

Sistema de Detección para la Evasión de Obstáculos en Drones Autónomos Mediante Filtrado por Altitud y Estimación de Profundidad

Jose Quintana

Visión por Computador Avanzada

Universidad Tecnológica de Pereira

Pereira, Colombia

jose.quintana@utp.edu.co

Resumen—Este proyecto presenta un sistema de detección y evasión de obstáculos en para drones autónomos, basado en un pipeline que combina redes neuronales profundas, estimación de profundidad y filtrado adaptativo según altitud y distancia. La solución integra modelos avanzados como YOLOv11 para detección de objetos y MiDaS para estimación de profundidad, entrenados con datasets preprocesados como VisDrone, AU-AIR y CityScapes.

El objetivo principal del sistema es reducir el costo computacional asociado a la navegación autónoma en entornos urbanos. Esto se logra mediante la aplicación de filtros basados en altitud y profundidad, que permiten identificar únicamente los objetos que, por sus características espaciales, representan posibles obstáculos, evitando así procesar elementos irrelevantes. Esta optimización no solo mejora la eficiencia del sistema, sino que también garantiza una respuesta más rápida y precisa en escenarios complejos.

Palabras clave—Drones, Detección de Objetos, Estimación de Profundidad, Filtrado por Altitud, Navegación Autónoma, Evasión de Obstáculos, Visión por Computador.

I. INTRODUCCIÓN

En los últimos años, los drones autónomos han revolucionado una amplia gama de industrias, abarcando aplicaciones como la vigilancia, la agricultura de precisión, la entrega de paquetes y las operaciones de rescate. Estas tecnologías, capaces de operar en entornos complejos y de ejecutar tareas con autonomía, han demostrado ser herramientas esenciales para la recopilación de datos y la ejecución de actividades críticas [1]. Sin embargo, garantizar la navegación segura en escenarios dinámicos y con alta densidad de obstáculos sigue siendo un desafío significativo, ya que los drones deben ser capaces de detectar, clasificar y evitar obstáculos en tiempo real, sin comprometer su eficiencia ni seguridad [2].

Aunque las soluciones tradicionales como las cámaras RGB y los sensores LiDAR han desempeñado un papel fundamental en la detección de obstáculos, estas tecnologías presentan limitaciones considerables. Las cámaras RGB, por ejemplo, son altamente dependientes de las condiciones de iluminación y carecen de la capacidad para estimar distancias con precisión; por otro lado, los sistemas LiDAR, aunque muy precisos, suelen ser costosos y voluminosos, lo que

dificulta su integración en drones pequeños o medianos [3]. En este contexto, surge la necesidad de desarrollar sistemas más accesibles y ligeros que combinen múltiples fuentes de información, como imágenes y datos de profundidad, para superar estas barreras y mejorar la percepción del entorno.

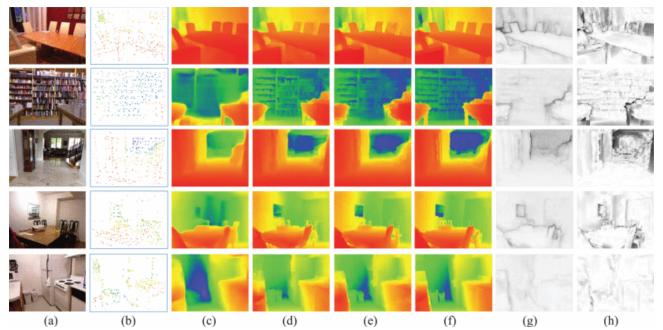


Fig. 1. Mapas de Información Visual en Drones [4]

El proyecto detallado en el presente documento responde a las necesidades mencionadas, mediante el diseño de un sistema autónomo que integra técnicas avanzadas de visión por computador con datos de sensores para optimizar la navegación y la evasión de obstáculos. Al emplear YOLOv11, un modelo de detección de objetos optimizado para tiempo real, complementado con mapas de profundidad y datos de altitud [5], el sistema es capaz de filtrar objetos irrelevantes en función de su altura y distancia, lo que reduce la carga computacional, mejora la precisión en la toma de decisiones y garantiza una mayor seguridad durante las operaciones del dron.

La relevancia de esta propuesta radica en su capacidad para transformar la manera en que los drones interactúan con su entorno, al integrar un análisis detallado de la distancia y la altitud en el proceso de detección. Esto no solo incrementa la precisión y eficiencia del sistema, sino que también abre nuevas oportunidades para su aplicación en entornos urbanos densos, áreas rurales o escenarios dinámicos [7]–[9], donde la capacidad de reacción inmediata y la adaptabilidad son esenciales. Así, el sistema propuesto no solo atiende las demandas actuales de navegación autónoma, sino que también establece un marco para futuros desarrollos que busquen una

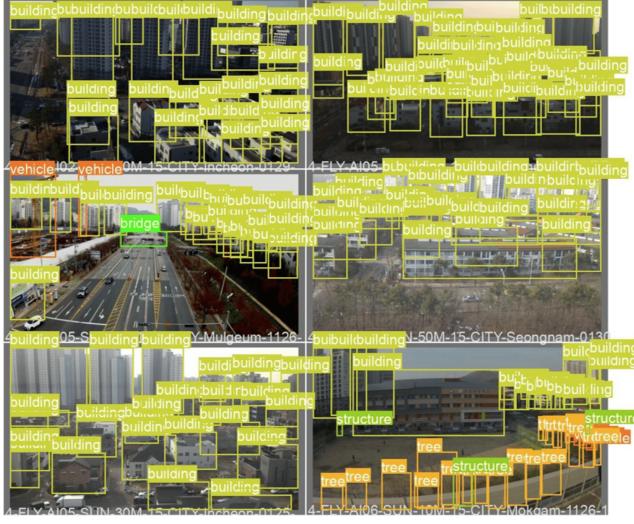


Fig. 2. Información Visual en drones basada en Altitud [6]

mayor integración de tecnologías accesibles y de bajo costo.

Este documento aborda de manera integral el desafío de la navegación autónoma para drones, proporcionando el contexto necesario para comprender la importancia de esta problemática en aplicaciones prácticas, donde la seguridad y la eficiencia son fundamentales. Se detalla el modelo propuesto, explicando su arquitectura y los hiperparámetros seleccionados, junto con el enfoque utilizado durante el entrenamiento, posteriormente, se presentan los resultados obtenidos, y se discuten posibles mejoras que podrían potenciar aún más las capacidades del sistema en escenarios diversos y dinámicos. Finalmente, se concluye con una reflexión sobre las consideraciones éticas asociadas a la implementación de este modelo, analizando tanto los riesgos como las implicaciones sociales y legales que su adopción puede conllevar.

II. DESCRIPCIÓN DEL MODELO

El modelo desarrollado para este proyecto integra técnicas avanzadas de visión por computador y redes neuronales profundas, con el objetivo de facilitar la detección, el filtrado y la evasión de obstáculos en drones autónomos. Su diseño combina modelos de detección de objetos, como YOLOv11 [5], [10], con redes para la estimación de mapas de profundidad, como MiDaS [11], en un pipeline optimizado para procesar datos en tiempo real. Este enfoque permite mejorar la precisión y velocidad en la toma de decisiones, asegurando un desempeño confiable en entornos complejos.

El sistema se entrena utilizando conjuntos de datos especializados que representan entornos dinámicos y urbanos, como VisDrone [9] y AU-AIR [7], lo que garantiza su adaptabilidad a diversas condiciones. La arquitectura propuesta implementa técnicas de preprocesamiento de imágenes, detección de objetos y filtrado por altitud y distancia, integrando los resultados en una representación gráfica que facilita la navegación autónoma. A continuación, se detallan las partes clave del modelo.

A. Preparación y Preprocesamiento de Datos

1) *Organización de Datasets:* Para este proyecto, se emplearon dos datasets principales: VisDrone y AU-AIR. Además, se llevaron a cabo análisis y pruebas adicionales utilizando el dataset CityScapes, que contiene información de mapas de profundidad. Esta característica resultó fundamental para evaluar y validar los procedimientos de estimación de profundidad, sirviendo como referencia para extender dichas técnicas a los datasets que no incluyen mapas de profundidad asociados, como VisDrone y AU-AIR.

Cada dataset fue sometido a un proceso de preprocesamiento cuidadoso para garantizar su compatibilidad con los requisitos del modelo. En el caso de VisDrone y AU-AIR, las anotaciones originales se transformaron al formato YOLO, que utiliza coordenadas normalizadas para representar las cajas delimitadoras (*bounding boxes*) y las clases correspondientes de los objetos detectados. Este formato es ampliamente adoptado para la detección de objetos debido a su eficiencia y simplicidad.

Por otro lado, el dataset CityScapes mantuvo sus etiquetas de segmentación y mapas de profundidad en formato .npy, optimizado para su integración con la red MiDaS. Este formato permite un acceso rápido y eficiente a los datos necesarios para la estimación de profundidad, lo que facilita su uso en modelos de percepción visual para entornos complejos. En conjunto, la organización y adaptación de estos datasets garantizan un pipeline robusto para la detección, filtrado y navegación autónoma de los drones en escenarios dinámicos.

2) *Conversión a Formato YOLO:* Las anotaciones de los datasets se transformaron al formato estándar de YOLO:

$$(c, x_n, y_n, w_n, h_n)$$

Donde:

- c : Clase del objeto.
- x_n, y_n : Coordenadas normalizadas del centro del bounding box.
- w_n, h_n : Ancho y alto normalizados del bounding box.

Esta conversión asegura la compatibilidad con los modelos de detección utilizados, mejorando la eficiencia durante el entrenamiento y la inferencia. Además, las imágenes fueron ajustadas a una resolución de 640×640 píxeles.

3) *Aumentación de Datos:* Para aumentar la variabilidad del sistema de detección, se aplicaron técnicas de aumentación como:

- Rotaciones aleatorias.
- Variaciones en la iluminación y el contraste.
- Desplazamientos y recortes aleatorios.

Estas técnicas se aplicaron tanto en las imágenes como en las etiquetas correspondientes, lo que permitió aumentar la robustez del modelo frente a diferentes condiciones ambientales.

B. Entrenamiento del Modelo YOLO

1) *Configuración del Entrenamiento:* El entrenamiento se realizó utilizando un modelo YOLOv11 preentrenado, ajustado con los siguientes hiperparámetros:

- **Número de épocas:** 20.
- **Tamaño del lote:** 32.
- **Resolución de entrada:** 640×640 .
- **Optimizador:** Adam con precisión mixta activada (AMP) para acelerar el proceso de entrenamiento.
- **Dataset:** Dividido en un 80% para entrenamiento, 10% para validación y 10% para prueba.

2) *Definición de los Archivos YAML:* Cada dataset fue configurado mediante un archivo YAML que especifica las rutas de los conjuntos de entrenamiento, validación y prueba, así como el número de clases y sus nombres. A continuación, se presenta un ejemplo del archivo YAML utilizado para el dataset VisDrone:

```

train: /content/split_dataset/VisDrone2019
    /train/images
val: /content/split_dataset/VisDrone2019
    /val/images
test: /content/split_dataset/VisDrone2019
    /test/images
nc: 12
names:
    - Ignore
    - Pedestrian
    - People
    - Bicycle
    - Car
    - Van
    - Truck
    - Tricycle
    - Awning-Tricycle
    - Bus
    - Motor
    - Others
  
```

Para optimizar el entrenamiento y mejorar la detección en el sistema, se utilizó un dataset preprocesado denominado *Drone_Detection* [12], el cual combina imágenes de los datasets VisDrone y AU-AIR, previamente adaptadas al formato YOLO. Además, al integrar imágenes de ambos datasets, se logró aumentar la diversidad y variabilidad del conjunto de datos, mejorando la capacidad del modelo para generalizar en escenarios urbanos complejos y en entornos dinámicos.

Un aspecto destacado de este dataset es la introducción de nuevas clases específicas para tareas relacionadas con el entorno de drones, con etiquetas ajustadas para reconocer objetos como `['Building', 'Fence', 'House', 'Pole', 'Street Sign', 'Tree']`. Estas nuevas clases amplían las capacidades del modelo para detectar elementos relevantes en el entorno y adaptarse a un rango más amplio de aplicaciones. El dataset preprocesado incluye todas las configuraciones necesarias para su uso inmediato en tareas de entrenamiento, validación y prueba, lo que simplifica el pipeline y garantiza un mejor rendimiento.

C. Arquitectura por Bloques

El diseño del sistema propuesto se estructura en tres bloques fundamentales que trabajan en conjunto para la percepción y filtrado de objetos en tiempo real. Este diseño modular permite un flujo eficiente de datos desde la captura inicial hasta la generación de resultados listos para la toma de decisiones. A continuación, se describe la arquitectura en detalle.

D. Entrada de Datos

• Captura de Imágenes:

- *Fuente:* Cámaras RGB montadas en el dron.
- *Resolución:* Imágenes en formato 640x480 píxeles, ajustadas para compatibilidad con los modelos YOLO y MiDaS.
- *Frecuencia:* Procesamiento en tiempo real a 30 FPS para garantizar una respuesta rápida.

E. Procesamiento

• Detección de Objetos:

- *Modelo:* YOLOv11 preentrenado.
- *Salida:*
 - * Coordenadas de bounding boxes (x, y, w, h).
 - * Clase del objeto (C).
 - * Confianza asociada (p).
- *Optimización:* Uso de GPU para procesamiento acelerado.

• Generación de Mapas de Profundidad:

- *Modelo:* MiDaS para estimación monocular de profundidad.
- *Salida:* Mapas de profundidad (D) con valores normalizados en el rango $[0, 1]$.
- *Postprocesamiento:* Normalización y suavizado de los mapas de profundidad para eliminar artefactos en áreas poco iluminadas.

• Filtrado Combinado:

- *Por Altitud:*
 - * Comparación entre la altitud actual del dron (A) y las clases de objetos permitidas (C_{alt}).
 - * Filtrado basado en reglas específicas para cada altitud.
- *Por Distancia:*
 - * Cálculo del promedio de profundidad (d_{avg}) en la región del objeto detectado.
 - * Eliminación de objetos cuya distancia supere el umbral predefinido ($d_{threshold}$).

F. Salida de Datos

• Resultados Filtrados:

- Lista de objetos válidos con:
 - * Coordenadas finales de bounding boxes.
 - * Clase asociada y confianza.
 - * Información de profundidad.
- Representación gráfica en tiempo real, con visualización de objetos relevantes y descartados.

G. Modelado Matemático

El sistema desarrollado combina detección de objetos, estimación de profundidad y filtros basados en altitud y distancia para garantizar la evasión autónoma de obstáculos en drones. A continuación, se describen los modelos matemáticos que sustentan cada componente.

1) *Detección de Objetos*: El modelo YOLO (*You Only Look Once*) realiza detección de objetos generando una salida estructurada para cada objeto detectado O_i . Esta salida se describe matemáticamente como:

$$O_i = \{C, (x, y, w, h), p\}$$

Donde:

- C : Clase del objeto detectado, representada por un índice entero que identifica categorías predefinidas, como vehículos, peatones, edificios, entre otros.
- (x, y, w, h) : Coordenadas normalizadas del *bounding box*, donde:
 - x, y : Coordenadas del centro del *bounding box*, normalizadas entre 0 y 1 con respecto al ancho (W) y alto (H) de la imagen.
 - w, h : Ancho y alto del *bounding box*, también normalizados en el rango $[0, 1]$.
- p : Confianza del modelo en la predicción, un valor entre 0 y 1 que indica la probabilidad de que la detección sea correcta.

El proceso de entrenamiento de YOLO minimiza una pérdida total L que combina tres componentes principales:

$$L = L_{\text{loc}} + L_{\text{conf}} + L_{\text{class}}$$

Donde:

- L_{loc} : Pérdida de localización, mide la discrepancia entre las coordenadas predichas (x, y, w, h) y las anotaciones reales.
- L_{conf} : Pérdida de confianza, penaliza las predicciones erróneas y refuerza la precisión en la estimación de p .
- L_{class} : Pérdida de clasificación, evalúa la precisión en la predicción de la clase C .

2) *Mapas de Profundidad*: El modelo MiDaS estima un mapa de profundidad D para una imagen de entrada I :

$$D = f_{\text{MiDaS}}(I), \quad D \in \mathbb{R}^{H \times W}, \quad 0 \leq D \leq 1$$

Donde:

- I : Imagen de entrada en formato RGB.
- D : Mapa de profundidad estimado, donde cada valor $D(x, y)$ representa una profundidad relativa en la escala $[0, 1]$, con 0 indicando proximidad y 1 indicando mayor distancia relativa.
- H, W : Altura y ancho de la imagen.

Para cada objeto detectado, se define una región de interés R en el mapa de profundidad, correspondiente al área delimitada por su *bounding box*. La distancia promedio d_{avg} dentro de esta región se calcula como:

$$d_{\text{avg}} = \frac{1}{|R|} \sum_{(x,y) \in R} D(x, y)$$

Donde $|R|$ es el número total de píxeles en la región R .

3) *Filtrado por Altitud y Distancia*: El filtrado se realiza para descartar objetos que no cumplen con ciertos criterios basados en altitud y distancia:

Altitud: Cada objeto detectado O_i se clasifica en función de un nivel de altitud A del dron, utilizando los datos de altitud proporcionados por el conjunto de datos AU-AIR. Este nivel de altitud permite filtrar los objetos detectados, aceptando únicamente aquellos que pertenezcan al conjunto de clases permitidas $C_{\text{alt}}(A)$, definido para cada rango de altitud:

$$O_i \in C_{\text{alt}}(A)$$

Donde $C_{\text{alt}}(A)$ es el conjunto de clases relevantes para la altitud A .

Distancia: El filtrado adicional descarta objetos cuya distancia promedio d_{avg} excede un umbral predefinido $d_{\text{threshold}}$:

$$d_{\text{avg}} \leq d_{\text{threshold}}$$

Este umbral asegura que el dron priorice objetos más cercanos que representan un riesgo inmediato.

La integración de estos modelos permite priorizar detecciones relevantes, mejorando la capacidad del dron para navegar y evitar obstáculos de manera autónoma en tiempo real. Los resultados del filtrado se visualizan mediante *bounding boxes*, donde los objetos aceptados se resaltan en verde y los rechazados en rojo, asegurando una interpretación clara para análisis posteriores.

III. RESULTADOS

A. VisDrone2019 con YOLOv11n

El entrenamiento con el modelo YOLOv11n para el dataset VisDrone2019 se realizó a lo largo de 20 épocas, logrando resultados relevantes en la detección de objetos en entornos urbanos y rurales. A continuación, se presentan las métricas obtenidas, visualizaciones de las predicciones y un análisis detallado de los resultados.

1) *Métricas de Entrenamiento*: En términos de desempeño, las métricas clave indican un progreso constante a lo largo del entrenamiento. La Tabla I muestra las estadísticas principales calculadas durante las 20 épocas:

- **Precisión promedio**: 0.228, alcanzando un máximo de 0.3759.
- **Recall promedio**: 0.0973, con un mínimo de 0.0468.
- **mAP@50**: Promedio de 0.0786, con un máximo de 0.11374.
- **mAP@50-95**: Promedio de 0.0386, reflejando desafíos en umbrales más estrictos.

TABLE I
MÉTRICAS DEL ENTRENAMIENTO PARA VISDRONE2019

Métrica	Media	Desv. Estándar	Mínimo	Máximo
Precisión (B)	0.228	0.087	0.010	0.376
Recall (B)	0.097	0.028	0.047	0.133
mAP@50 (B)	0.079	0.028	0.017	0.114
mAP@50-95 (B)	0.039	0.015	0.008	0.058

En la Figura 3, se presenta la evolución de las métricas clave a lo largo de las épocas, destacándose un comportamiento

ascendente en precisión y *recall*, aunque con desafíos en el mAP@50-95.

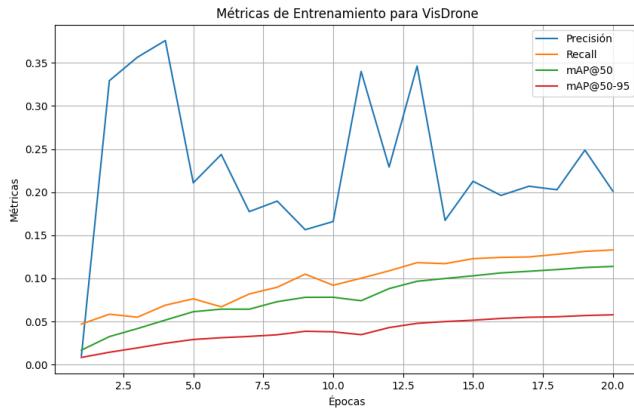


Fig. 3. Evolución de las métricas durante el entrenamiento en VisDrone2019.

2) *Visualización de Predicciones*: El desempeño del modelo se evalúa visualmente en las imágenes de validación, como se muestra en las Figuras 4, 5, y 6. Estas imágenes reflejan la capacidad del modelo para detectar vehículos, peatones y bicicletas en entornos urbanos.



Fig. 4. Predicciones en imágenes de validación para VisDrone2019 (Batch 0).

3) *Curvas de Métricas*: Las curvas de desempeño permiten un análisis más detallado del modelo en términos de precisión, *recall*, y confianza en las predicciones:

- **Curva de Precisión-Confianza (Figura 7)**: Ilustra la relación entre confianza en las predicciones y precisión.
- **Curva de Precisión-Recall (Figura 8)**: Muestra el balance entre precisión y *recall* a diferentes umbrales.
- **Curva de Recall-Confianza (Figura 9)**: Evalúa el rendimiento en términos de *recall* basado en la confianza de las predicciones.

4) *Discusión de Resultados*: Los resultados del modelo YOLOv11n en VisDrone2019 muestran fortalezas y oportunidades de mejora:

- **Clases Predominantes**: El modelo demostró un desempeño sólido en la identificación de clases predominantes como vehículos y peatones.

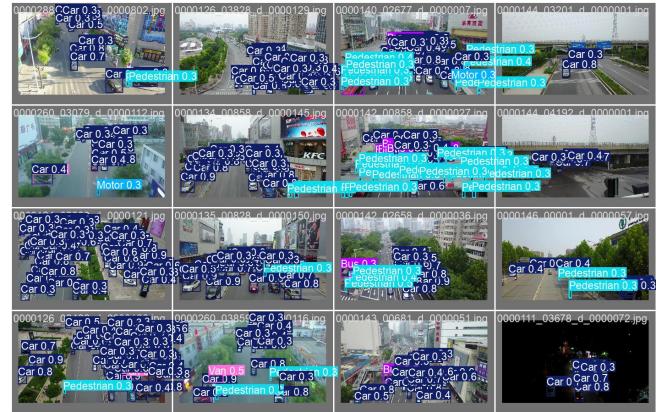


Fig. 5. Predicciones en imágenes de validación para VisDrone2019 (Batch 1).



Fig. 6. Predicciones en imágenes de validación para VisDrone2019 (Batch 2).

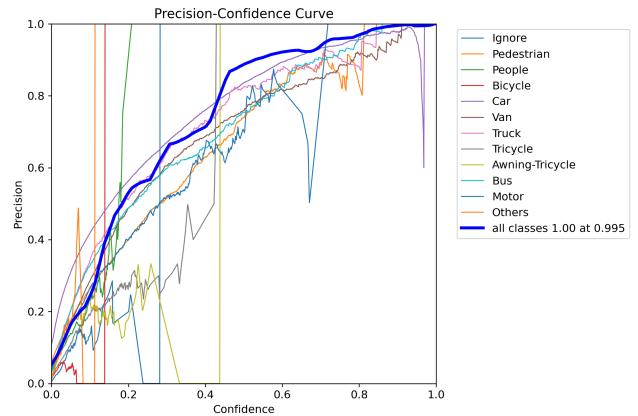


Fig. 7. Curva de Precisión-Confianza para VisDrone2019.

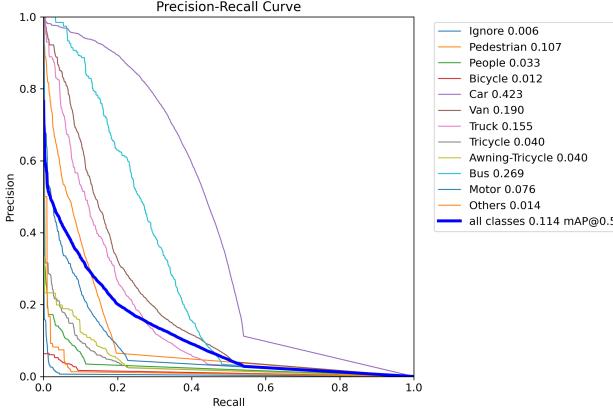


Fig. 8. Curva de Precisión-*Recall* para VisDrone2019.

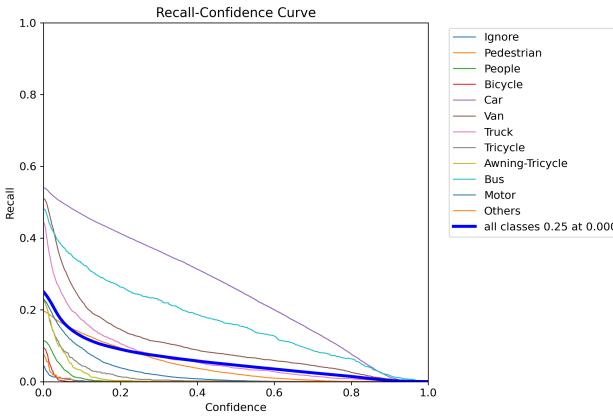


Fig. 9. Curva de *Recall*-Confianza para VisDrone2019.

- Desafíos en Clases Minoritarias:** Clases menos representadas, como bicicletas y motocicletas, tuvieron un rendimiento inferior, lo cual es evidente en las curvas de precisión-*recall*.
- mAP a Diferentes Niveles de IoU:** El contraste entre mAP@50 (0.1137) y mAP@50-95 (0.0576) refleja la necesidad de mejorar en umbrales más estrictos de detección.
- Entornos Urbanos Densos:** El alto nivel de complejidad del dataset presentó retos para el modelo, como mayor ruido en las imágenes y clases múltiples por escena.

Para mejorar el desempeño del modelo, se podrían aplicar estrategias como:

- Incrementar la diversidad del dataset con técnicas de *data augmentation*.
- Ajustar los hiperparámetros del modelo para optimizar el equilibrio entre precisión y *recall*.
- Implementar un posprocesamiento más robusto para reducir falsos positivos en clases menos representadas.

En general, el modelo mostró resultados prometedores para aplicaciones de monitoreo urbano, aunque aún enfrenta desafíos en la detección de clases minoritarias y entornos densos.

B. AU-AIR con YOLOv11n

El dataset AU-AIR contiene datos de drones en ambientes semiurbanos. El modelo YOLOv11n fue entrenado con este dataset por 20 épocas, produciendo resultados satisfactorios en términos de detección de vehículos, peatones y otros objetos.

1) *Métricas del Entrenamiento:* Durante el entrenamiento, el modelo alcanzó resultados consistentes en términos de precisión, *recall*, y *mean Average Precision* (mAP). En la Tabla II se resumen las métricas principales obtenidas a lo largo de las 20 épocas de entrenamiento:

- La precisión promedio alcanzó un valor de 0.456, con un máximo de 0.578.
- El *recall* promedio fue de 0.264, con un rango entre 0.093 y 0.331.
- El mAP@50 alcanzó un máximo de 0.337, mientras que el mAP@50-95 mostró un promedio de 0.135.

TABLE II
MÉTRICAS DEL ENTRENAMIENTO PARA AU-AIR

Métrica	Media	Desv. Estándar	Mínimo	Máximo
Precisión (B)	0.456	0.119	0.087	0.578
Recall (B)	0.264	0.062	0.093	0.331
mAP@50 (B)	0.250	0.078	0.039	0.337
mAP@50-95 (B)	0.135	0.048	0.016	0.194

En la Figura 10 se ilustra la evolución de las métricas clave a lo largo de las 20 épocas de entrenamiento, mostrando una mejora progresiva en precisión y mAP.

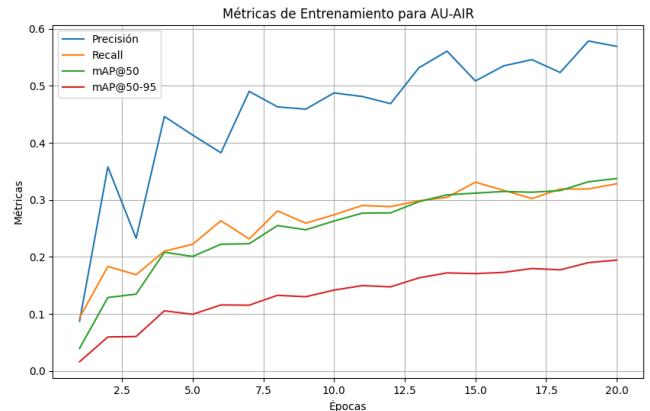


Fig. 10. Evolución de las métricas durante el entrenamiento en AU-AIR.

2) *Visualización de Predicciones:* Las imágenes de validación permiten evaluar el desempeño del modelo en entornos semiurbanos. Las Figuras 11, 12, y 13 muestran ejemplos de predicciones realizadas por el modelo, destacando su capacidad para identificar múltiples objetos como vehículos y humanos.

3) *Curvas de Métricas:* Las curvas de rendimiento del modelo permiten analizar el balance entre precisión, recall, y confianza en las predicciones. Estas se presentan en las Figuras 14, 15, y 16.

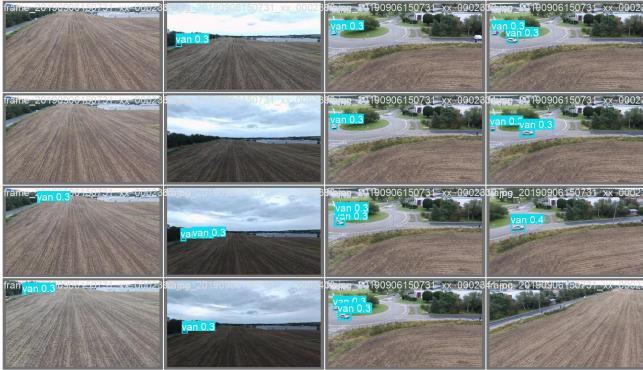


Fig. 11. Predicciones en imágenes de validación para AU-AIR (Batch 0).



Fig. 12. Predicciones en imágenes de validación para AU-AIR (Batch 1).

- **Curva de Precisión-Confianza:** Ilustra cómo varía la precisión a medida que se ajusta el umbral de confianza en las predicciones.
- **Curva de Precisión-Recall:** Muestra el balance entre precisión y recall en diferentes niveles de confianza.
- **Curva de Recall-Confianza:** Evalúa la relación entre el recall y la confianza en las predicciones realizadas por el modelo.

4) *Discusión de Resultados:* El modelo YOLOv11n mostró un desempeño robusto en el dataset AU-AIR, con resultados destacables en clases predominantes como autos y vans. Sin embargo, se identificaron áreas de oportunidad:

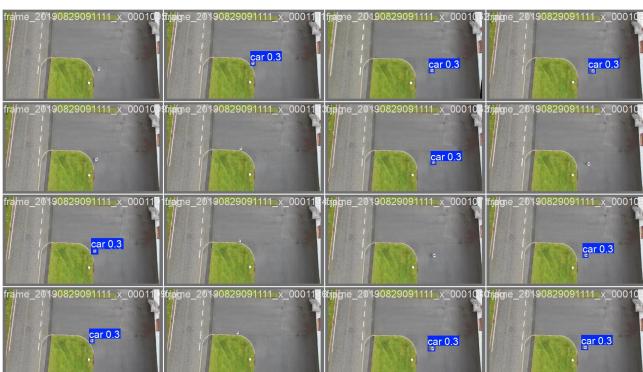


Fig. 13. Predicciones en imágenes de validación para AU-AIR (Batch 2).

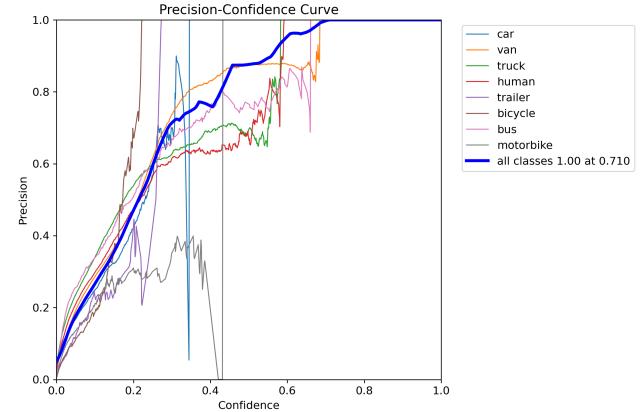


Fig. 14. Curva de Precisión-Confianza para AU-AIR.

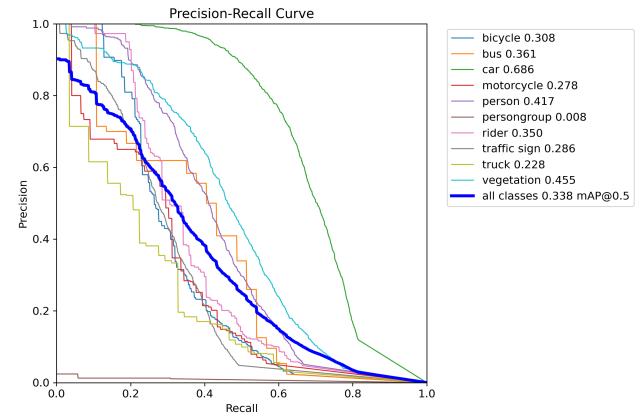


Fig. 15. Curva de Precisión-Recall para AU-AIR.

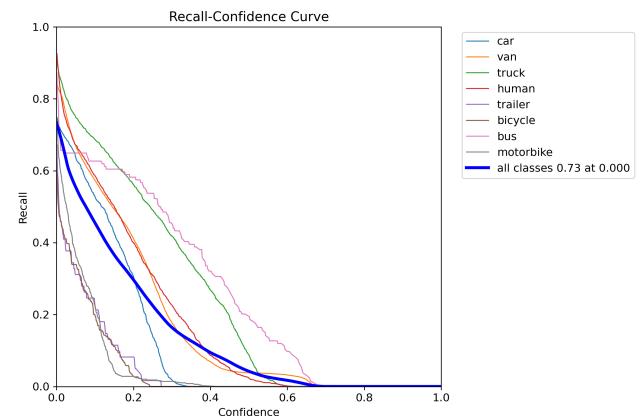


Fig. 16. Curva de Recall-Confianza para AU-AIR.

- Clases Predominantes:** El modelo logró alta precisión en vehículos y humanos, con una buena representación en las predicciones.
- Clases Menores:** Las clases menos representadas, como bicicletas y motocicletas, presentaron menor desempeño, evidenciado en la dispersión de las curvas de precisión-recall.
- mAP a Diferentes Niveles de IoU:** Aunque el mAP@50 alcanzó un valor elevado (0.337), el mAP@50-95 reflejó desafíos en umbrales de detección más estrictos.
- Efecto del Contexto Semiurbano:** Los datos semiurbanos del dataset presentaron menor ruido en comparación con datasets urbanos como VisDrone2019, contribuyendo a un mejor rendimiento general.

Estos hallazgos sugieren que un enfoque en técnicas de aumento de datos (*data augmentation*) y ajustes en los hiperparámetros del modelo podrían mejorar el rendimiento en clases minoritarias. En general, el modelo demostró un desempeño sólido en aplicaciones semiurbanas, como vigilancia aérea o monitoreo de tráfico.

C. CityScapes con YOLOv11n

El dataset CityScapes es un conjunto de datos ampliamente utilizado para tareas de segmentación y detección de objetos en entornos urbanos. El modelo YOLOv11n fue entrenado con este dataset a lo largo de 20 épocas, generando métricas clave y resultados visuales que se presentan a continuación.

1) *Métricas del Entrenamiento:* Durante el entrenamiento, el modelo alcanzó resultados consistentes en términos de precisión, *recall* y *mean Average Precision* (mAP). En la Tabla III se presentan las estadísticas principales calculadas durante las 20 épocas. Entre los puntos destacados se encuentran:

- La precisión promedio alcanzó un valor de 0.456, con un máximo de 0.578.
- El *recall* promedio fue de 0.264, con un rango entre 0.093 y 0.331.
- El mAP@50 alcanzó un valor máximo de 0.337, mientras que el mAP@50-95 se estabilizó en un promedio de 0.134.
- Las pérdidas de validación reflejan una mejora progresiva, aunque con fluctuaciones en clases menos representadas.

TABLE III
MÉTRICAS DEL ENTRENAMIENTO PARA CITYSCAPES

Métrica	Media	Desv. Estándar	Mínimo	Máximo
Precisión (B)	0.456	0.120	0.087	0.578
Recall (B)	0.264	0.062	0.093	0.331
mAP@50 (B)	0.250	0.078	0.040	0.337
mAP@50-95 (B)	0.134	0.048	0.016	0.194
Val Box Loss	1.657	0.223	1.452	2.331
Val Class Loss	1.816	1.239	1.137	6.300

En la Figura 17 se ilustra la evolución de las métricas clave durante las 20 épocas de entrenamiento, reflejando una mejora gradual en las métricas de precisión y mAP.

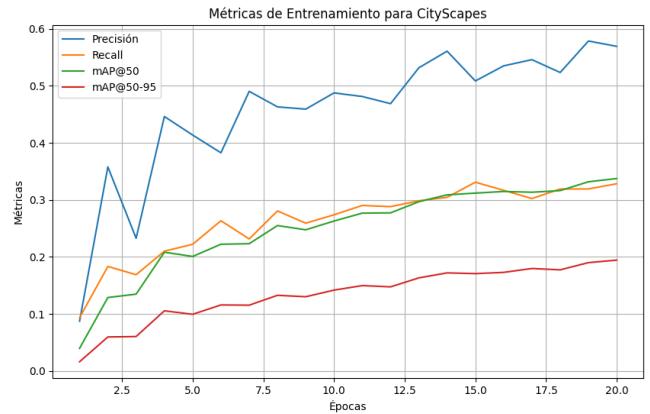


Fig. 17. Evolución de las métricas durante el entrenamiento en CityScapes.

2) *Visualización de Predicciones:* Las imágenes de validación permiten evaluar el desempeño del modelo en entornos urbanos complejos. Las Figuras 18, 19, y 20 muestran ejemplos representativos de predicciones realizadas por el modelo, destacando su capacidad para identificar múltiples objetos como vehículos, señales de tráfico y peatones.



Fig. 18. Predicciones en imágenes de validación para CityScapes (Batch 0).

3) *Curvas de Métricas:* Las curvas de rendimiento del modelo permiten analizar el balance entre precisión, recall y confianza en las predicciones. Estas se presentan en las Figuras 21, 22, y 23.

- Curva de Precisión-Confianza:** Ilustra cómo varía la precisión a medida que se ajusta el umbral de confianza en las predicciones.
- Curva de Precisión-Recall:** Muestra el balance entre precisión y recall en diferentes niveles de confianza.



Fig. 19. Predicciones en imágenes de validación para CityScapes (Batch 1).

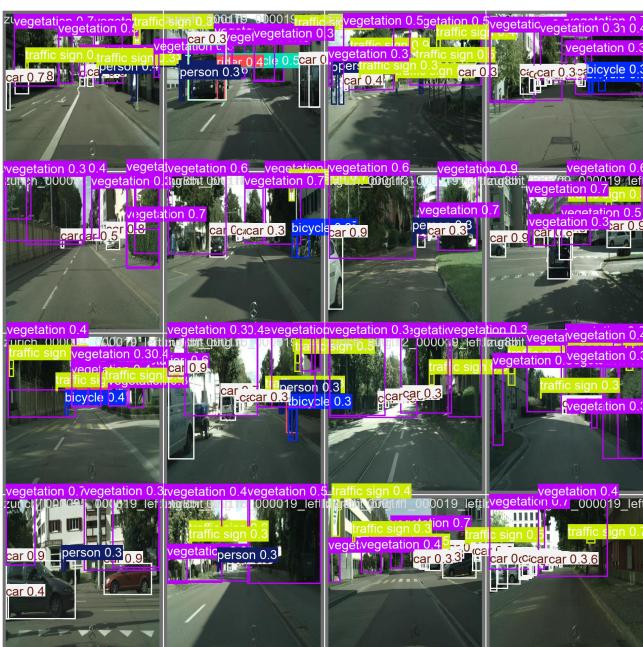


Fig. 20. Predicciones en imágenes de validación para CityScapes (Batch 2).

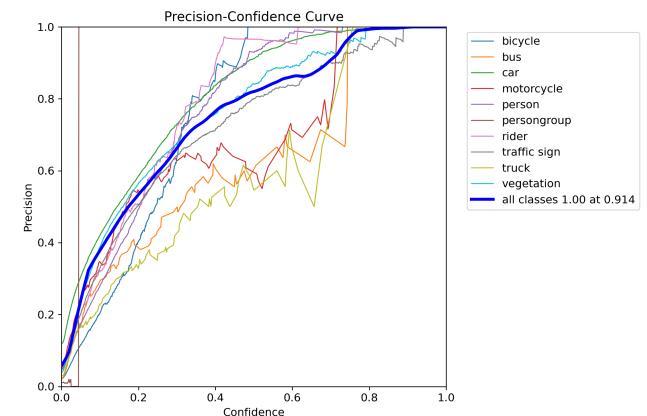


Fig. 21. Curva de Precisión-Confianza para CityScapes.

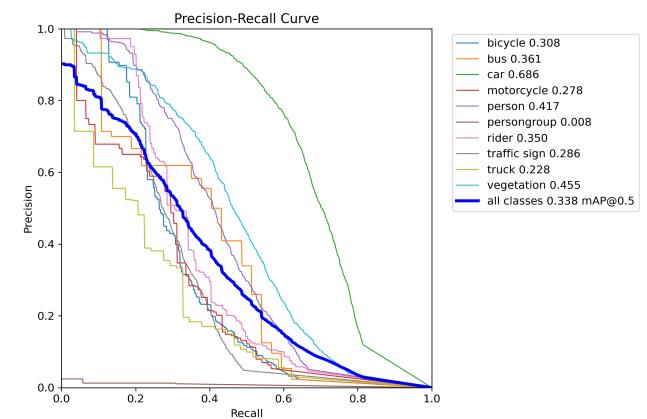


Fig. 22. Curva de Precisión-Recall para CityScapes.

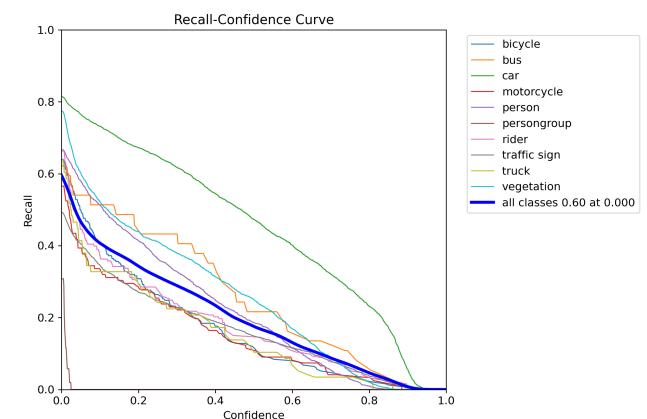


Fig. 23. Curva de Recall-Confianza para CityScapes.

- **Curva de Recall-Confianza:** Evalúa la relación entre el recall y la confianza en las predicciones realizadas por el modelo.

4) *Discusión de Resultados:* El modelo YOLOv11n mostró un desempeño competitivo en el dataset CityScapes, con resultados destacables en clases predominantes como vehículos y señales de tráfico. Sin embargo, se observaron áreas de oportunidad en las siguientes categorías:

- **Clases Minoritarias:** Las clases menos representadas, como bicicletas y motocicletas, mostraron una menor precisión debido a la limitada cantidad de datos disponibles.
- **Relación Precisión-Recall:** Aunque el modelo alcanzó altos valores de precisión, el recall fue relativamente bajo, lo cual indica que algunos objetos no fueron detectados.
- **Optimización de Pérdidas:** A pesar de las mejoras en las pérdidas de validación, el modelo mostró fluctuaciones significativas, particularmente en las clases menos frecuentes.

Estos resultados subrayan la importancia de aumentar la diversidad del dataset y ajustar los hiperparámetros del modelo para mejorar su rendimiento en escenarios urbanos complejos. Además, la implementación de estrategias de *data augmentation* podría contribuir a un mejor balance entre precisión y recall en clases minoritarias.

D. Filtrado de Objetos Segundo Altitud y Profundidad

En este apartado se presentan los resultados obtenidos tras aplicar los filtros de altitud y profundidad al sistema de detección de objetos. Los bounding boxes en verde representan los objetos aceptados que cumplen con los criterios establecidos, mientras que los bounding boxes en rojo indican los objetos descartados por no satisfacer las restricciones de altitud y/o profundidad. Este proceso de filtrado es crucial para garantizar una detección más robusta y precisa en entornos urbanos complejos.

Estos resultados subrayan la importancia de aumentar la diversidad del dataset y ajustar los hiperparámetros del modelo para mejorar su rendimiento. Además, la implementación de estrategias de *data augmentation* podría contribuir a un mejor balance entre precisión y recall en clases minoritarias, mejorando así la capacidad del sistema para enfrentar escenarios variados y desafiantes.

En la Figura 24, se muestran ejemplos de imágenes con los resultados del filtrado:

Los bounding boxes en verde corresponden a los objetos aceptados, mientras que los bounding boxes en rojo son los objetos descartados. A la derecha, se muestra el mapa de profundidad estimado.

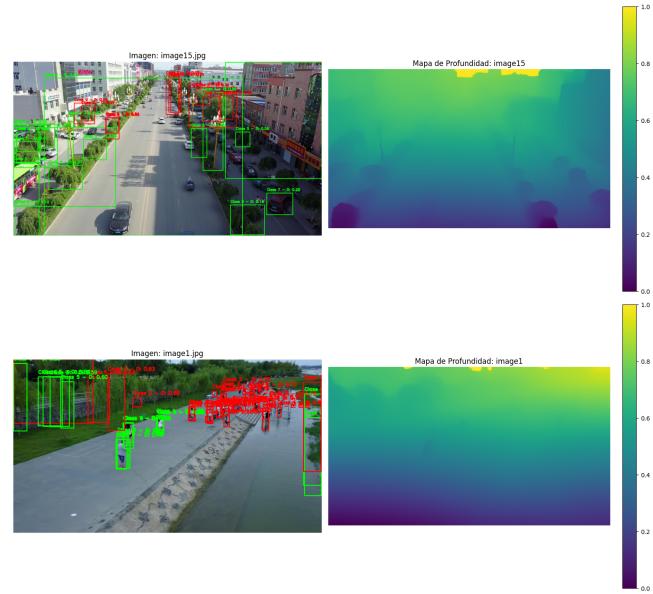


Fig. 24. Resultados del filtrado de objetos según altitud y profundidad.

Para mejorar la comprensión, también se presentan las mismas imágenes destacando únicamente los objetos aceptados (bounding boxes verdes), como se observa en la Figura 25:

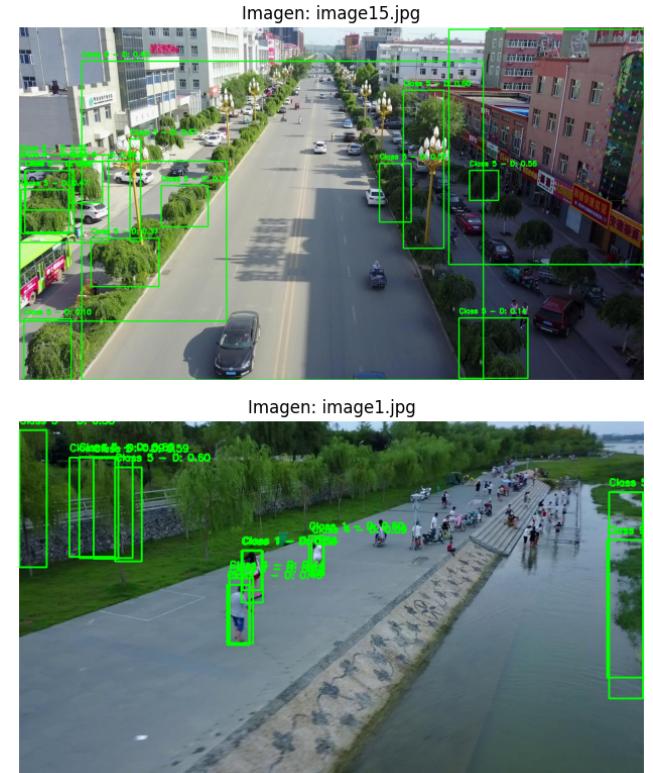


Fig. 25. Resultados del filtrado mostrando únicamente los objetos aceptados (bounding boxes en verde).

Estos resultados destacan la capacidad del sistema para realizar un filtrado efectivo, pero también resaltan la necesidad

de mejorar el modelo para manejar clases minoritarias y aumentar la diversidad del dataset para escenarios más complejos.

IV. TRABAJOS FUTUROS

El sistema propuesto ha demostrado ser una solución prometedora para la navegación autónoma de drones en entornos dinámicos. Sin embargo, existen diversas áreas de mejora y posibles extensiones que podrían ampliar las capacidades y aplicaciones de este proyecto. A continuación, se enumeran algunos de los trabajos futuros más relevantes:

A. Optimización del Sistema en Tiempo Real

Aunque el sistema actual opera en tiempo real bajo ciertas condiciones, es crucial optimizar aún más los procesos de detección y filtrado. Esto incluye:

- Implementar versiones más ligeras de los modelos YOLO y MiDaS que mantengan un alto desempeño, reduciendo la latencia en hardware embebido.
- Explorar frameworks como TensorRT para la optimización de inferencias en dispositivos con recursos limitados.
- Integrar técnicas de compresión de modelos, como podado y cuantización, para minimizar el uso de memoria y aumentar la velocidad de ejecución.

B. Implementación de Visión Estéreo con Cámaras Dual

Una dirección prometedora para el desarrollo futuro del sistema es la integración de cámaras estéreo para mejorar significativamente la estimación de profundidad. A diferencia de la estimación monocular basada en redes neuronales como MiDaS, el uso de un sistema de visión estéreo permite calcular la profundidad de manera más precisa y directa a partir de la disparidad entre las imágenes capturadas por dos cámaras ligeramente separadas. Este enfoque tiene varias ventajas:

- Mejora en la Precisión: La visión estéreo genera mapas de profundidad más exactos al calcular la disparidad directamente entre imágenes capturadas por dos cámaras.
- Menor Dependencia de Modelos Preentrenados: Reduce la necesidad de redes neuronales para estimar profundidad, adaptándose mejor a diferentes entornos.
- Capacidades en Tiempo Real: Los sistemas estéreo pueden calcular mapas de profundidad rápidamente, ideal para la evasión de obstáculos en tiempo real.
- Robustez en Escenarios Complejos: Incrementa el rendimiento en entornos dinámicos y urbanos donde las estimaciones monoculares pueden fallar.

C. Mejoras en la Planificación de Trayectorias

El sistema actual se enfoca en la detección y filtrado de objetos, pero la integración con algoritmos avanzados de planificación de trayectorias podría llevar el proyecto a un nuevo nivel. Algunas propuestas incluyen:

- Implementar algoritmos como RRT* o A* tridimensionales que consideren las restricciones dinámicas del dron.

- Incorporar predicciones de movimiento de obstáculos en tiempo real, utilizando redes neuronales recurrentes o modelos probabilísticos.
- Diseñar trayectorias más eficientes energéticamente, minimizando el consumo de batería durante la operación.

D. Ampliación del Dataset y Aumentación de Datos

Para mejorar la robustez y la capacidad generalizadora del sistema, se podrían ampliar los datos utilizados en el entrenamiento:

- Capturar y etiquetar imágenes en condiciones diversas, como entornos rurales, áreas boscosas y zonas con alta densidad de objetos.
- Generar datos sintéticos mediante simuladores como AirSim o CARLA, incluyendo escenarios con obstáculos dinámicos y eventos inesperados.
- Aplicar aumentación avanzada, como cambios de dominio o técnicas de estilo GAN, para simular condiciones extremas de iluminación y clima.

E. Validación en Entornos Reales

Aunque el sistema ha sido evaluado en simulaciones, la validación en escenarios reales es fundamental para garantizar su viabilidad en aplicaciones prácticas:

- Probar el sistema en drones comerciales equipados con cámaras RGB y hardware compatible, como NVIDIA Jetson Nano o Raspberry Pi.
- Realizar pruebas en entornos urbanos con tráfico peatonal y vehicular para evaluar la detección en situaciones complejas.
- Evaluar la capacidad del sistema para adaptarse a condiciones meteorológicas adversas, como viento, lluvia o poca visibilidad.

F. Aplicaciones Futuras

El sistema desarrollado podría adaptarse a otras áreas más allá de la navegación autónoma:

- Monitoreo ambiental: Usar drones para detectar obstáculos naturales, como ramas caídas o pendientes pronunciadas, en misiones de rescate.
- Inspección industrial: Aplicar el sistema en inspecciones de infraestructuras, como puentes o torres eléctricas, donde la precisión en la detección de obstáculos es crítica.
- Entregas autónomas: Optimizar rutas para drones de transporte, garantizando la seguridad en entornos con obstáculos dinámicos.

G. Incorporación de Inteligencia Artificial Adaptativa

Para mejorar la capacidad del sistema en situaciones cambiantes, se podrían explorar modelos de aprendizaje adaptativo:

- Entrenar modelos que ajusten sus parámetros en tiempo real según el entorno detectado.
- Implementar aprendizaje por refuerzo para que el dron mejore su desempeño a medida que opera en diferentes escenarios.

V. CONSIDERACIONES ÉTICAS

El sistema propuesto tiene un gran potencial para mejorar la navegación autónoma y la seguridad de los drones en entornos dinámicos. Sin embargo, al operar de manera autónoma en entornos complejos, pueden conllevar desafíos relacionados con la privacidad, la seguridad y el uso responsable de la tecnología, lo que exige un marco ético claro y acciones preventivas en su diseño y despliegue [13]–[15].

Uno de los aspectos más críticos es la privacidad, ya que los drones equipados con cámaras y sensores pueden recopilar datos sensibles o personales, como imágenes de personas o propiedades privadas [14]. Para mitigar este riesgo, es imprescindible que el sistema sea diseñado de forma que cumpla con normativas de protección de datos, como las que se mencionan al finalizar la sección, y que limite el almacenamiento o transmisión de información que pueda ser considerada invasiva [13], [15]. Además, se deben establecer protocolos claros sobre cómo y quién tiene acceso a estos datos, asegurando que su uso esté justificado y sea transparente.

La seguridad también constituye un eje central en las consideraciones éticas, ya que un fallo en el sistema podría poner en peligro tanto a las personas como a las propiedades en el área de operación del dron [14]. La fiabilidad del modelo debe ser prioritaria, asegurando que el sistema de detección y evasión de obstáculos funcione adecuadamente en tiempo real, incluso en condiciones adversas. Por lo tanto, resulta esencial llevar a cabo pruebas exhaustivas en entornos simulados y reales antes de implementar el modelo en aplicaciones prácticas, además de incluir redundancias en el diseño para minimizar el impacto de posibles errores.

Russian drone appears to injure civilians

Drone footage of Kherson attack in early October 2024

1. Drone flies over Kherson



2. Drone approaches two people



3. Drone drops explosive



4. Civilians appear injured



Fig. 26. Uso de Drones en contextos bélicos [16]

Otro elemento clave es el uso responsable de la tecnología. Los sistemas autónomos pueden ser utilizados en una variedad de contextos, pero es fundamental establecer limitaciones claras para evitar aplicaciones que puedan ser consideradas poco éticas o ilegales, como la vigilancia no autorizada,

el uso militar ofensivo o cualquier actividad que viole los derechos humanos [16]–[18]. Los desarrolladores y operadores deben comprometerse con lineamientos éticos que prioricen el bienestar social y el respeto por las leyes locales e internacionales.

Es necesario remarcar la importancia que el sistema cumpla con regulaciones internacionales, como las normas de la FAA en Estados Unidos y las directrices de la EASA en Europa, que establecen límites operativos, requisitos de certificación y evaluaciones de riesgos [19], [20]. Asimismo, la protección de datos personales debe alinearse con normativas como el GDPR en Europa, que regula la recolección, almacenamiento y uso de información sensible. Estas regulaciones, junto con la adopción de lineamientos éticos globales, como los principios de IA de la UNESCO, aseguran que la tecnología se despliegue de forma justa y responsable, evitando discriminación y promoviendo la transparencia algorítmica [21], [22].

Finalmente, es importante considerar el impacto social de esta tecnología, especialmente en términos de equidad y accesibilidad. Si bien los avances en autonomía prometen revolucionar diversas industrias, existe el riesgo de que su adopción quede limitada a organizaciones con mayores recursos, ampliando la brecha tecnológica [15]. Por ello, resulta relevante explorar formas de democratizar estas soluciones, promoviendo su accesibilidad y desarrollando estrategias que permitan su implementación responsable en entornos con menos recursos.

VI. CONCLUSIÓN

Los resultados obtenidos destacan la efectividad del sistema desarrollado para la detección y filtrado de objetos según altitud y profundidad, mejorando la capacidad de navegación autónoma en drones. En las imágenes procesadas, los bounding boxes verdes representan los objetos aceptados tras el filtrado, mientras que los rojos indican los objetos descartados. Este enfoque permite priorizar elementos relevantes para la toma de decisiones, como peatones, vehículos y obstáculos estáticos, reduciendo falsas detecciones y minimizando riesgos en entornos complejos.

La implementación de mapas de profundidad estimados con MiDaS, junto con la integración de datos de altitud de AU-AIR, ha permitido calcular distancias promedio precisas en áreas de interés. Este sistema no solo mejora la precisión, sino que también contribuye a la eficiencia computacional al procesar solo datos esenciales. Asimismo, los resultados subrayan la importancia de un preprocesamiento cuidadoso y una correcta adaptación de los datasets, como VisDrone y AU-AIR, al formato YOLO para garantizar la compatibilidad y el desempeño óptimo del modelo.

A pesar de los avances logrados, los resultados de entrenamiento sobre la detección presentan limitaciones significativas, atribuibles principalmente a la baja cantidad de épocas entrenadas. Este aspecto se mantuvo limitado debido a restricciones computacionales, ya que el sistema fue implementado utilizando la versión gratuita de GPU en Google Colab. Estas limitaciones afectaron la capacidad del modelo

para generalizar y alcanzar un desempeño más robusto en términos de precisión y *recall*.

Por último, las consideraciones éticas del sistema propuesto destacan la importancia de garantizar la privacidad, la seguridad y el uso responsable de la tecnología. Cumplir con normativas internacionales como el GDPR y establecer protocolos claros para el manejo de datos asegura la protección de la información personal y fomenta la transparencia. Además, priorizar la fiabilidad del sistema mediante pruebas exhaustivas y redundancias minimiza riesgos en entornos reales, mientras que limitar su uso a aplicaciones éticas evita prácticas como la vigilancia no autorizada o el uso ofensivo en conflictos. Finalmente, democratizar el acceso a estas tecnologías es esencial para reducir desigualdades y promover un impacto equitativo en la sociedad, alineándose con lineamientos globales que fomenten la equidad y la transparencia algorítmica.

REFERENCES

- [1] J. N. Yasin, S. A. S. Mohamed, M.-H. Haghbayan, J. Heikkonen, H. Tenhunen, and J. Plosila, “Unmanned aerial vehicles (uavs): Collision avoidance systems and approaches,” *IEEE Access*, vol. 8, pp. 105139–105155, 2020.
- [2] A. Anwar and A. Raychowdhury, “Autonomous navigation via deep reinforcement learning for resource constraint edge nodes using transfer learning,” *IEEE Access*, vol. 8, pp. 26549–26560, 2020.
- [3] S. Sun, G. Cioffi, C. de Visser, and D. Scaramuzza, “Autonomous quadrotor flight despite rotor failure with onboard vision sensors: Frames vs. events,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 580–587, 2021.
- [4] X. Yang, J. Chen, Y. Dang, H. Luo, Y. Tang, C. Liao, P. Chen, and K.-T. Cheng, “Fast depth prediction and obstacle avoidance on a monocular drone using probabilistic convolutional neural network,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 1, pp. 156–167, 2021.
- [5] Ultralytics, “Yolo11 performance metrics,” 2024.
- [6] Y. Lee, J. An, and I. Joe, “Deep-learning-based object filtering according to altitude for improvement of obstacle recognition during autonomous flight,” *Remote Sensing*, vol. 14, no. 6, p. 1378, 2022.
- [7] I. Bozcan and E. Kayacan, “Au-air: A multi-modal unmanned aerial vehicle dataset for low altitude traffic surveillance,” 2020.
- [8] S. Mahna, “Cityscapes - depth and segmentation,” 2020.
- [9] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling, “Detection and tracking meet drones challenge,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “YOLO: You Only Look Once,” pp. 779–788, 2016.
- [11] R. Ranftl, A. Bochkovskiy, *et al.*, “MiDaS: Robust Monocular Depth Estimation.” <https://github.com/isl-org/MiDaS>, 2020. Accessed: 2024-11-26.
- [12] D. O. Detection, “Drone object detection dataset.” <https://universe.roboflow.com/drone-obstacle-detection/drone-object-detection-yhpn6> , may 2024. visited on 2024-11-26.
- [13] R. Luppicini and A. So, “A technoeethical review of commercial drone use in the context of governance, ethics, and privacy,” *Technology in Society*, vol. 46, pp. 109–119, 2016.
- [14] R. L. Finn and D. Wright, “Privacy, data protection and ethics for civil drone practice: A survey of industry, regulators and civil society organisations,” *Computer Law Security Review*, vol. 32, no. 4, pp. 577–586, 2016.
- [15] R. Iphofen and M. Kritikos, “Regulating artificial intelligence and robotics: ethics by design in a digital society,” *Contemporary Social Science*, vol. 16, no. 2, pp. 170–184, 2021.
- [16] Y. Limaye, “Russian drones hunt civilians, evidence suggests,” Oct 2024. Accessed: Nov. 24, 2024.
- [17] P. Boucher, “Domesticating the drone: The demilitarisation of unmanned aircraft for civil markets,” *Science and Engineering Ethics*, vol. 21, no. 5, pp. 1393–1412, 2015.
- [18] M. Coeckelbergh, “Drones, information technology, and distance: Mapping the moral epistemology of remote fighting,” *Ethics and Information Technology*, vol. 15, no. 2, pp. 87–98, 2013.
- [19] F. A. A. (FAA), “Summary of small unmanned aircraft rule (part 107),” 2024.
- [20] E. U. A. S. A. (EASA), “Regulations for unmanned aircraft systems (uas),” 2024.
- [21] E. Parliament and C. of the European Union, “General data protection regulation (gdpr),” 2016.
- [22] S. United Nations Educational and C. O. (UNESCO), “Recommendation on the ethics of artificial intelligence,” 2021.