciales y ambientales.

Cabe destacar que las imágenes no fueron sometidas a técnicas tradicionales de aumentación de datos (como rotaciones, flips, o escalados aleatorios), debido a que dichas transformaciones podrían alterar la firma radar de los objetos.

No obstante, el tamaño original de las imágenes dificultaba su uso directo en el entrenamiento. Redimensionarlas para ajustarlas al tamaño de entrada del modelo YOLO provocaría que las aeronaves resultasen prácticamente invisibles, afectando negativamente el rendimiento del sistema. Para resolver este inconveniente, se recurrió a un enfoque de sliding window, dividiendo cada imagen en recortes de 100×100 píxeles (ver Figura 3.1). Esta dimensión fue elegida como compromiso entre la visibilidad del objetivo y la eficiencia computacional.

Para evitar que las aeronaves quedaran fragmentadas entre varios recortes, se introdujo superposición entre ventanas adyacentes, generando una estructura de celdas parcialmente solapadas. Un recorte era retenido únicamente si al menos el 75 % del área de una aeronave se encontraba dentro de sus límites.

En los recortes seleccionados, las coordenadas de los bounding boxes fueron recalculadas respecto al sistema de referencia local del recorte. Posteriormente, se transformaron al formato requerido por YOLO, con coordenadas normalizadas entre 0 y 1 tanto para el centro del objeto (x, y) como para sus dimensiones (ancho y alto). Cada recorte con presencia de aeronaves generó un archivo .txt con el mismo nombre que la imagen, en el que se registraban el ID de clase, el centro y las dimensiones del bounding box.

Figura 3.1: Extracción de recortes de imágenes SAR mediante una ventana deslizante. La cuadrícula roja representa ventanas de 100×100 píxeles con un solapamiento del 50 %, y se muestran cinco recortes representativos resaltados en verde para demostrar la metodología de muestreo. (PAZ Satellite@Hisdesat Servicios Estratégicos)

División y estructura para la detección y clasificación

Para las dos primeras fases del proyecto, el conjunto de datos se dividió en subconjuntos de entrenamiento, validación y prueba en una proporción de 80:10:10. Esta división dio lugar a 6.824 recortes para entrenamiento, 852 para validación y 853 para prueba, incluyendo tanto instancias positivas (con presencia de aeronaves) como negativas (recortes de fondo sin objetivos). En total, se anotaron 6.563 instancias de aeronaves, clasificadas en cinco categorías principales: 0: Fighter, 1: Helicopter, 2: Transport, 3: Bomber, y 4: Other. Estas clases quedan subdivididas para el segundo problema de clasificación según modelos concretos dentro de cada tipo de aeronave. Esta subdivisión se aclarará en apartados consiguientes.

El etiquetado de las instancias se adaptó según el objetivo de cada tarea:

- **Detección de aeronaves**: formulada como un problema de clasificación binaria que distingue entre aeronaves y fondo.
- **Clasificación de tipos de aeronaves**: planteada como un problema de clasificación multiclase que abarca las clases de aeronaves mencionadas más una clase adicional de fondo.

Dado que los modelos fueron entrenados de forma independiente, ambas fases comparten la misma estructura y distribución de subconjuntos, lo que permite aprovechar la totalidad de los datos en cada entrenamiento.

Adaptación del conjunto de datos para la generación de imágenes

En la tercera fase del proyecto, centrada en la generación sintética de imágenes, se trabajó con un subconjunto reducido del dataset original. Con el fin de mitigar la alta dimensionalidad y la variabilidad intrínseca de los datos SAR, se restringió el estudio a una única base aérea y un solo tipo de aeronave. Esta elección responde a dos motivaciones principales:

- **Control experimental del dominio de entrada**: reducir la variabilidad geográfica y de clases permite entrenar el modelo bajo condiciones controladas, facilitando la convergencia y mejorando la coherencia de las imágenes generadas. Al centrarse en un solo tipo de aeronave, el modelo puede aprender en mayor profundidad las características específicas de su firma radar, evitando la confusión derivada de tratar múltiples clases simultáneamente.
- **Evaluación de robustez en escenarios realistas**: aunque se limita a una única localización, las imágenes seleccionadas presentan suficiente variabilidad en términos de orientación, posición relativa, condiciones de iluminación y firmas radar.

El enfoque de generación se centró exclusivamente en recortes que contenían aeronaves, omitiendo la generación de entornos sin objetivos. Esta decisión responde a la necesidad de abordar toda la complejidad de la tarea: por un lado, modelar la variabilidad espacial del contexto en el que se encuentra una aeronave; por otro, capturar la variabilidad intrínseca de su firma radar bajo distintas condiciones ambientales y temporales.

Tras aplicar estos criterios de filtrado, se obtuvo un subconjunto final de 1,009 recortes, todos pertenecientes al mismo tipo de aeronave y base aérea. A pesar de su tamaño reducido, este conjunto es adecuado para la adaptación de modelos de difusión preentrenados mediante técnicas como LoRA, que permiten una adaptación efectiva con cantidades moderadas de datos.

En cuanto al etiquetado, se optó por una estrategia de captioning mínima. Si bien los modelos de difusión permiten la generación condicionada mediante descripciones detalladas (*prompt engineering*), se decidió prescindir de etiquetas individualizadas por imagen para esta fase inicial, con el fin de reducir la complejidad semántica del entrenamiento y evitar introducir ruido sintáctico en el modelo.

Durante el filtrado y la construcción del subconjunto, las anotaciones fueron utilizadas exclusivamente para identificar recortes válidos con presencia de aeronaves. Posteriormente, durante el entrenamiento, se empleó un único caption fijo o trigger para todos los recortes. Esta decisión responde a un enfoque progresivo: empezar con un entrenamiento simplificado y controlar el comportamiento del modelo antes de incorporar conditioning más avanzado en futuras iteraciones.

3.2. Experimentos

3.2.1. Modelado con YOLO

Configuración experimental

El entrenamiento de la red y la evaluación de su desempeño se realizaron en una GPU NVIDIA RTX 3090 Ti con 24 GB de memoria GDDR6, utilizando el framework PyTorch en su versión 2.2.2. El proceso se configuró con un `batch_size` de 16 y 8 `workers` para la carga y procesamiento paralelo de datos. Se planificaron hasta 100 `epochs`, empleando un umbral de patience para detener el entrenamiento una vez alcanzada la curva óptima de aprendizaje.

Para la optimización, se utilizó el optimizador AdamW por defecto de Ultralytics YOLOv8 [46]. Se evaluaron dos estrategias de tasa de aprendizaje (`lr0` y `lrf`). En la primera configuración, se empleó una tasa inicial (`lr0`) de 0.01 y una final (`lrf`) de 0.1, lo que implica una reducción progresiva hasta 0.001, ajustada automáticamente por el planificador interno del framework. En la segunda estrategia, se utilizó una tasa inicial más conservadora de 0.001 con una final de 0.01, lo que redujo la tasa de aprendizaje hasta un mínimo de 1e-5.

Los parámetros de optimización incluyeron un `momentum` de 0.937 y un `weight_decay` de 0.0005. Además, se aplicó una fase de warm-up durante los tres primeros epochs, iniciando con un momentum de 0.8 y una tasa de aprendizaje (*bias learning rate*) de 0.1, con el objetivo de facilitar una convergencia inicial más estable.

Para validar la robustez del modelo, se realizaron experimentos con imágenes de tamaño variable: 256, 320 y 416 píxeles.

Las configuraciones detalladas de los hiperparámetros, las versiones de software utilizadas y las especificaciones del hardware se presentan de forma exhaustiva en las Tablas 3.2 y 3.3.

Tabla 3.2: Configuración del entorno

Configuración de software	
Nombre	**Versión**
Ubuntu	22.04.4 LTS
Python	3.12.10
PyTorch	2.7.0
CUDA	12.6
Configuración de hardware	
Componente	**Especificaciones técnicas**
GPU	NVIDIA RTX 3090 Ti, 24 GB GDDR6X
CPU	Intel(R) Core(TM) i7-4790K CPU @ 4.00GHz
Memoria RAM	32 GB

Tabla 3.3: Configuración de hiperparámetros

Parámetro	**Valor**
Framework	Ultralytics YOLOv8
Optimizador	AdamW
Estrategia de learning rate 1	0.01, reducida cíclicamente hasta 0.001
Estrategia de learning rate 2	0.001, reducida hasta 0.00001
Momentum	0.937
Weight Decay	0.0005
Tamaño del batch	16
Número de workers	8
Número total de epochs	100
Paciencia (patience)	100
Epochs de warm-up	3
Momentum inicial de warm-up	0.8
Bias learning rate	0.1
Tamaños de imagen evaluados	256, 320, 416

Métricas de evaluación

El rendimiento de los modelos será evaluado utilizando métricas estándar de aprendizaje profundo: Accuracy, Precision, Recall y F1-Score. Antes de definir las métricas, es fundamental comprender los siguientes conceptos:

- **Verdaderos Positivos (TP)**: instancias positivas correctamente clasificadas (real = 1, predicho = 1)
- **Verdaderos Negativos (TN)**: instancias negativas correctamente clasificadas (real = 0, predicho = 0)
- **Falsos Positivos (FP)**: instancias negativas incorrectamente clasificadas como positivas (real = 0, predicho = 1)
- **Falsos Negativos (FN)**: instancias positivas incorrectamente clasificadas como negativas (real = 1, predicho = 0)

1. **Accuracy**

   Ofrece una visión general del rendimiento del modelo calculando la proporción de predicciones correctas sobre el conjunto completo de datos. Sin embargo, en escenarios con desequilibrio entre clases o tareas de clasificación complejas, el nivel de detalle que ofrece el accuracy resulta insuficiente por sí solo.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

2. **Precision**

   Mide la proporción de predicciones positivas que son correctas. Esta métrica enfatiza la fiabilidad de las predicciones positivas, algo especialmente importante en la detección de aviones, debido a las posibles consecuencias de las falsas detecciones.

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

3. **Recall**

   Mide la proporción de instancias positivas reales que fueron correctamente identificadas. Un alto valor indica que el modelo identifica correctamente la mayoría de los casos positivos, aunque no considera las falsas alarmas.

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

4. **F1-Score**

   Combina precision y recall en una sola métrica mediante su media armónica.

$$F1{-}Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3.4}$$

Configuración del modelo

YOLOv8, desarrollado por Ultralytics y lanzado en enero de 2023, ofrece cinco variantes del modelo con tamaños y complejidades crecientes: v8n (nano), v8s (small), v8m (medium), v8l (large) y v8x (extra large). A medida que aumenta el tamaño del modelo, suele mejorar la precisión, aunque esto conlleva un mayor coste computacional y tiempos de inferencia más altos.

Todos los modelos YOLOv8 vienen preentrenados con el conjunto de datos COCO (Common Objects in Context) de Microsoft, que incluye aproximadamente 330.000 imágenes etiquetadas distribuidas en 80 clases de objetos.

El uso de aprendizaje por transferencia a partir de estos modelos preentrenados resulta especialmente ventajoso en este estudio, ya que entrenar YOLOv8 desde cero requeriría un volumen de datos significativamente mayor al disponible.

La elección de la variante del modelo constituye un paso inicial clave, ya que impacta directamente en el rendimiento del sistema y en la eficiencia computacional del pipeline de detección. Modelos más grandes tienden a ofrecer mayor precisión, pero a costa de tiempos más largos de entrenamiento e inferencia.

En los experimentos iniciales se utilizaron las dos versiones extremas, la más pequeña (v8n) y la más grande (v8x), con el objetivo de explorar si las posibles mejoras en precisión con v8x justificaban la pérdida de velocidad de inferencia respecto a v8n.

Para este proyecto, la implementación estándar de YOLOv8 proporcionada por Ultralytics fue suficiente para alcanzar los resultados esperados, sin necesidad de modificar su arquitectura interna. No obstante, fue necesario realizar un fine-tuning de los pesos en función del conjunto de datos específico empleado para el entrenamiento.

Dado que el tamaño del modelo afecta directamente la viabilidad del sistema en términos de precisión, velocidad de inferencia y posibilidad de despliegue práctico, esta variable se estableció como un factor clave a evaluar.

Antes de abordar los detalles relativos al diseño experimental, se presenta a continuación (Figura 3.2) una visión general del pipeline desarrollado para la detección y clasificación de aeronaves en las imágenes SAR. Este flujo integra las etapas de preprocesamiento, segmentación en recortes e inferencia mediante YOLOv8, ejecutándose de forma secuencial con cualquier imagen una vez completado el entrenamiento del modelo.

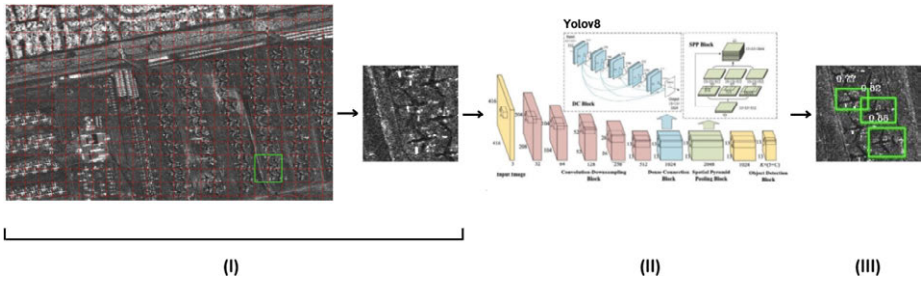


Figura 3.2: Esquema del pipeline propuesto para la detección/clasificación de aeronaves en imágenes de radar de apertura sintética (SAR). El proceso consta de: (i) segmentación de la imagen SAR en recortes solapados; (ii) procesamiento de cada recorte mediante el modelo de detección de objetos YOLOv8; y (iii) agregación de las detecciones sobre la imagen completa.

Detección de aeronaves

La exploración se centró en dos aspectos principales: los hiperparámetros que influyen directamente en el rendimiento del modelo, como `imgsz` (tamaño de imagen) y `lr` (tasa de aprendizaje), y los parámetros personalizados específicos del conjunto de datos, que afectan a la distribución de recortes, como `no_label` y `n_urban`. El parámetro no_label controla la proporción de recortes sin aeronaves etiquetadas, mientras que n_urban determina la fracción de estos recortes que corresponden a zonas urbanas. Estos parámetros se ajustaron de forma sistemática para evaluar su impacto en el rendimiento del modelo final.

El ajuste de estos se automatizó mediante un script que recorría todas las combinaciones posibles, sin necesidad de herramientas externas de optimización de hiperparámetros. El parámetro `imgsz` es clave en YOLOv8, ya que define las dimensiones a las que se redimensionan las imágenes antes de ser introducidas en el modelo. Aunque el valor por defecto en YOLO es 640, se probaron tamaños menores para evaluar su efecto. Por su parte, la tasa de aprendizaje (`lr`) se exploró a través de dos componentes: la tasa inicial (`lr0`), que define el tamaño del paso en la actualización de los pesos al inicio del entrenamiento, y el factor de reducción (`lrf`), que disminuye gradualmente la tasa desde lr0 hasta una fracción de su valor original (`lr0 * lrf`). Se probaron dos configuraciones: (`lr0` = 0.01, `lrf` = 0.001) y (`lr0` = 0.001, `lrf` = 0.0001).

Además, se ajustó sistemáticamente la composición del conjunto de datos para examinar su influencia en el entrenamiento. El parámetro `no_label` se varió de 0 a 1.5, controlando la proporción de recortes sin aeronaves. Por ejemplo, un valor de 1 garantiza un número igual de recortes con y sin aeronaves, mientras que valores superiores incrementan la presencia de recortes de fondo. Esto permitió explorar cómo afecta al modelo la incorporación de una mayor cantidad de recortes de fondo. Por otro lado, `n_urban` controló la fracción de recortes de fondo que correspondían a áreas urbanas, las cuales suelen presentar características visuales complejas y ambiguas que dificultan la detección. Al variar `n_urban` entre 0.0 y 0.5, se evaluó cómo influía la presencia de imágenes urbanas en el rendimiento final del modelo y su robustez ante estos entornos.

En cuanto a la preparación del dataset para entrenamiento, prueba y validación, inicialmente se contaba con 6.563 recortes con aeronaves. Estos se mezclaron aleatoriamente y se distribuyeron en las tres particiones (entrenamiento, prueba y validación). Una vez realizada esta división, se aplicaron los parámetros `no_label` y `n_urban` para determinar la cantidad de recortes de fondo a añadir en cada subconjunto.

Clasificación multiclase

A partir de los resultados obtenidos en la fase de detección, esta etapa del estudio se centró en la tarea de clasificación. Para cada caso, las etiquetas del conjunto de datos se adaptaron al objetivo específico. En la fase de detección, como se explicó previamente, las clases 0 a 4 se agruparon en una única categoría que indicaba la presencia de una aeronave, diferenciando únicamente entre dos clases: clase 0 (aeronave) y clase 1 (fondo).

En cambio, para la tarea de clasificación, las instancias correspondientes a aeronaves se reorganizaron inicialmente en cinco categorías: 0 (Fighter), 1 (Helicopter), 2 (Transport), 3 (Bomber) y 4 (Other), transformando el problema en una clasificación de seis clases al incluir el fondo como clase 5. Posteriormente, se exploró una clasificación más detallada con diez tipos de aeronaves: 0 (Fighter Sukhoi), 1 (Fighter other), 2 (Helicopter), 3 (Transport An-12), 4 (Transport other), 5 (Bomber Tu-22), 6 (Bomber Tu-95), 7 (Bomber Tu-160), 8 (Bomber other) y 9 (Other), sumando un total de once clases al incluir el fondo.

Los experimentos de clasificación reutilizaron las configuraciones e hiperparámetros que habían mostrado buen desempeño en la fase de detección. Este enfoque ofreció dos ventajas principales: por un lado, redujo el esfuerzo de reconfiguración y, por otro, permitió analizar la capacidad del modelo para adaptarse a distintos dominios de aplicación. Además de evaluar la adaptabilidad general, se prestó especial atención al impacto del desequilibrio entre clases en el rendimiento del modelo.

Tal como se observa en las Figuras 3.3 y 3.4, el conjunto de datos presenta un claro desbalance. En particular, según la Figura 3.3, la clase 3 es la más abundante, seguida de las clases 0 y 4, mientras que las clases 1 y 2 están notablemente subrepresentadas. Este desequilibrio se acentúa aún más en la Figura 3.4, donde la clasificación se detalla por modelo de aeronave. Esta distribución desigual introduce sesgos potenciales en el modelo, lo que puede incrementar significativamente la tasa de error en las clases con menor representación y afectar la capacidad de generalización del sistema.

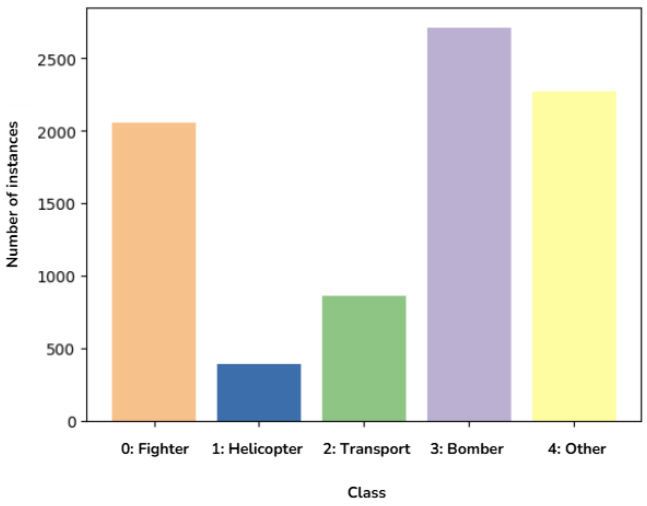


Figura 3.3: Distribución de instancias de aeronaves por clase en el conjunto de datos. Los tipos de aeronaves se clasifican en cinco categorías. El eje X representa las clases de aeronaves, mientras que el eje Y indica el número de instancias por clase.

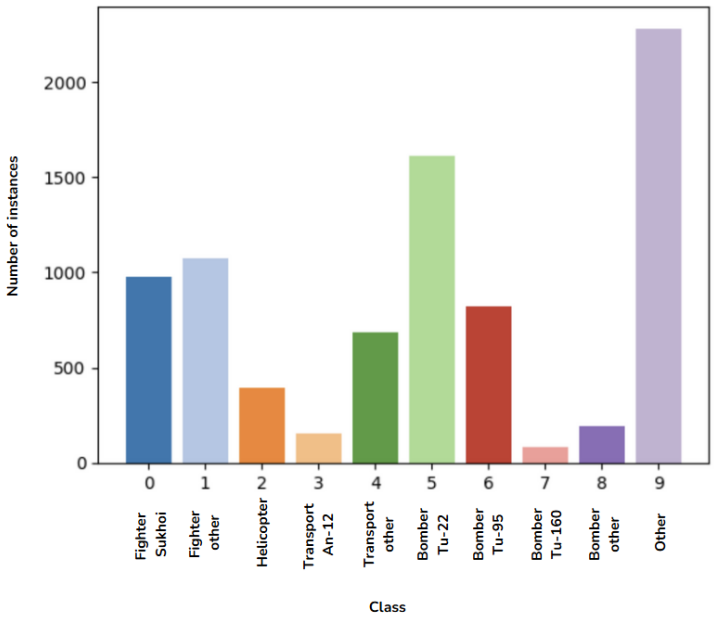


Figura 3.4: Distribución de instancias de aeronaves por clase extendidas en el conjunto de datos. Los tipos de aeronaves se clasifican en diez tipos de aeronaves. El eje X representa las clases de aeronaves, mientras que el eje Y indica el número de instancias por clase.

3.2.2. Generación de imágenes SAR sintéticas

Los experimentos de generación se han desarrollado bajo las mismas especificaciones de hardware y software utilizadas en fases anteriores del trabajo (ver Tabla 3.2). No obstante, se introducen diferencias importantes en cuanto a los entornos y dependencias, ya que esta fase incorpora distintos frameworks adaptados a los modelos de difusión empleados. En concreto, los entrenamientos se han centrado en modelos LoRA sobre dos arquitecturas principales: Flux 1 y Stable Diffusion 3.5 (SD3.5).

Para los entrenamientos iniciales, cada modelo ha requerido un entorno de desarrollo específico. En el caso de Flux 1, se ha utilizado el entorno visual ComfyUI [47], mientras que para SD3.5 se ha empleado el entorno ai-toolkit de Ostris [48], cada uno con sus propias dependencias de instalación y workflows particulares de entrenamiento.

A continuación, se describen las condiciones y configuraciones utilizadas en cada uno de los modelos.

Entrenamiento de LoRAs con Flux 1

- **Dataset y condiciones de entrada**

  Se trabajó con un subconjunto de 1009 recortes de 100×100 píxeles, todos pertenecientes a la misma clase de aeronave y base aérea. El captioning aplicado fue mínimo, limitándose a un trigger «radarize», utilizado pos-

teriormente para el condicionamiento por texto. No se aplicaron técnicas de data augmentation (sin `color_aug, flip_aug, shuffle_caption` ni `caption_dropout_rate`), y se mantuvo una única repetición por muestra (`num_repeats` = 1).

La principal variabilidad introducida en esta fase fue doble: por un lado, se exploraron distintas resoluciones de entrada (100×100 y 512×512, esta última tras un redimensionamiento de los recortes originales); por otro, se evaluó el efecto del tamaño del subconjunto de entrenamiento, utilizando la totalidad de los 1009 recortes así como subconjuntos aleatorios de 50 y 500 imágenes (ver Tabla 3.4).

Tabla 3.4: Parámetros de prueba para el conjunto de entrenamiento

Variable	**Valores explorados**
Resolución de entrada	100×100, 512×512
Tamaño del conjunto de entrenamiento	50, 500 , 1009

El tamaño de entrada es un factor clave en modelos de difusión, ya que determina la granularidad y fidelidad de los patrones espaciales aprendidos.

- **Parámetros del modelo y entrenamiento**

  Todos los entrenamientos con Flux 1 utilizaron los siguientes componentes estables del pipeline (ver Tabla 3.5):

Tabla 3.5: Configuración del modelo Flux LoRA

Parámetro	**Valor**
Modelo base	flux1-dev-fp8
Autoencoder (VAE)	ae
Modelo CLIP	clip_l
Modelo T5	t5ccl_fp8_e4m3fn
`network_dim`	16
`network_alpha`	1.00
`learning_rate`	0.0004
`timestep_sampling`	shift
`discrete_flow_shift`	3.1582
`gradient_dtype`	bf16

Estos parámetros han sido seleccionados por su compatibilidad y buen rendimiento observado en entrenamientos previos. Por ejemplo, el uso de bf16 permite optimizar el uso de memoria sin afectar negativamente a la estabilidad numérica del entrenamiento.

La única variable que se ha modificado entre experimentos ha sido el número de epochs, probándose configuraciones de 1600 y 3000 iteraciones completas sobre el dataset.

Entrenamiento de LoRAs con Stable Diffusion 3.5

- **Dataset y condiciones de entrada**

  En este caso se ha reutilizado el mismo subconjunto de recortes, aplicando el mismo trigger («radarize») y una única repetición por muestra. Las resoluciones de entrada y los tamaños de subconjunto empleados han sido idénticos a los utilizados con Flux 1, permitiendo así una comparación más justa entre modelos.

  La principal diferencia respecto al modelo Flux ha sido la aplicación de un `drop_caption_rate` = 0.05, siguiendo las recomendaciones del entorno ai-toolkit. Esta técnica introduce cierta aleatoriedad en la presencia del caption durante el entrenamiento.

- **Parámetros del modelo y entrenamiento**

  Dado que el entrenamiento de LoRAs sobre SD3.5 es menos frecuente en la comunidad de IA generativa, y considerando su coste computacional elevado, los parámetros se seleccionaron cuidadosamente para garantizar estabilidad. Se utilizaron las siguientes configuraciones (ver Tabla 3.6):

Tabla 3.6: Configuración del modelo Stable Diffusion 3.5

Parámetro	**Valor**
Modelo base	stable-diffusion-3.5-large
`network_linear`	16
`network_linear_alpha`	16
Optimizador	adamw8bit
`learning_rate`	1e-4
`timestep_type`	linear
`noise_scheduler`	flowmatch
`dtype`	float16
`epochs`	2000

Pruebas de generación

Para evaluar y comparar el rendimiento de los modelos de difusión entrenados con LoRAs, se diseñó un conjunto de experimentos estructurados en varias etapas.

La primera etapa experimental consistió en la generación de imágenes exclusivamente a partir de texto (text2img). Este enfoque inicial fue crucial para dos propósitos principales. Primero, permitió evaluar la capacidad de generación intrínseca y la calidad visual de los LoRAs de manera independiente, sin la influencia de entradas visuales adicionales. Segundo, sirvió como un paso preliminar para la evaluación cualitativa del estilo y la similitud visual de las imágenes generadas con respecto a las muestras reales. Los resultados de estas pruebas iniciales facilitaron el descarte de varias configuraciones para ambos modelos de difusión (Flux y SD), orientando las etapas posteriores.

Considerando esta fase como pruebas preliminares, las etapas subsiguientes se enfocaron en la generación de instancias específicas de aeronaves y la fiabilidad visual de las firmas radar. Para ello, se introdujo un nuevo elemento de entrada: imágenes simuladas de modelos de aeronaves presentes en las muestras de entrenamiento. La inclusión de estas imágenes simuladas en el proceso de generación es fundamental, ya que permite controlar una variable crítica: el ángulo de visión. Esta variable introduce una variación significativa en las firmas radar y, por ende, en la apariencia visual de las aeronaves.

Las simulaciones de entrada se obtienen mediante un procesamiento detallado de simulaciones SLC (Single Look Complex) generadas por un simulador SAR. A partir de estas, se generan imágenes de intensidad multilook normalizadas, que incluyen el enmascaramiento de áreas de sombra, y la aplicación de condiciones de fondo y ruido (NESZ). La presencia de estas condiciones de fondo y ruido en las simulaciones es otra variable clave que fue probada a lo largo de los experimentos. La Figura 3.5 muestra ejemplos representativos de las simulaciones de entrada utilizadas.

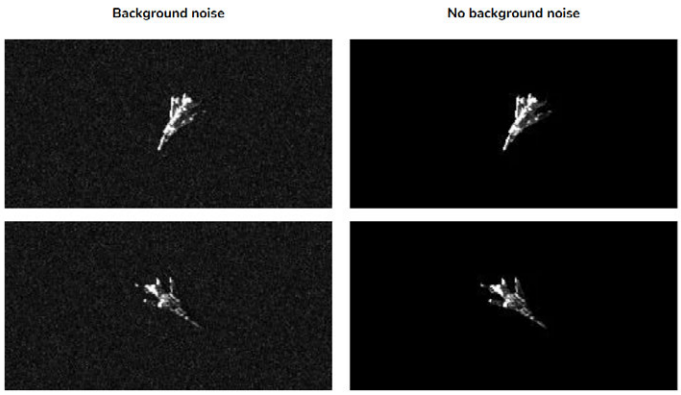


Figura 3.5: Simulaciones de entrada utilizadas para el condicionamiento por imagen. Se muestran dos variantes del mismo modelo, con diferencias en el ángulo de visión y el nivel de ruido de fondo. Hisdesat Servicios Estratégicos

Para las pruebas de text2img e img2img, se empleó una configuración de muestreo consistente para asegurar la comparabilidad de los resultados (ver Tabla 3.7). Los parámetros clave son el sampler y el scheduler, que guían la generación y controlan la eliminación del ruido respectivamente. Se eligió la combinación `karras` y `DPM++` por ofrecer un buen equilibrio entre calidad de imagen y velocidad de procesamiento [49].

Tabla 3.7: Parámetros utilizados en la fase de muestreo

Parámetro	**Valor**
Sampler	ksampler
Seed	Variable (para explorar diferentes resultados)
Steps	30
CFG (Classifier Free Guidance)	7.0
Sampler Name	dpmpp_2m
Scheduler	karras

La principal distinción entre estos dos flujos de trabajo reside en la imagen latente de entrada que recibe el muestreador y el parámetro de eliminación de ruido (`denoise`). En el caso de la generación text2img, el muestreador recibe una imagen latente vacía. Para img2img, la entrada es la imagen latente de la simulación. El parámetro `denoise` se estableció en 1 para la generación pura (text2img), mientras que para img2img se exploró un rango de valores entre 0.5 y 0.9, permitiendo controlar el nivel de variación introducido por el modelo sobre la imagen de entrada.

A continuación, se detallan los mecanismos de control implementados durante el proceso de generación de imágenes mediante ControlNet e IPAdapter.

- **ControlNet**

  ControlNet es una arquitectura de red neuronal que permite un control más preciso sobre la generación de imágenes en modelos de difusión. Funciona mediante la adición de módulos entrenables a un modelo de difusión preexistente, permitiendo que la información de una imagen de entrada (como bordes, poses o mapas de profundidad) guíe el proceso de difusión sin alterar los pesos originales del modelo.

  El flujo de trabajo implementado en ComfyUI para ambos modelos (Flux y SD) comparte una estructura común, lo que permite una explicación conjunta. Este flujo está compuesto por las siguientes partes clave:

  1. **Carga del LoRA**: Se inicializa el modelo con los safetensors de los checkpoints iniciales, el safetensor específico del LoRA entrenado, y los modelos de CLIP y VAE correspondientes.

2. **Carga del ControlNet**: Cada modelo de difusión requiere su modelo de ControlNet correspondiente. Es importante destacar que ControlNet ofrece diversas versiones, cada una especializada en un tipo particular de condicionamiento. Estas versiones, como Depth, Canny, Blur y Openpose, refieren al tipo de preprocesamiento aplicado a la imagen de entrada que se utiliza para guiar la generación. Dada la naturaleza de las imágenes de entrada (simulaciones de aeronaves), se realizaron pruebas exhaustivas con las variaciones Depth, Blur y Canny de ControlNet para ambos modelos (Flux y SD).
3. **Procesamiento de la imagen de entrada**: Antes de ser introducida en el nodo que aplica ControlNet sobre el LoRA, la imagen de entrada se somete al preprocesamiento específico requerido por el tipo de ControlNet seleccionado.

El flujo general para ControlNet utiliza estas componentes para generar una imagen a partir de una imagen latente vacía, empleando tanto el trigger de condicionamiento de texto como el condicionamiento de imagen proporcionado por ControlNet. Durante las pruebas, se variaron dos parámetros clave para equilibrar la libertad creativa del modelo con el peso de la información de la simulación de entrada: el `strength` de ControlNet (0.3-0.6) y el `guidance conditioning` del LoRA (2.8-3.5). La Figura 3.6 ilustra conceptualmente este flujo de trabajo.

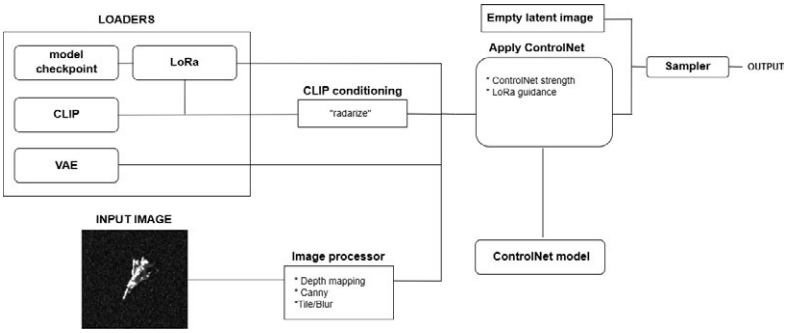


Figura 3.6: Visualización del flujo de trabajo para la aplicación de ControlNet en el proceso de generación.

- **IPAdapter**

  IPAdapter (Image Prompt Adapter) es una técnica que permite integrar la información visual de una imagen de referencia (prompt de imagen) en el proceso de generación de un modelo de difusión, sin necesidad de reentrenar completamente el modelo o el LoRA. A diferencia de ControlNet, IPAdapter no requiere un preprocesamiento específico de la imagen de entrada y es más flexible en cómo el contenido de la imagen de referencia influye en la generación, enfocándose en la transferencia de estilo o contenido semántico.

  Aunque existen nodos distintos en ComfyUI para aplicar IPAdapter en

modelos de Flux y SD3.5, conceptualmente siguen los mismos principios, lo que permite una explicación conjunta de su flujo de trabajo. Este flujo se compone de las siguientes partes:

1. **Carga del LoRA**: Similar a los flujos anteriores, se inicializa el modelo con los safetensors de los checkpoints iniciales, el safetensor específico del LoRA entrenado, y los modelos de CLIP y VAE correspondientes.
2. **Carga de la imagen de entrada**: Una característica distintiva de IPAdapter es que la imagen de entrada no requiere ningún preprocesamiento adicional antes de ser utilizada.
3. **Carga del modelo de IPAdapter**: Existen varias versiones del modelo IPAdapter. En este caso, ambos modelos de difusión utilizaron el IPAdapter correspondiente de InstantX.

El flujo de trabajo para IPAdapter, por lo tanto, utiliza estas componentes para generar una imagen a partir de una imagen latente vacía, empleando tanto el trigger de condicionamiento de texto como el condicionamiento de imagen proporcionado por IPAdapter. Para las pruebas, se ajustaron las siguientes variables con el objetivo de encontrar un equilibrio entre la libertad generativa del modelo y el peso de la imagen de simulación de entrada: el `strength` del IPAdapter (0.5-1.5) y el `guidance conditioning` del LoRA (2.8-3.5). La Figura 3.7 ilustra conceptualmente este flujo de trabajo.

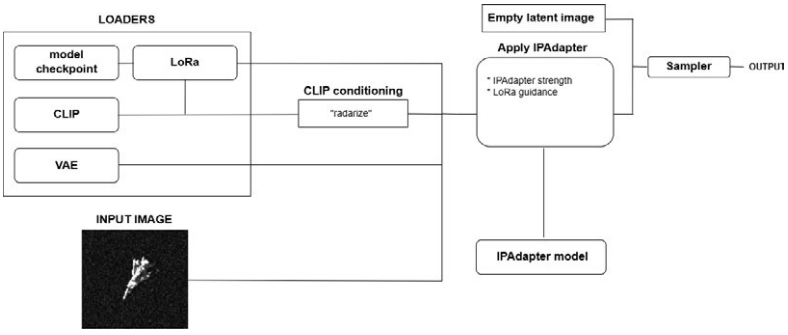


Figura 3.7: Visualización del flujo de trabajo para la aplicación de IPAdapter en el proceso de generación.

Capítulo 4

Resultados

4.1. Detección de aeronaves

Para evaluar el rendimiento de los modelos, se presentan los resultados de los experimentos de esta primera fase en términos de Falsos Positivos (FP), Falsos Negativos (FN) y Accuracy. Las métricas se analizaron sobre dos conjuntos de datos distintos:

- **Conjunto de datos original completo**: Incluye todos los recortes, incluso aquellos sin aeronaves. Este primer conjunto es particularmente útil para analizar el comportamiento del modelo respecto a los Falsos Positivos, dado que el 84 % de los recortes del conjunto completo no contienen aeronaves, y solo una parte seleccionada de estos se incluye finalmente en el conjunto de entrenamiento (en función del valor asignado al parámetro `no_label`).
- **Subconjunto de prueba específico**: Creado con proporciones predefinidas (`no_label` = 1, `n_urban` = 0.5).

Los cálculos de estas métricas se realizaron aplicando un umbral de Intersection over Union (IoU) de 0.7 para la supresión de no máximos (Non-Maximum Suppression, NMS), que es el valor predeterminado en YOLO. Este parámetro reduce el número de detecciones al eliminar las cajas delimitadoras superpuestas. Adicionalmente, se aplicó un umbral de confianza de 0.5, descartando cualquier detección con una puntuación de confianza inferior a este valor.

4.1.1. Rendimiento según variaciones de imgsz y tasa de aprendizaje

Las figuras de esta sección presentan un análisis del rendimiento del modelo en función de distintos tamaños de imagen (`imgsz`), configuraciones de tasa de aprendizaje (learning rate) y variantes del modelo YOLOv8.

El eje X de los gráficos representa el tamaño de imagen (`imgsz`), evaluado en tres valores discretos: 260, 320 y 420 píxeles. El eje Y corresponde a tres gráficos diferentes, cada uno representando una métrica de rendimiento distinta:

- **Gráfico A**: muestra la tasa de Falsos Positivos (FP).
- **Gráfico B**: representa la tasa de Falsos Negativos (FN).
- **Gráfico C**: muestra el accuracy general.

Las líneas de colores en los gráficos indican cuatro configuraciones experimentales distintas, que combinan dos variantes del modelo YOLOv8 (nano y xlarge) con dos configuraciones de tasa de aprendizaje:

- **Configuración 1**: `lr0` = 0.01, `lrf` = 0.001
- **Configuración 2**: `lr0` = 0.001, `lrf` = 0.0001

Estos parámetros se seleccionaron debido a su influencia directa en la velocidad, el proceso de entrenamiento y el rendimiento general del modelo.

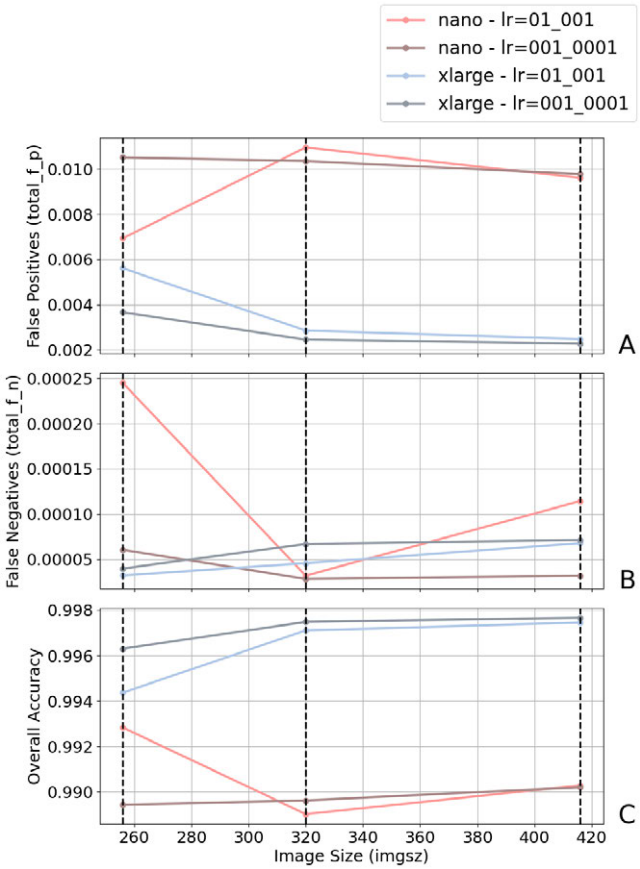


Figura 4.1: Análisis de rendimiento de las variantes del modelo YOLOv8 en el conjunto de datos completo. El eje X representa los tamaños de las imágenes y el eje Y muestra tres métricas de rendimiento: tasa de Falsos Positivos, tasa de Falsos Negativos y Accuracy. Las líneas representan las variantes del modelo Nano y XLarge con sus respectivas tasas de aprendizaje.

La Figura 4.1 ilustra el rendimiento de los modelos sobre el conjunto completo de datos según los hiperparámetros descritos. A continuación, se analiza el comportamiento de los modelos según las tres métricas presentadas.

- **Falsos Positivos** (Figura 4.1A): YOLOv8x supera consistentemente a YOLOv8n en la reducción de FP, especialmente con la configuración de tasa de aprendizaje `lr0` = 0.001 y `lrf` = 0.0001. El tamaño de imagen óptimo para reducir los FP parece ser `imgsz` = 416, aunque `imgsz` = 320 también ofrece resultados competitivos.
- **Falsos Negativos** (Figura 4.1B): YOLOv8n muestra un mejor rendimiento, sobre todo con `lr0` = 0.001 y `lrf` = 0.0001, aunque YOLOv8x exhibe un comportamiento más estable en diferentes configuraciones. Ambos modelos reducen la tasa de FN cuando se utilizan imágenes de menor tamaño, particularmente `imgsz` = 256 o 320.
- **Accuracy** (Figura 4.1C): YOLOv8x obtiene sus mejores resultados con `imgsz` = 416 y `imgsz` = 320 cuando se emplea `lr0` = 0.001 y `lrf` = 0.0001, demostrando su capacidad para mantener un alto valor de accuracy en diversas configuraciones.

La Figura 4.2 presenta el rendimiento de YOLOv8n y YOLOv8x sobre el conjunto de prueba, donde se observa una tendencia similar.

- **Falsos Positivos** (Figura 4.2A): Ambos modelos logran mejores resultados con `imgsz` = 320 y las tasas de aprendizaje `lr0` = 0.001 y `lrf` = 0.0001.
- **Falsos Negativos** (Figura 4.2B): YOLOv8x muestra una notable reducción bajo estas mismas condiciones, así como con `lr0` = 0.01 y `lrf` = 0.001, lo que sugiere una mayor eficacia de YOLOv8x en la detección cuando se emplean imágenes pequeñas.
- **Accurracy** (Figura 4.2C): YOLOv8n alcanza su mejor rendimiento con `imgsz` = 320 y `lr0` = 0.001 / `lrf` = 0.0001. Esto indica que, aunque YOLOv8x destaca en la reducción de FP, YOLOv8n es más efectivo en minimizar FN y mejorar el accuracy bajo ciertas condiciones. En general, ambos modelos tienden a obtener mejores resultados con tamaños de imagen más pequeños, mostrando rendimientos comparables en términos de accuracy y reducción de FN.

En este contexto, la configuración óptima se identificó como YOLOv8x con `imgsz` = 320 y tasas de aprendizaje `lr0` = 0.001 / `lrf` = 0.0001. Esta combinación logró el mejor equilibrio entre accuracy, eficiencia computacional y reducción de Falsos Positivos.

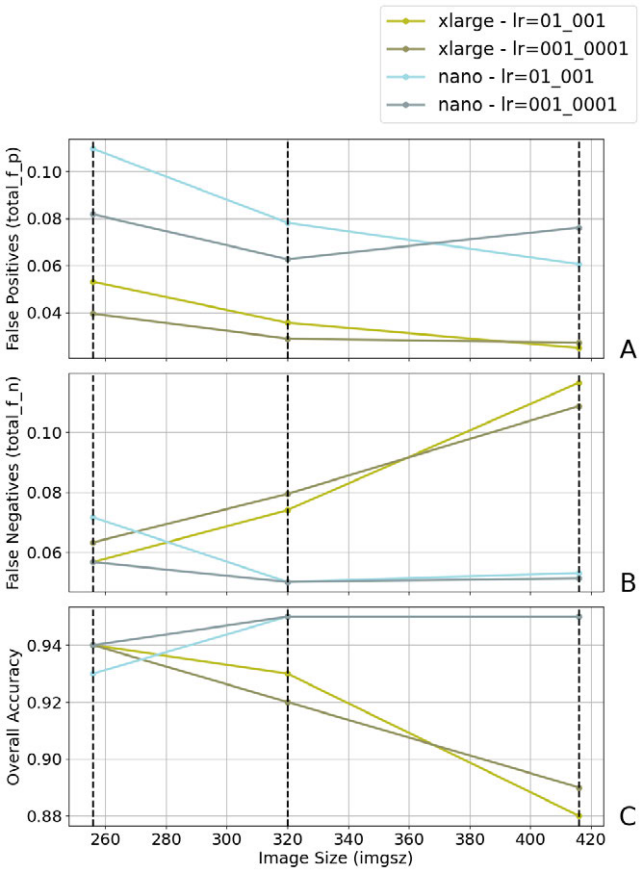


Figura 4.2: Análisis de rendimiento de las variantes del modelo YOLOv8 en el conjunto de prueba. El eje X representa los tamaños de las imágenes y el eje Y muestra tres métricas de rendimiento: tasa de Falsos Positivos, tasa de Falsos Negativos y Accuracy. Las líneas representan las variantes del modelo Nano y XLarge con sus respectivas tasas de aprendizaje.

4.1.2. Impacto de la composición del dataset

Las siguientes figuras analizan el impacto de la composición del conjunto de datos en el rendimiento del modelo.

El eje X de los gráficos representa el porcentaje de recortes sin etiqueta (`no_label`), con valores entre 0 y 1.5 (escalados de 0 a 15). Este parámetro indica la proporción de recortes que no contienen aeronaves etiquetadas, de modo que un valor creciente implica la incorporación de más muestras de fondo al conjunto de entrenamiento. El eje Y se compone de tres gráficos que muestran distintas las métricas de rendimiento:

- **Gráfico A**: muestra la tasa de Falsos Positivos (FP).
- **Gráfico B**: representa la tasa de Falsos Negativos (FN).
- **Gráfico C**: muestra el accuracy general.

Las líneas de colores en los gráficos reflejan distintas configuraciones del modelo, organizadas en dos paletas cromáticas: una para YOLOv8x y otra para YOLOv8n. Ambas arquitecturas se evaluaron con distintos valores del parámetro `n_urban` (0, 2, 4, 5).

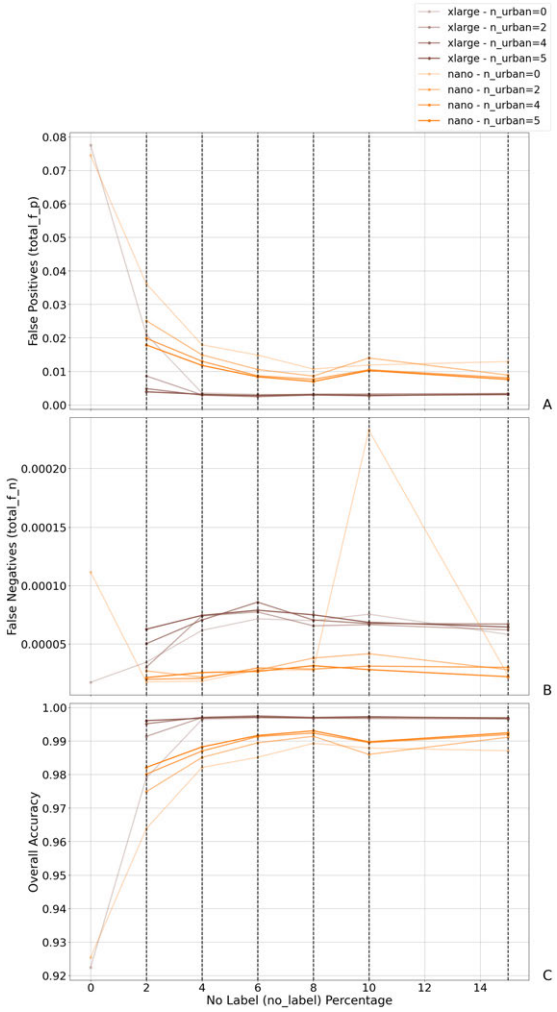


Figura 4.3: Rendimiento en el conjunto de datos completo con diferentes composiciones del conjunto de entrenamiento. El eje X representa el porcentaje de recortes sin etiqueta (sin aeronaves), variando de 0 a 15. El eje Y muestra tres métricas de rendimiento: tasa de Falsos Positivos, tasa de Falsos Negativos y Accuracy. Las líneas representan los modelos YOLOv8 Xlarge y Nano con diferentes proporciones de fondo urbano.

La Figura 4.3 muestra el rendimiento de los modelos sobre el conjunto completo de datos. A continuación, se analiza el comportamiento de los modelos según las tres métricas presentadas.

- **Falsos Positivos** (Figura 4.3A): YOLOv8x supera consistentemente a YOLOv8n, especialmente cuando `n_urban` es 4-5 y `no_label` es 0.8. Esto sugiere que YOLOv8x es más robusto frente a la inclusión de fondos complejos y un mayor porcentaje de parches sin objetos.
- **Falsos Negativos** (Figura 4.3B): En contraste, YOLOv8n logra mejores resultados en la reducción de Falsos Negativos, mostrando una menor sensibilidad al parámetro `n_urban`. Alcanza su mejor rendimiento cuando `no_label` se sitúa entre 0.2 y 0.8. Esto indica que YOLOv8n es más efectivo para detectar aeronaves, incluso en presencia de más ruido de fondo, siempre y cuando la proporción de `no_label` se mantenga dentro de ese rango.
- **Accuracy** (Figura 4.3C): YOLOv8x alcanza su pico de rendimiento con valores elevados de `n_urban` (4-5) y `no_label` de 0.8, demostrando su capacidad para mantener un alto accuracy en distintas configuraciones, especialmente aquellas con mayor complejidad de fondo urbano.

La Figura 4.4 presenta los resultados obtenidos sobre un subconjunto de prueba, confirmando parte de las tendencias observadas.

- **Falsos Positivos** (Figura 4.4A): YOLOv8x conserva su ventaja, con el mejor rendimiento para `n_urban` entre 4-5 y `no_label` entre 0.4 y 0.8. Esto refuerza su superioridad en la minimización de detecciones incorrectas.
- **Falsos Negativos** (Figura 4.4B): Ambos modelos muestran tasas similares de Falsos Negativos cuando `no_label` se encuentra entre 0.2 y 1, aunque YOLOv8x presenta una ligera mejora con valores bajos de `no_label`. Esto sugiere una competencia más cerrada en la capacidad de detección.
- **Accuracy** (Figura 4.4C): YOLOv8n obtiene resultados ligeramente superiores con `n_urban` = 2 y `no_label` entre 0.2 y 0.8, lo que sugiere que puede ofrecer ventajas bajo ciertas condiciones específicas donde el fondo urbano es menos complejo.

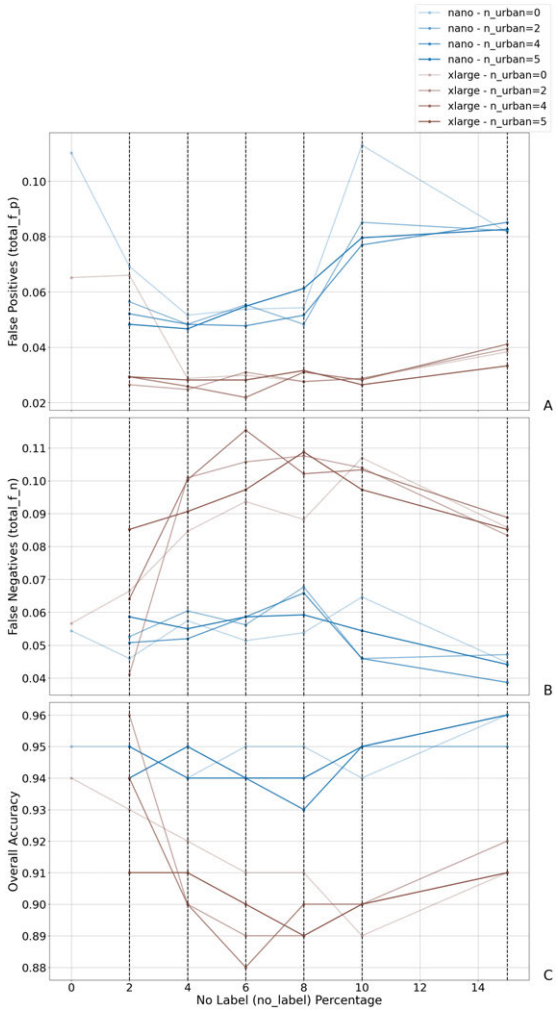


Figura 4.4: Rendimiento en el conjunto de prueba con diferentes composiciones del conjunto de entrenamiento. El eje X representa el porcentaje de recortes sin etiqueta (sin aeronaves), variando de 0 a 15. El eje Y muestra tres métricas de rendimiento: tasa de Falsos Positivos, tasa de Falsos Negativos y Accuracy. Las líneas representan los modelos YOLOv8 Xlarge y Nano con diferentes proporciones de fondo urbano.

La configuración óptima se identifica como `n_urban` = 4/5 y `no_label` = 0.8. Esta combinación logra un equilibrio adecuado entre la reducción de Falsos Positivos y accuracy. Los resultados demuestran un rendimiento robusto tanto en el conjunto completo como en el subconjunto de prueba. YOLOv8x destaca consistentemente en la reducción de Falsos Positivos, mientras que YOLOv8n ofrece mejor accuracy en situaciones concretas con menor complejidad urbana.

4.1.3. Tiempo de inferencia

Con el objetivo de evaluar las demandas computacionales de las distintas variantes del modelo, se realizaron pruebas de temporización para comparar el rendimiento de YOLOv8n y YOLOv8x.

Los tiempos indicados en la Tabla 4.1 se calcularon en segundos durante la predicción de una imagen de dimensiones 6718×2846 píxeles, procesada mediante divisiones en recortes de 100×100 píxeles con un solapamiento del 50 %. Las predicciones se ejecutaron sobre la totalidad de los recortes generados, agrupándolos en lotes cuyo tamaño está definido por el parámetro `batch_size`.

Para cualquier imagen de dimensiones $H \times W$, el número de parches en cada dimensión se calcula mediante:

$$N_D = \left\lfloor \frac{D - 100}{50} \right\rfloor + 1, \quad D \in \{H, W\} \tag{4.1}$$

donde 50 representa el stride, correspondiente al 50 % de solapamiento sobre recortes de 100 píxeles.

El número total de recortes se obtiene multiplicando la cantidad de divisiones en ambas dimensiones:

$$N_{total} = N_H \times N_W \tag{4.2}$$

Aplicando esta fórmula a la imagen concreta de 6718×2846 píxeles utilizada en los experimentos de temporización, se obtiene:

$$N_{total} = \left(\left\lfloor \frac{6718-100}{50} \right\rfloor + 1 \right) \times \left(\left\lfloor \frac{2846-100}{50} \right\rfloor + 1 \right) = 7448 \tag{4.3}$$

Cabe destacar que estas cifras son específicas de la imagen utilizada en las pruebas de rendimiento. El conjunto de datos original incluye imágenes de distintas dimensiones, por lo que el número de recortes procesados varía en función del tamaño de cada imagen.

La Tabla 4.1 resume las métricas temporales obtenidas para los distintos modelos y `batch_size`. Cada una de estas métricas se caracteriza del siguiente modo:

- **Total time**: tiempo transcurrido total, incluyendo tanto el procesamiento de la imagen como la predicción sobre todos los recortes.
- **Prediction Total**: tiempo total dedicado exclusivamente a realizar predicciones con YOLO, considerando todos los recortes procesados.
- **Prediction Average**: tiempo medio por predicción, calculado como el tiempo total de predicción dividido entre el número de recortes del batch correspondiente.

Los resultados experimentales muestran que YOLOv8x requiere entre 15 y 20 segundos para procesar la imagen, frente a los 4–8 segundos necesarios para YOLOv8n. Este incremento en el tiempo de procesamiento está justificado por diversos factores clave, entre ellos la mayor capacidad del modelo para reducir Falsos Positivos.

Tabla 4.1: Comparación del tiempo de inferencia según modelo

Version	Batch Size	Total time	Prediction Total	Prediction Average
YOLOn	10	8.6391	7.8405	0.0010825
YOLOn	50	4.8166	4.0660	0.0005818
YOLOn	100	4.5058	3.7608	0.0005169
YOLOn	200	4.8255	4.0788	0.0005657
YOLOn	300	4.5797	3.7334	0.0005087
YOLOx	**10**	**20.8949**	**20.0914**	**0.0027586**
YOLOx	50	17.5441	16.7810	0.0023167
YOLOx	100	16.0211	15.9381	0.0022984
YOLOx	**200**	**15.6632**	**15.1487**	**0.0022516**
YOLOx	300	15.7004	15.1995	0.0022624

Las pruebas con diferentes tamaños de batch (entre 10 y 300) revelan que ambos modelos se benefician de tamaños más grandes, observándose el mejor rendimiento en el rango de 100 a 200 elementos por batch. En particular, YOLOv8x mejora su eficiencia con tamaños de mayores, reduciendo el tiempo de procesamiento total de 20.89 segundos (tamaño de 10) a 15.66 segundos (tamaño de 200). Esta optimización contribuye a mitigar parcialmente el coste computacional adicional.

En aplicaciones como la detección de aeronaves, donde la precisión y la fiabilidad priman sobre los requisitos de procesamiento en tiempo real, el mayor coste computacional de YOLOv8x representa una compensación razonable. Su mejor desempeño en escenarios complejos y su mayor fiabilidad justifican el aumento en el tiempo de inferencia por imagen, sobre todo considerando que en este caso se prioriza la calidad de detección por encima de la velocidad.

4.1.4. Conclusiones de la exploración

Para la tarea de detección de objetos abordada en este estudio, y según el análisis realizado, la variante YOLOv8x ha demostrado el mejor rendimiento general. Este modelo consigue minimizar eficazmente los Falsos Positivos (FP), un aspecto clave para reducir las detecciones incorrectas. Además, el uso de un `batch_size` entre 200 y 300 permitió alcanzar tiempos de inferencia razonables para los objetivos planteados.

La configuración óptima del modelo emplea una tasa de aprendizaje inicial de `lr0` = 0.001 y final de `lrf` = 0.0001, combinada con un tamaño de imagen (`imgsz`) de 320 píxeles, que ofrece un buen equilibrio entre eficiencia computacional y reducción de Falsos Negativos (FN). En cuanto a la composición del conjunto de datos, se observó un mejor rendimiento al establecer la proporción de recortes sin anotaciones (`no_label`) en 0.8, lo cual permite mantener una relación adecuada entre muestras de fondo y muestras con aeronaves. Asimismo, utilizar una mayor proporción de recortes procedentes de áreas urbanas (`n_urban` entre 4 y 5) favorece el desempeño del modelo en entornos visuales complejos y diversos.

La siguiente tabla (ver Tabla 4.2) resume el rendimiento del modelo YOLOv8x optimizado, tanto sobre el conjunto completo de datos como sobre el conjunto de prueba.

Tabla 4.2: Métricas de rendimiento en los conjuntos de datos

Data	**Accuracy**	**Precision**	**Recall**	**F1**
Whole Dataset	0.99605	0.52115	0.98985	0.68280
Test Set	0.96019	0.95863	0.94939	0.95398

En el conjunto completo, el modelo alcanzó un accuracy global del 99.61 %. Sin embargo, la métrica de precisión fue notablemente inferior, con un valor de 0.52115, lo que indica una tendencia significativa del modelo a clasificar erróneamente elementos de fondo como aeronaves, generando un número elevado de Falsos Positivos. No obstante, se cuenta con un recall del 98.99 %. El F1-score obtenido fue del 68.28 %, señalando que, si bien el modelo detecta eficazmente las aeronaves reales, sigue existiendo margen de mejora en la reducción de detecciones erróneas.

En el caso del conjunto de prueba, el modelo mostró una mejora sustancial en accuracy (0.9587) y un recall de 0.9494, lo que se tradujo en un F1-score de 0.95398. Esto sugiere que, aunque el modelo mantiene un alto rendimiento en la detección de aeronaves, el equilibrio entre precisión y recall es más favorable en este conjunto, posiblemente debido a una mayor homogeneidad en los datos o a una menor complejidad en el fondo.

Las Figuras 4.5 y 4.6 ilustran ejemplos del rendimiento del modelo, mostrando tanto detecciones correctas de aeronaves como errores en los que se identifican incorrectamente detalles del fondo como aviones.

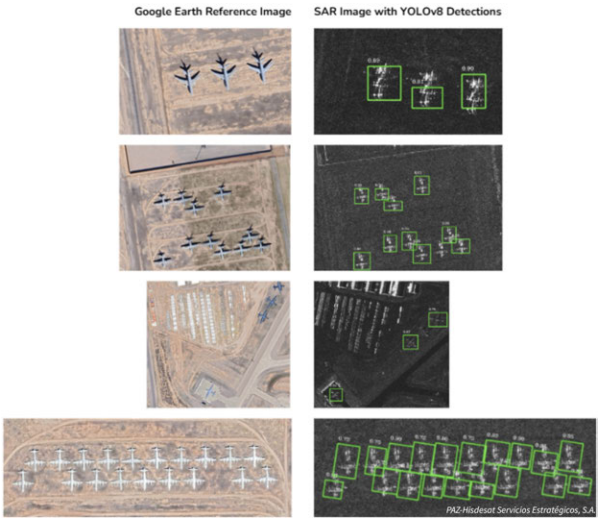


Figura 4.5: Comparación de resultados de detección. La columna izquierda muestra la referencia del área de la aeronave en una imagen de Google Earth, mientras que columna derecha presenta las predicciones del modelo YOLOv8 con cuadros delimitadores verdes sobre la imagen SAR original. PAZ Satellite@Hisdesat Servicios Estratégicos

Figura 4.6: Falsos Positivos de detección. La columna izquierda muestra la referencia del área de la aeronave en una imagen de Google Earth. La columna derecha presenta las predicciones del modelo YOLOv8 sobre la imagen SAR original, con los bounding boxes verdes indicando detecciones correctas y los rojos, Falsos Positivos. PAZ Satellite@Hisdesat Servicios Estratégicos

En la Figura 4.6 se destacan escenarios visuales especialmente complejos, en los que ciertas formas geométricas, características del terreno o elementos de infraestructura son erróneamente clasificados como aeronaves. Estos errores suelen deberse a similitudes visuales, como estructuras lineales o superficies reflectantes que pueden asemejarse al contorno de un avión. Por ejemplo, maquinaria industrial de gran tamaño o estructuras arquitectónicas con formas similares a alas o fuselajes pueden inducir a errores de detección.

En resumen, aunque el modelo presenta un alto recall, su precision sigue siendo un aspecto susceptible de mejora, especialmente en contextos con alta presencia de elementos de fondo.

4.2. Clasificación

Para evaluar la capacidad del modelo de adaptarse a tareas de clasificación, se desarrolló una nueva fase basada en la exploración realizada previamente.

El cambio clave fue modificar el sistema de etiquetado de los recortes que contienen aeronaves. Inicialmente, el conjunto de datos incluía imágenes con coordenadas y un texto indicando el modelo o clase específica de la aeronave. Durante la fase de detección, se simplificó este etiquetado a una única clase constante (clase 0), solo para indicar la presencia de una aeronave. Sin embargo, para la primera tarea de clasificación, se codificaron los distintos tipos de aeronaves como etiquetas numéricas del 0 al 4, sumando cinco clases. Posteriormente, se analizó el rendimiento de clasificación añadiendo más complejidad, dividiendo algunas clases de aeronaves según sus modelos específicos, resultando en diez clases codificadas del 0 al 9.

Utilizando los parámetros previamente identificados como óptimos, el modelo YOLOv8x fue entrenado para la tarea de clasificación con los siguientes valores: `epochs` = 100, `imgsz` = 320, `lr0` = 0.001 y `lrf` = 0.0001. Las métricas de rendimiento para estas tareas, evaluadas sobre los mismos conjuntos de datos descritos anteriormente, se detallan a continuación.

4.2.1. Clasificación de 5 Clases de Aeronaves + Fondo

Las Tablas 4.3 y 4.4 muestran la efectividad del primer modelo de clasificación y cómo influye la distribución de clases en su desempeño.

En el conjunto de prueba, se observa una variabilidad importante en las métricas de precision y recall entre las clases. Las clases mayoritarias (como 0, 3 y 4) suelen alcanzar valores altos de F1-score. Sin embargo, las clases minoritarias (clases 1 y 2) presentan un recall ligeramente inferior. Esto indica que el modelo tiene dificultades para detectar todas las instancias de clases poco representadas, lo cual se puede atribuir al desbalance de clases en los datos de entrenamiento.

Tabla 4.3: Métricas de rendimiento de la clasificación sobre el conjunto de prueba

Class	TP	FP	FN	Precision	Recall	F1
0	146	1	8	0.9932	0.9481	0.9701
1	28	3	5	0.9032	0.8485	0.8750
2	56	6	8	0.9032	0.8750	0.8889
3	204	3	27	0.9855	0.8831	0.9315
4	151	11	14	0.9321	0.9152	0.9235
5	386	62	25	0.8616	0.9392	0.8987

Tabla 4.4: Métricas de rendimiento de la clasificación sobre el conjunto completo

Class	TP	FP	FN	Precision	Recall	F1
0	2034	28	19	0.9864	0.9907	0.9886
1	367	29	20	0.9268	0.9483	0.9374
2	835	15	27	0.9824	0.9687	0.9755
3	2663	23	49	0.9914	0.9819	0.9867
4	2204	47	63	0.9791	0.9722	0.9757
5	2003015	178	8357	0.9999	0.9958	0.9979

La clase 5, correspondiente al fondo, presenta un número relativamente alto de Falsos Positivos (62 FP), lo que afecta negativamente a su precisión. No obstante, mantiene un recall elevado, lo que sugiere que el modelo es capaz de distinguir de manera efectiva entre objetos relevantes y regiones de fondo, aunque siguen produciéndose errores.

Al analizar el rendimiento sobre el conjunto completo de datos, se observa una mejora significativa en las métricas globales, con la mayoría de las clases alcanzando F1-scores superiores al 0.97. La Figura 4.7 muestra cuatro ejemplos de clasificación exitosa correspondientes a diferentes clases de aeronaves, ilustrando la capacidad del modelo para distinguir entre ellas. La clase de fondo (clase 5) alcanza un precision y recall casi perfectos, lo que refuerza la robustez del modelo para diferenciar entre objetos de interés y regiones irrelevantes.

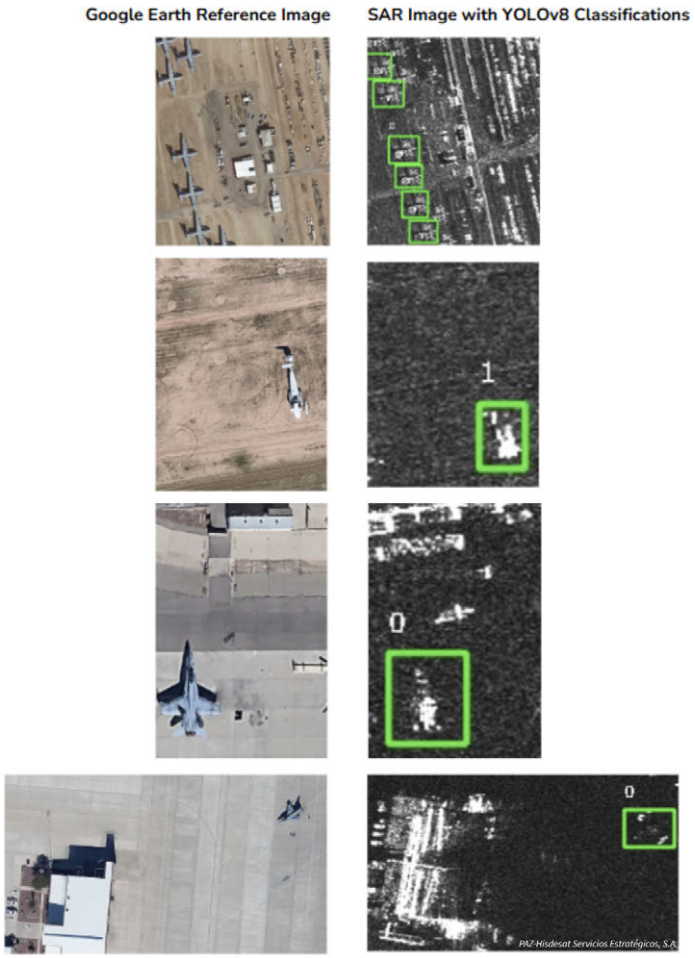


Figura 4.7: Comparación de resultados de clasificación. La columna izquierda muestra la referencia del área de la aeronave en una imagen de Google Earth. La columna derecha presenta las predicciones de clasificación del modelo YOLOv8, indicadas con bounding boxes verdes, sobre la imagen SAR original. PAZ Satellite@Hisdesat Servicios Estratégicos

No obstante, la presencia de Falsos Negativos en ciertas clases, especialmente en la clase 3 (49 FN), sugiere que la variabilidad intra-clase o la superposición de características visuales entre clases podrían estar contribuyendo a errores de clasificación.

4.2.2. Clasificación de 10 Clases de Aeronaves + Fondo

Las Tablas 4.5 y 4.6 muestran el rendimiento del segundo modelo de clasificación, ahora con un mayor número de clases de aeronaves.

Tabla 4.5: Rendimiento de la clasificación extendida sobre el conjunto de prueba

Class	TP	FP	FN	Precision	Recall	F1 Score
0	68	3	6	0.9577	0.9189	0.9379
1	81	4	1	0.9529	0.9878	0.9701
2	23	2	5	0.9200	0.8214	0.8679
3	11	1	0	0.9167	1.0000	0.9565
4	39	6	4	0.8667	0.9070	0.8864
5	114	7	10	0.9421	0.9194	0.9306
6	78	6	1	0.9286	0.9873	0.9571
7	6	0	3	1.0000	0.6667	0.8000
8	14	1	0	0.9333	1.0000	0.9655
9	144	8	22	0.9474	0.8675	0.9057
10	391	52	38	0.8826	0.9114	0.8968

Los resultados obtenidos en el conjunto de prueba son aceptables (Tabla 4.5). La precisión no disminuye en comparación con la prueba de clasificación anterior (Tabla 4.3), lo que indica una mejora o al menos estabilidad en el desempeño del modelo a pesar de la mayor complejidad. Sin embargo, al calcular las métricas sobre el conjunto completo de datos (Tabla 4.6), se observa una cantidad elevada de Falsos Positivos. Esto sugiere que el modelo tiende a generar muchas predicciones para áreas que no contienen aviones.

Tabla 4.6: Rendimiento de la clasificación extendida sobre el conjunto completo

Class	TP	FP	FN	Precision	Recall	F1 Score
0	961	475	18	0.6692	0.9816	0.7959
1	1071	1003	3	0.5164	0.9972	0.6804
2	375	691	19	0.3518	0.9518	0.5137
3	154	137	0	0.5292	1.0000	0.6921
4	673	1086	15	0.3826	0.9782	0.5501
5	1584	1617	28	0.4948	0.9826	0.6582
6	811	1501	9	0.3508	0.9890	0.5179
7	78	216	7	0.2653	0.9176	0.4116
8	191	116	4	0.6221	0.9795	0.7610
9	2217	1972	63	0.5292	0.9724	0.6854
10	2002685	166	8814	0.9999	0.9956	0.9978

Tras analizar la procedencia de estos Falsos Positivos (Tabla 4.7), se observa que más del 97 % provienen de detecciones realizadas por el modelo sobre recortes donde, según las etiquetas, no debería haber aviones. La Tabla 4.7 muestra los porcentajes detallados y en la Figura 4.8 se pueden observar algunos ejemplos de estas detecciones. Curiosamente, en uno de los ejemplos, se puede identificar que, en efecto, había un avión en un recorte que originalmente no estaba etiquetado, lo que podría indicar un error en la anotación original del dataset o la presencia de aeronaves difíciles de percibir visualmente.

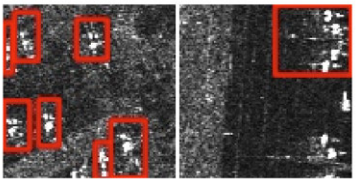


Figura 4.8: Muestras de Falsos Positivos en recortes sin aeronaves. Los bounding boxes en rojo. PAZ Satellite@Hisdesat Servicios Estratégicos

Tabla 4.7: Porcentaje de Falsos Positivos por clase sobre el fondo

Class	Percentage
0	0.9726
1	0.9811
2	0.9667
3	0.9635
4	0.9788
5	0.9944
6	0.9840
7	1.0000
8	0.9138
9	0.9731

El aumento de Falsos Positivos y un posible empeoramiento general al incrementar la granularidad de las clases puede deberse a varios factores:

1. **Mayor ambigüedad entre clases**: Al dividir las aeronaves en modelos más específicos, las diferencias visuales entre algunas subclases pueden ser muy sutiles. Esto aumenta la ambigüedad y dificulta que el modelo las distinga con precisión, llevando a más clasificaciones erróneas.

2. **Aumento del desbalance de clases**: Incrementar el número de clases suele exacerbar el problema del desbalance.

3. **Complejidad del fondo**: La persistencia de Falsos Positivos en recortes sin aeronaves sugiere que el modelo aún confunde ciertas características del fondo con aeronaves, incluso al intentar clasificar a un nivel más detallado. La alta proporción de recortes sin aeronaves en el conjunto de datos total (más del 80 %) sigue siendo un desafío significativo.

4.2.3. Discusión

Los resultados obtenidos demuestran el potencial destacado del modelo YOLOv8. En la tarea de detección binaria, se observaron diferencias significativas en las métricas de rendimiento entre el conjunto de datos completo y el conjunto de prueba final, especialmente en precisión. Esta disparidad sugiere distintos niveles de complejidad entre las distribuciones de los datos, posiblemente debido a la alta proporción de casos de fondo en el conjunto completo.

En contraste, el modelo entrenado para clasificación multiclase (5 tipos de aeronaves) exhibió una precisión y F1-score considerablemente superiores. Esto indica que YOLOv8 se desempeña mejor al discriminar entre múltiples categorías que al limitarse a detectar presencia/ausencia. Este comportamiento se explica por la exigencia de la clasificación multiclase, que obliga al modelo a

extraer representaciones más específicas y robustas de cada categoría.

La diferencia de rendimiento radica en la naturaleza de cada tarea. En el enfoque multiclase, el modelo debe detectar y categorizar correctamente cada instancia, elevando el umbral de selectividad y logrando precisiones individuales superiores a 0.92. Por el contrario, la configuración binaria agrupa todas las aeronaves bajo una única etiqueta, creando una categoría heterogénea que dificulta el reconocimiento de patrones consistentes y compromete el rendimiento general.

Para contextualizar los hallazgos, a continuación, se presenta un análisis comparativo entre los resultados obtenidos y los modelos representativos del estado del arte en detección y clasificación de aeronaves en imágenes SAR, detallados en las Tablas 4.8 y 4.9, respectivamente. Si bien una tendencia común en este campo es la evaluación sobre conjuntos públicos como SARAircraft-1.0 [50], la originalidad de esta propuesta radica en el análisis de rendimiento sobre un conjunto de datos no estandarizado.

Tabla 4.8: Comparación del rendimiento de detección de varios modelos

Model	**Precision**	**Recall**	**Dataset**
YOLOv7	0.880	0.878	SAR-AIRcraft-1.0
YOLO-SAD	0.893	0.879	SAR-AIRcraft-1.0
FCCS-YOLO	0.901	0.932	SAR-AIRcraft-1.0
SAR-NTV-YOLOv8	0.9353	0.9217	SAR-AIRcraft-1.0
Fine-tuned YOLOv8	0.9586	0.9493	Hisdesat

Tabla 4.9: Comparación del rendimiento de clasificación de varios modelos

Model	**Average Accuracy**	**Dataset**
AConvNet	0.8157	SARAircraft-1.0
VGG	0.8571	SARAircraft-1.0
ResNet	0.8729	SARAircraft-1.0
ViT	0.8871	SARAircraft-1.0
EfficientNet	0.8957	SARAircraft-1.0
ConvNeXt	0.9029	SARAircraft-1.0
CMSF	0.9143	SARAircraft-1.0
SFSA	0.9243	SAR-AIG
Fine-tuned YOLOv8	0.9434	Hisdesat

En la detección de objetos (Tabla 4.8), el modelo YOLOv8 de este proyecto fue comparado con otras variantes de la familia YOLO evaluadas sobre SARAircraft-1.0, incluyendo YOLOv7, YOLO-SAD [51], FCCS-YOLO [20] y

SAR-NTV-YOLOv8 [19]. El modelo con mejor desempeño previo en el conjunto público, FCCS-YOLO, alcanzó una precisión de 0.901 y un recall de 0.932. En contraste, los resultados de este proyecto superaron consistentemente estos valores, logrando una precisión de 0.9586 y un recall de 0.9493, lo que enfatiza la eficacia de la adaptación de YOLO sobre un conjunto personalizado. A pesar de este rendimiento superior, la reducción de Falsos Positivos emerge como un área clave para mejoras.

En cuanto a la clasificación (Tabla 4.9), dado que la investigación con arquitecturas basadas en YOLO aún es limitada en este ámbito, los resultados fueron comparados con modelos clásicos de clasificación entrenados sobre SARAircraft-1.0. Entre los mejores resultados de la literatura se encuentran ConvNeXt [52] (0.9029), CMSF [53] (0.9143) y SFSA [54] (0.9243). YOLOv8 ajustado superó a todos ellos, alcanzando una precisión media del 0.9434.

En conjunto, estos hallazgos refuerzan la creciente relevancia de YOLOv8 en tareas de análisis de aeronaves basadas en SAR y destacan su notable capacidad de adaptación a distintos conjuntos y tareas. No obstante, a pesar de los prometedores resultados, se identificaron desafíos clave inherentes a las características del conjunto de datos utilizado.

Ante estos desafíos, se incorporó al proyecto una fase exploratoria centrada en la generación de datos sintéticos. Esta estrategia busca no solo aumentar la cantidad de instancias de aeronaves en el conjunto de entrenamiento, sino también, y especialmente, reforzar las clases menos representadas, con el objetivo de mitigar los efectos del desbalanceo y la escasez de datos, y así potencialmente mejorar aún más la robustez y generalización del modelo.

4.3. Generación de imágenes sintéticas

En un inicio, se adoptó un enfoque exploratorio centrado en analizar la calidad de la generación base antes de introducir información condicionante. En todo este apartado se prescindió de métricas cuantitativas de similitud, no como una omisión arbitraria, sino como una decisión metodológica deliberada. El objetivo principal en esta fase fue analizar la fiabilidad y la similitud visual entre las imágenes generadas y las muestras reales. Se buscaba identificar aproximaciones iniciales prometedoras que sentaran las bases para las líneas futuras del proyecto.

Esta fase inicial permitió descartar configuraciones ineficaces y refinar los parámetros óptimos de los modelos de difusión, asegurando que las iteraciones subsiguientes se basaran en una comprensión sólida de sus capacidades inherentes. La justificación de esta aproximación radica en la naturaleza de la investigación y el desarrollo de nuevas metodologías: antes de cuantificar, es imperativo cualificar. Solo así se puede asegurar que las métricas de rendimiento posteriores se apliquen a un conjunto de datos base que ha demostrado, visual y conceptualmente, su relevancia y potencial.

4.3.1. Exploración inicial con FLUX y Stable Diffusion 3.5

Para el modelo FLUX, se realizaron entrenamientos iniciales variando los hiperparámetros tal como se detalla en el Apartado 3.2.2. Se entrenó un total de 10 modelos, explorando diferentes combinaciones de tamaño de entrada y número de epochs.

Debido a limitaciones del entorno de entrenamiento (ComfyUI), los experimentos con un tamaño de entrada mayor (512x512) se restringieron a subconjuntos de entrenamiento de 50 y 500 recortes. En contraste, los entrenamientos con un tamaño de entrada de 100x100 pudieron realizarse con las tres variaciones de tamaño de subconjunto: 50, 500 y 1009 recortes.

En esta misma etapa, se procesó la generación img2img con los LoRAs resultantes, variando el parámetro denoise (0.5, 0.7, 0.9) para observar cómo la firma radar se alteraba a medida que el modelo disponía de mayor libertad para modificar la imagen.

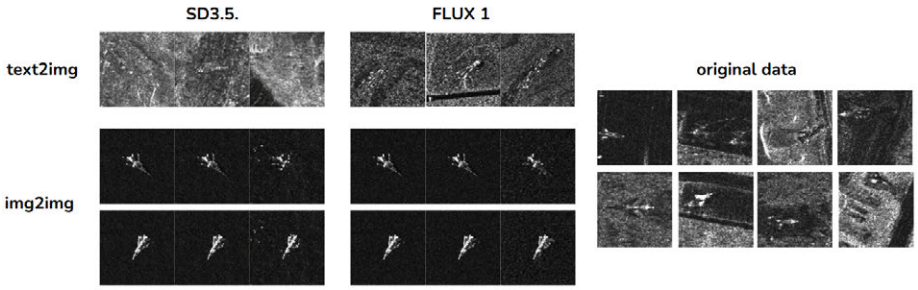


Figura 4.9: Comparación de resultados de generación text2img e img2img. La zona izquierda muestra los mejores resultados obtenidos con SD3.5. En la zona central, se presentan los mejores resultados de FLUX. La zona derecha corresponde a los recortes originales de las imágenes SAR utilizados como referencia.

En la Figura 4.9 (zona central), se presentan los mejores resultados obtenidos con FLUX. La configuración óptima resultó ser: input redimensionado a 512x512, 1600 epochs y un tamaño de entrenamiento de 50 recortes. Estos resultados pueden compararse directamente con los recortes originales mostrados en la zona derecha de la Figura 4.9, lo que permite una apreciación visual de la similitud lograda. Las filas superiores de la figura muestran los resultados de la generación pura (text2img), mientras que las filas inferiores ilustran los resultados del proceso img2img para cada una de las imágenes de simulación. Esto destaca la variabilidad de los resultados en función de la variación del ángulo de la aeronave, un aspecto crucial para la robustez del modelo.

Por otro lado, con Stable Diffusion 3.5, se entrenó un total de 3 modelos. Las limitaciones del entorno de entrenamiento, donde el workflow solo aceptaba imágenes de tamaños superiores o iguales a 512x512, restringieron los experimentos a recortes redimensionados a 512x512. Por lo tanto, los entrenamientos realizados fueron de 2000 epochs con variaciones en el tamaño del subconjunto de entrenamiento e input de 512x512.

En la Figura 4.9 (zona izquierda), se observan los mejores resultados obtenidos con SD3.5. La configuración más efectiva fue: input redimensionado a 512x512, 2000 epochs y un tamaño de entrenamiento de 1009 recortes.

Es importante mencionar que, para una comparación justa y precisa entre FLUX y SD3.5, el conjunto de outputs que se presenta ha sido generado utilizando el mismo conjunto de semillas de generación. De un total de 10 muestras generadas, se han seleccionado y elegido las 3 más representativas para su visualización.

Tras esta exploración inicial, se realizó una evaluación cualitativa de los resultados. Se observó una alta alteración de la firma radar con valores de denoise más altos en los modelos de SD3.5, lo que indicaba una menor fidelidad a la estructura original. En contraste, los modelos de FLUX mostraron una mayor similitud en el grano del ruido y una mejor preservación de la firma radar. Basándonos en estas observaciones, y con el objetivo de mantener la fidelidad a las características visuales de los datos SAR reales, se optó por continuar las consiguientes fases de experimentación exclusivamente con el modelo de FLUX.

4.3.2. Generación condicionada por imagen

A continuación, se desarrollarán los resultados obtenidos con el modelo FLUX, aplicando dos workflows de condicionamiento por imagen: ControlNet e IPAdapter.

Como se mencionó anteriormente, en las pruebas con ControlNet e IPAdapter, se variaron dos parámetros clave para equilibrar la libertad creativa del modelo con el peso de la información de la simulación de entrada: el strength de ControlNet/IPAdapter y el guidance conditioning del LoRA. La optimización de estos parámetros fue crucial para lograr un equilibrio entre la fidelidad a la entrada visual y la capacidad del modelo para generar variaciones realistas.

En la Figura 4.10, se muestran los mejores resultados obtenidos con ControlNet. La configuración más efectiva fue: strength de ControlNet = 0.6 y guidance conditioning del LoRA = 3.5. Estos resultados se consiguieron utilizando el tipo de ControlNet dependiente del preprocesamiento con Blur/Tile y con la imagen de input de la simulación que incluía grano. En la figura, se observan dos filas representativas: A y B, que corresponden a los outputs de dos pruebas finales realizadas con la configuración comentada.

La prueba A equivale a la generación aleatoria de la configuración mencionada, utilizando el input original con grano. Por otro lado, la prueba B corresponde a una variación en el preprocesamiento del input del simulador: se aplicó un shifting y rescaling a la simulación. Esta estrategia se implementó para aportar 2 grados más de libertad a la generación con ControlNet.

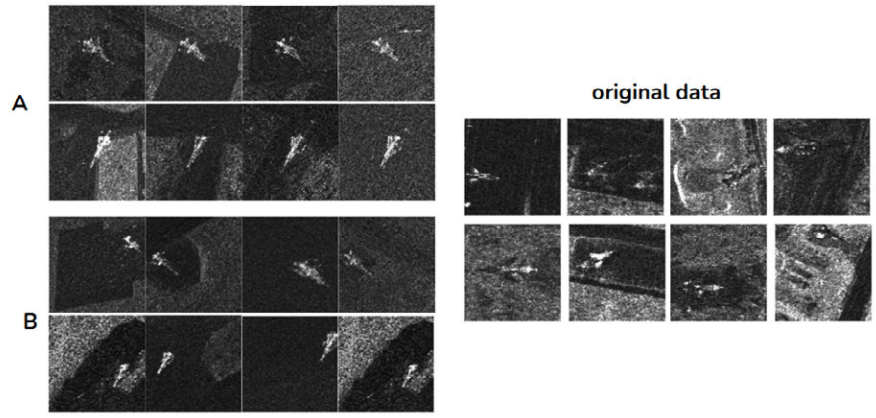


Figura 4.10: Comparación de resultados de generación con ControlNet. La zona izquierda presenta las dos pruebas de entrada realizadas: (A) el input original y (B) el input con aplicación de desplazamiento (shifting) y escalado (scaling). La zona derecha muestra los recortes originales de las imágenes SAR.

Debido a la naturaleza del funcionamiento de ControlNet, las imágenes generadas respetan, además del ángulo de la aeronave (como es el objetivo principal), la posición y el tamaño de dicha aeronave dentro de la imagen. Por lo tanto, la introducción de la variabilidad de la escala del avión y su posición dentro de la imagen ofrece una variabilidad significativa en las imágenes generadas.

En la Figura 4.11, se presentan los mejores resultados obtenidos con IPAdapter. La configuración óptima fue: strength de IPAdapter = 1.2 y guidance conditioning del LoRA = 3.0. En la figura, al igual que con ControlNet, se observan dos filas, A y B, representativas de los outputs de dos pruebas finales con la configuración comentada. Cada una de estas filas fue generada con una de las dos imágenes originales de input del simulador, aunque sin el grano añadido.

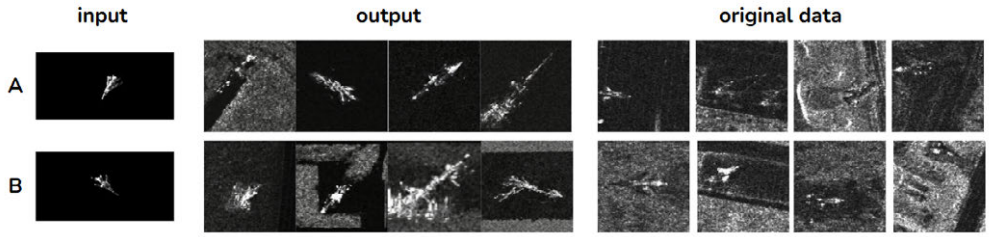


Figura 4.11: Comparación de resultados de generación con IPAdapter. La zona izquierda presenta las dos pruebas realizadas, cada una correspondiente a una imagen de entrada. La zona derecha muestra los recortes originales de las imágenes SAR.

Como se observa, IPAdapter utiliza el objeto del input original y lo incorpora de manera efectiva en la generación final. Esto resulta en una variación de la firma radar en posición, tamaño y ángulo. A diferencia de ControlNet, que mantiene una rigidez mayor en la disposición espacial, IPAdapter ofrece una flexibilidad que permite al modelo interpretar y recontextualizar el objeto de interés, generando un conjunto de datos más diverso en términos de variaciones geométricas y de perspectiva.

4.3.3. Detección con YOLO

Posterior a estas exploraciones y a la identificación de las configuraciones óptimas de generación, se realizaron simulaciones de detección con YOLO para observar su comportamiento en las imágenes generadas. Se llevó a cabo una generación aleatoria de 100 muestras para cada una de las pruebas mencionadas: dos de ControlNet (Prueba A y B, referidas en la Tabla 4.10) y dos de IPAdapter (Pruebas A y B, referidas en la Tabla 4.11). A continuación, se presentan las métricas de detección obtenidas.

Tabla 4.10: Rendimiento de detección y confianza bajo diferentes transformaciones de entrada

Input Type	**Detection Rate (%)**	**Over 0.5 Conf. (%)**
Original Input (both images)	64.00	53.00
Shifting + Rescaling	36.67	30.83

Tabla 4.11: Rendimiento de detección y confianza por tipo de imagen en el proceso de generación

Input Type	**Detection Rate (%)**	**Over 0.5 Conf. (%)**
Image 0 Only	38.00	30.00
Image 1 Only	36.00	22.00

Los resultados de detección en la generación con ControlNet muestran un mayor acierto de detección en la prueba original A, con un total del 53 % de confianza superior a 0.5. Esto indica que las imágenes generadas con esta configuración, que mantenían una mayor fidelidad a la disposición espacial del input original, fueron más fácilmente detectables por el modelo YOLO. Sin embargo, una vez introducida la variabilidad de escala y posición (prueba B), estos resultados se reducen a un 30 %. Por otro lado, los resultados con IPAdapter mantienen un nivel similar a la prueba B de ControlNet, cercanos al 30 %.

Cabe mencionar que el hecho de que YOLO no sea capaz de detectar todas las generaciones con un porcentaje de confianza alto no es directamente representativo de una mala calidad o un fallo inherente en los experimentos de generación. Por el contrario, esta observación puede atribuirse a factores clave que subrayan la complejidad del problema:

- Variabilidad introducida: Los métodos de generación como ControlNet e IPAdapter fueron diseñados precisamente para introducir variabilidad (posición, escala, ángulo). Si bien esto es beneficioso para la generalización futura de un detector, un modelo YOLO que no ha sido entrenado con esta gama de variaciones puede percibir estas imágenes como «out-of-distribution».

- La fase de generación se centró en la similitud visual y la capacidad de los modelos para replicar las firmas radar de manera convincente. Una imagen puede ser visualmente indistinguible de una real y, sin embargo, presentar sutiles artefactos o características que el detector aún no ha aprendido a interpretar correctamente.

Capítulo 5

Conclusiones y trabajos futuros

5.1. Cumplimiento de los objetivos

A lo largo del desarrollo de este proyecto se han abordado de manera sistemática los objetivos planteados inicialmente. En primer lugar, se ha diseñado, implementado y evaluado un sistema automático basado en técnicas de aprendizaje profundo para la detección y clasificación de aeronaves en imágenes de Radar de Apertura Sintética (SAR). El sistema ha sido validado mediante un conjunto de experimentos, obteniendo resultados que evidencian su capacidad para operar en el complejo dominio de las imágenes SAR.

El sistema ha sido construido con una arquitectura modular que permite el despliegue independiente de sus distintos componentes. En particular, se ha desarrollado un módulo específico para las tareas de entrenamiento, evaluación e inferencia con los modelos YOLO, estructurado mediante scripts diferenciados para detección y clasificación multiclase. Esta organización modular ha sido concebida para simplificar su mantenimiento e integración, incluyendo una librería propia y una instalación de dependencias contenida en un único paquete.

Dentro del proceso de validación, se ha llevado a cabo una evaluación comparativa de diversas configuraciones de modelos de la familia YOLOv8, concretamente las variantes nano y xlarge. Se han explorado diferentes combinaciones de parámetros como el tamaño de la imagen de entrada (imgsz), la tasa de aprendizaje (lr0, lrf) y la composición del conjunto de datos de entrenamiento. Esta comparación ha permitido identificar las arquitecturas más adecuadas para el conjunto de datos utilizado y seleccionar las configuraciones óptimas.

Asimismo, se han implementado capacidades de clasificación de aeronaves con dos niveles de granularidad: una configuración básica de 5 clases y otra extendida de 10 clases. Los resultados obtenidos han ofrecido información relevante sobre cómo el desbalance de clases y el aumento de granularidad afectan a la capacidad predictiva del sistema.

En paralelo, se ha iniciado una línea de investigación centrada en el uso de datos sintéticos generados mediante modelos de difusión, como FLUX y Stable Diffusion. Los primeros hallazgos han demostrado la viabilidad de esta técnica como estrategia de aumento de datos, sentando las bases para futuros desarrollos orientados a mejorar la robustez del sistema.

Finalmente, los resultados presentados sobre detección y clasificación están actualmente en proceso de publicación en un artículo científico. Esto refuerza la validez y el impacto de los resultados obtenidos. Cabe destacar que tanto las evaluaciones como las futuras líneas de trabajo se están desarrollando en cola-

boración continua con la empresa Hisdesat.

5.2. Impacto social y medioambiental

El desarrollo de un sistema automático para la detección y clasificación de aeronaves en imágenes SAR implica consideraciones relevantes más allá del plano técnico. Sus posibles aplicaciones abarcan tanto el ámbito civil como el de la seguridad, lo que le confiere un impacto potencial significativo a nivel social y, en ciertos contextos, medioambiental. Aunque el objetivo del proyecto se ha centrado en la mejora del rendimiento técnico de los modelos de detección, es necesario contextualizar su utilidad en escenarios reales.

Desde el punto de vista social, esta tecnología podría contribuir al refuerzo de la seguridad aérea. En entornos con visibilidad reducida, zonas remotas o situaciones donde los sistemas de radar convencionales presentan limitaciones, el uso de imágenes SAR procesadas de forma automatizada permite complementar los sistemas actuales.

En el ámbito de defensa, estos sistemas ofrecen herramientas que podrían integrarse en tareas de vigilancia fronteriza o detección de aeronaves no autorizadas.

Este proyecto puede tener un papel relevante también en situaciones de emergencia. La capacidad de detectar aeronaves de forma autónoma y fiable en condiciones meteorológicas adversas o en terrenos de difícil acceso podría agilizar las labores de búsqueda y rescate.

No obstante, resulta fundamental tener presente el carácter de doble uso que posee cualquier tecnología de vigilancia. Aunque las aplicaciones presentadas en este trabajo se enmarcan en un contexto técnico y académico, su posible implementación a escala real requiere una reflexión ética. Es imprescindible que cualquier uso futuro se rija por principios de transparencia, respeto a los derechos fundamentales y adecuación a los marcos legales establecidos para garantizar un despliegue responsable.

Aunque este proyecto no aborda de forma directa aplicaciones medioambientales, existen ciertos beneficios indirectos que podrían explorarse en trabajos futuros. Por ejemplo, una mejor gestión del tráfico aéreo, basada en una detección más precisa, podría facilitar la optimización de rutas y, con ello, la reducción del consumo de combustible. Asimismo, en situaciones de emergencia vinculadas a fenómenos climáticos extremos, el uso de imágenes SAR para la planificación logística y el despliegue de medios aéreos puede reforzar las capacidades de respuesta, ayudando a reducir daños ambientales adicionales.

5.3. Lineas futuras

La evolución constante del campo del aprendizaje automático y la complejidad propia de los datos SAR abren múltiples líneas de trabajo que podrían fortalecer y ampliar los resultados obtenidos. En este contexto, se identifican direcciones de futuro que no solo permitirían optimizar el sistema desarrollado, sino también abordar algunos de los principales desafíos detectados a lo largo del proyecto.

Una de las áreas con mayor potencial para el desarrollo futuro es la profundización en la generación de imágenes SAR sintéticas. Aunque la primera toma de contacto con modelos de difusión, como FLUX, ha ofrecido resultados visualmente prometedores, resulta imprescindible avanzar hacia una evaluación más sistemática de la calidad y la similitud de las imágenes generadas respecto a los datos reales. Para ello, sería recomendable aplicar métricas cuantitativas que midan la similitud entre ambas fuentes. Entre estas métricas, destacan el Error Cuadrático Medio (MSE), la Relación Señal-Ruido Pico (PSNR) y el Índice de Similitud Estructural (SSIM).

El objetivo último de la generación sintética es contribuir a mejorar la robustez y la capacidad de generalización de los modelos de detección y clasificación. En esta línea, un paso clave será la integración estratégica de las imágenes generadas dentro del conjunto de entrenamiento de YOLO. Esto implica diseñar datasets equilibrados que compensen la escasez de muestras con aeronaves y el desbalance de clases previamente observado.

Por otro lado, será fundamental validar el rendimiento del modelo enriquecido con datos sintéticos en escenarios reales y condiciones operativas variadas. Esta evaluación permitirá determinar si la incorporación de estos datos se traduce efectivamente en una mejora en las métricas.

En resumen, las líneas futuras de este trabajo se orientan a consolidar las aproximaciones iniciales mediante un sistema más robusto y eficaz. La generación de datos sintéticos se perfila como un componente esencial para superar las limitaciones derivadas de la disponibilidad y diversidad de los datos reales, contribuyendo así a ampliar el alcance y aplicabilidad de la solución desarrollada.

Bibliografía

[1] NASA EARTHDATA. (2025) Synthetic aperture radar (sar). Accessed on: Jan. 22, 2025. [Online]. Available: https://www.earthdata.nasa.gov/learn/earth-observation-data-basics/sar

[2] Z. Huang, L. Liu, S. Yang, Z. Wang, G. Cheng, and J. Han, "Physics-guided detector for sar airplanes," https://arxiv.org/abs/2411.12301, 2024, arXiv preprint arXiv:2411.12301.

[3] J. Wang, G. Liu, J. Liu, W. Dong, and W. Song, "Automatic aircraft identification with high precision from sar images considering multiscale problems and channel information enhancement," *Remote Sensing*, vol. 16, no. 17, p. 3177, 2024.

[4] S. El Ghazouali, A. Gucciardi, F. Venturini, N. Venturi, M. Rueegsegger, and U. Michelucci, "Flightscope: A deep comprehensive review of aircraft detection algorithms in satellite imagery," https://arxiv.org/abs/2404.02877, 2024, arXiv preprint arXiv:2404.02877.

[5] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. Papathanassiou, "A tutorial on synthetic aperture radar," *IEEE Geoscience and Remote Sensing Magazine*, vol. 1, no. 1, pp. 6–43, 2013.

[6] "Umbra open data," https://umbra.space/open-data/, accessed: 2025-06-17.

[7] M. A. Richards, *Fundamentals of Radar Signal Processing*, 2nd ed. McGraw-Hill, 2014.

[8] A. Jalil, H. Yousaf, and M. I. Baig, "Analysis of cfar techniques," in *2016 13th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, Islamabad, Pakistan, 2016, pp. 654–659.

[9] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8–36, 2017.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014.

[11] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2961–2969.

[15] M. Carranza-García, J. Torres-Mateo, P. Lara-Benítez, and J. García-Gutiérrez, "On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data," *Remote Sensing*, vol. 13, no. 1, p. 89, 2020.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[17] OpenMMLab Contributors, "Mmyolo: Yolov8 configuration files," https://github.com/open-mmlab/mmyolo/tree/main/configs/yolov8, 2025, accessed: 2025-06-18.

[18] N. Jegham, C. Y. Koh, M. Abdelatti, and A. Hendawi, "Yolo evolution: A comprehensive benchmark and architectural review of yolov12, yolo11, and their previous versions," 2025.

[19] X. Guo and B. Xu, "Sar-ntv-yolov8: A neural network aircraft detection method in sar images based on despeckling preprocessing," *Remote Sensing*, vol. 16, no. 18, p. 3420, 2024.

[20] J. Fang and X. Wang, "Fccs-yolo: Improved yolov8 with contrastive learning for aircraft detection in sar images," https://doi.org/10.20944/preprints202412.1653.v1, 2024, preprints 2024121653.

[21] N. Wiangkam and S. Jiriwibhakorn, "Comparison of yolov8 models for aircraft detection in airport apron using digital image processing," *Engineering and Technology Horizons*, vol. 41, no. 3, 2024.

[22] Q. Guo, H. Wang, and F. Xu, "Scattering enhanced attention pyramid network for aircraft detection in sar images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 9, pp. 7570–7587, 2021.

[23] C. He, M. Tu, D. Xiong, F. Tu, and M. Liao, "A component-based multi-layer parallel network for airplane detection in sar imagery," *Remote Sensing*, vol. 10, no. 7, p. 1016, 2018.

[24] X. Xiao, H. Jia, P. Xiao, and H. Wang, "Aircraft detection in sar images based on peak feature fusion and adaptive deformable network," *Remote Sensing*, vol. 14, no. 23, p. 6077, 2022.

[25] "Mstar dataset," https://www.sdms.afrl.af.mil/index.php?collection=mstar, Air Force Research Laboratory (AFRL), Sensor Data Management System (SDMS), n.d., accessed: 2025-06-18.

[26] "Sar target classification using deep learning," https://www.mathworks.com/help/radar/ug/sar-target-classification-using-deep-learning.html, accessed: 2025-06-17.

[27] O. Kechagias-Stamatis and N. Aouf, "Automatic target recognition on synthetic aperture radar imagery: A survey," *IEEE Aerospace and Electronic Systems Magazine*, vol. 36, no. 3, pp. 56–81, 2021.

[28] Y. Li, X. Li, W. Li, Q. Hou, L. Liu, M.-M. Cheng, and J. Yang, "SARDet-100K: Towards open-source benchmark and toolkit for large-scale sar object detection," *arXiv preprint arXiv:2403.06534*, 2024. [Online]. Available: https://arxiv.org/abs/2403.06534

[29] Y. Liu, W. Li, L. Liu, J. Zhou, B. Peng, Y. Song, and X. Li, "Atrnet-star: A large dataset and benchmark towards remote sensing object recognition in the wild," 1995.

[30] G. F. Araujo, R. Machado, and M. I. Pettersson, "Synthetic sar data generator using pix2pix cgan architecture for automatic target recognition," *IEEE Access*, vol. 11, pp. 143 369–143 386, 2023.

[31] Y. Sun, K. Yan, and W. Li, "Cyclegan-based sar-optical image fusion for target recognition," *Remote Sensing*, vol. 15, no. 23, p. 5569, 2023.

[32] M. Zhang, Z. Cui, X. Wang, and Z. Cao, "Data augmentation method of sar image dataset," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, Valencia, Spain, 2018, pp. 5292–5295.

[33] S. Oghim *et al.*, "Sar image generation method using dh-gan for automatic target recognition," *Sensors*, vol. 24, no. 2, p. 670, 2024.

[34] M. Rangzan, S. Attarchi, R. Gloaguen, and S. K. Alavipanah, "Tsgan: An optical-to-sar dual conditional gan for optical based sar temporal shifting," *arXiv preprint arXiv:2401.00440*, 2023.

[35] X. Bai and F. Xu, "Sar to optical image translation with color supervised diffusion model," in *IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium*, 2024, pp. 963–966.

[36] ——, "Accelerating diffusion for sar-to-optical image translation via adversarial consistency distillation," *arXiv preprint arXiv:2407.06095*, 2024.

[37] J. Do, J. Lee, and M. Kim, "C-diffset: Leveraging latent diffusion for sar-to-eo image translation with confidence-guided reliable object generation," *arXiv preprint arXiv:2411.10788*, 2024.

[38] K. Aydin, J. Hanna, and D. Borth, "Sar-to-rgb translation with latent diffusion for earth observation," *arXiv preprint arXiv:2504.11154*, 2025.

[39] S.-H. Kim and D. Chung, "Conditional brownian bridge diffusion model for vhr sar to optical image translation," *IEEE Geoscience and Remote Sensing Letters*, 2025.

[40] H. Shi *et al.*, "A brain-inspired approach for sar-to-optical image translation based on diffusion models," *Frontiers in Neuroscience*, vol. 18, p. 1352841, 2024.

[41] Z. Yu, M. Y. I. Idris, and P. Wang, "Dc4cr: When cloud removal meets diffusion control in remote sensing," *arXiv preprint arXiv:2504.14785*, 2025.

[42] Y. Hu *et al.*, "Multimodal diffusion bridge with attention-based sar fusion for satellite image cloud removal," *arXiv preprint arXiv:2504.03607*, 2025.

[43] W. Zhang, J. Mei, and Y. Wang, "Dmdiff: A dual-branch multimodal conditional guided diffusion model for cloud removal through sar-optical data fusion," *Remote Sensing*, vol. 17, no. 6, p. 965, 2025.

[44] Hisdesat. (2025) Paz. Accessed on: Feb. 04, 2025. [Online]. Available: https://www.hisdesat.com/paz/

[45] N. Verde, G. Mallinis, M. Tsakiri-Strati, C. Georgiadis, and P. Patias, "Assessment of radiometric resolution impact on remote sensing data classification accuracy," *Remote Sensing*, vol. 10, no. 8, p. 1267, 2018.

[46] Ultralytics, "ultralytics/yolo," 2025, accessed on: Jan. 24, 2025. [Online]. Available: https://github.com/ultralytics/ultralytics/blob/51d8cfa9c37b7b2b98b3d3ec5a6f1a9ff6b38359/ultralytics/yolo/engine/trainer.py#L634

[47] ComfyUI Development Team, "Comfyui," https://www.comfy.org/, accessed: 2025-06-19.

[48] ostris / ai-toolkit Contributors, "ai-toolkit," https://github.com/ostris/ai-toolkit/, accessed: 2025-06-19.

[49] F. Sanz. (2023) Guía completa de samplers en stable diffusion. Accessed on: Mar. 22, 2025. [Online]. Available: https://www.felixsanz.dev/es/articulos/guia-completa-de-samplers-en-stable-diffusion#familia-de-modelos-dpm

[50] W. Zhirui, K. Yuzhuo, Z. Xuan, W. Yuelei, Z. Ting, and S. Xian, "Sar-aircraft-1.0: High-resolution sar aircraft detection and recognition dataset," *J. Radars*, vol. 12, no. 4, pp. 906–922, 2023.

[51] J. Chen, Y. Shen, Y. Liang, Z. Wang, and Q. Zhang, "Yolo-sad: An efficient sar aircraft detection network," *Applied Sciences*, vol. 14, no. 7, p. 3025, 2024.

[52] Y. Zhu, K. Yuan, W. Zhong, and L. Xu, "Spatial–spectral convnext for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 5453–5463, 2023.

[53] Z. Ye, X. Xiao, and H. Wang, "Convolutional modulated scattering feature network for aircraft classification in sar images," in *IGARSS 2024 - IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2024, pp. 9329–9332.

[54] C. Zhao, S. Zhang, R. Luo, S. Feng, and G. Kuang, "Scattering features spatial-structural association network for aircraft recognition in sar images," *IEEE Geoscience and Remote Sensing Letters*, vol. 20, pp. 1–5, 2023.

[55] L. Du, R. Zhang, and X. Wang, "Overview of two-stage object detection algorithms," in *Journal of Physics: Conference Series*. IOP Publishing, 2020, p. 012033.

[56] J. Li, Z. Yu, L. Yu, P. Cheng, J. Chen, and C. Chi, "A comprehensive survey on sar atr in deep-learning era," *Remote Sensing*, vol. 15, no. 5, p. 1454, 2023.

[57] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023.

[58] Y. Ya, H. Pan, Z. Jing, X. Ren, and L. Qiao, "Fusion object detection with convolutional neural network," University of Toronto Institute for Aerospace Studies, Tech. Rep. Technical Note No. 71, 2019.

[59] Y. Xiao, Z. Tian, J. Yu *et al.*, "A review of object detection based on deep learning," *Multimedia Tools and Applications*, vol. 79, pp. 23 729–23 791, 2020.

[60] D. Malmgren-Hansen, A. Kusk, J. Dall, A. A. Nielsen, R. Engholm, and H. Skriver, "Improving sar automatic target recognition models with transfer learning from simulated data," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 9, pp. 1484–1488, 2017.

[61] Z. Sordo, E. Chagnon, and D. Ushizima, "A review on generative ai for text-to-image and image-to-image generation and implications to scientific images," *arXiv preprint arXiv:2502.21151*, 2025.