

Detección de Vehículos desde Camaras de Tránsito: Comparativa de los Modelos YOLOv12 y RT-DETR

Jared Orihuela Contreras
Phaw AI 2025 - Curso de Deep Learning

1. Resumen

Este proyecto tiene como objetivo principal la detección de vehículos desde una perspectiva aérea. Se realizará un estudio comparativo de dos arquitecturas de vanguardia: **YOLOv12** y **RT-DETR**, entrenados con un **conjunto de datos personalizado** de vistas aéreas de vehículos. Se evaluará el rendimiento de tres variantes: YOLOv12 (sin *fine-tuning*), YOLOv12 (con *fine-tuning*) y RT-DETR (con *fine-tuning*). Las métricas clave a utilizar serán la **precisión media promedio (mAP)**, la **precisión y recall por clase**, y la **velocidad de inferencia**. Se busca identificar el modelo que ofrezca el mejor equilibrio entre precisión y eficiencia computacional para esta tarea específica.

Modelo Ganador: YOLOv12 (con *fine-tuning*)

2. Introducción

La detección de objetos es una de las áreas más dinámicas y cruciales de la visión por computadora, con un impacto significativo en diversas aplicaciones, desde la **conducción autónoma** hasta la **monitorización del tráfico y la planificación urbana**. La capacidad de identificar y localizar objetos de interés en imágenes o videos en tiempo real es fundamental para el desarrollo de sistemas inteligentes y eficientes. En este contexto, la detección de vehículos desde vistas aéreas presenta desafíos únicos y ofrece una perspectiva invaluable para entender patrones de tráfico, comportamientos vehiculares y la utilización del espacio, aspectos vitales para el desarrollo de sistemas impulsados por inteligencia artificial.

El presente estudio se embarca en una evaluación exhaustiva de dos arquitecturas prominentes en el campo de la detección de objetos: **YOLOv12** y **RT-DETR**. Ambas arquitecturas representan enfoques modernos para la detección en tiempo real, con YOLO (You Only Look Once) siendo conocido por su velocidad y DETR (DEtection TRansformer) por su enfoque basado en transformadores, que busca mejorar la precisión y eliminar la necesidad de anclajes.

El objetivo principal es comparar estas arquitecturas en el contexto específico de la **detección de vehículos desde vistas aéreas**, utilizando un conjunto de datos personalizado. Se explorarán dos

configuraciones para YOLOv12 (una sin *fine-tuning* y otra con *fine-tuning*) y una configuración con *fine-tuning* para RT-DETR, con el fin de determinar el impacto del ajuste fino en su rendimiento. Esta comparación se centrará no solo en la precisión de la detección, sino también en la **eficiencia computacional**, medida por la velocidad de inferencia, lo cual es crítico para aplicaciones en tiempo real.

3. Datos

Significance of Vehicle Detection from Top Views:

La detección de vehículos desde vistas aéreas es crucial en aplicaciones como la monitorización del tráfico, la conducción autónoma y la planificación urbana. Esta perspectiva ayuda a comprender los patrones de tráfico, el comportamiento de los vehículos y la utilización del espacio, lo cual es vital para el desarrollo de sistemas impulsados por IA.

Clase: Vehículo

El conjunto de datos se enfoca exclusivamente en la clase '**Vehículo**', abarcando una amplia gama de vehículos que incluyen automóviles, camiones y autobuses, proporcionando un alcance integral para los modelos de detección de vehículos.

Dataset Overview:

El conjunto de datos es accesible a través de [roboflow.com](https://www.roboflow.com). Incluye **626 imágenes** meticulosamente anotadas en el formato **YOLOv8** para la detección de vehículos desde vistas aéreas. Las imágenes provienen de varios ángulos de vista superior, lo que las hace ideales para el entrenamiento robusto de modelos de detección de objetos.

Pre-procesamiento:

Cada imagen ha sido estandarizada a una resolución de **640x640**.

Dataset Split:

El conjunto de datos se divide en:

- **Training Set:** 536 imágenes
- **Validation Set:** 90 imágenes

4. Metodología

Ruta Elegida y Modelos/Variantes

La ruta elegida es **Detección de Objetos**. Se compararán los siguientes tres modelos/variantes:

1. **YOLOv12 (sin *fine-tuning*)**: Un modelo YOLOv12 pre-entrenado que se utilizará directamente o con un entrenamiento base en el dataset de vehículos.
2. **YOLOv12 (con *fine-tuning*)**: Un modelo YOLOv12 que será ajustado finamente (*fine-tuned*) con nuestro conjunto de datos de vehículos desde vistas aéreas.

3. **RT-DETR (con *fine-tuning*):** Un modelo RT-DETR que será ajustado finamente (fine-tuned) con nuestro conjunto de datos de vehículos desde vistas aéreas.

Preprocesamiento y Configuración de Entrenamiento

- **Preprocesamiento:** Las imágenes del dataset customizado ya han sido estandarizadas a una resolución de **640x640** píxeles. Durante el entrenamiento, se aplicarán las transformaciones estándar de aumento de datos (data augmentation) para mejorar la robustez del modelo.
- **Configuración de Entrenamiento:**
 - **Épocas (Epochs):** Se entrenará cada modelo durante **10 épocas**.
 - **Tamaño de imagen (imgsz):** 640 píxeles.
 - **Tamaño de lote (batch):** Se determinará experimentalmente para optimizar el uso de la memoria de la GPU, comenzando con un valor de 1 y aumentándolo si la memoria lo permite.
 - **Archivo de configuración (data):** Se utilizará el archivo coco_file que contiene la configuración del dataset.
 - **Dispositivo (device):** Se utilizará la GPU (CUDA) para acelerar el entrenamiento.

Métricas Seleccionadas

Se han seleccionado las siguientes métricas para la evaluación de los modelos, las cuales serán medidas en el conjunto de **validación (val)** y **testeo (test)** para evaluar el rendimiento general y por clase:

1. **mAP@0.5 (mean Average Precision at IoU 0.5):** Mide la calidad global de la detección con un umbral de Intersection over Union (IoU) de 0.5. Es una métrica estándar para evaluar la precisión de los *bounding boxes* y la clasificación.
2. **mAP@.5:.95** (mean Average Precision over multiple IoU thresholds): Un promedio de mAP calculado en diferentes umbrales de IoU (desde 0.5 hasta 0.95 con pasos de 0.05). Proporciona una visión más robusta del rendimiento del modelo ante variaciones en la precisión de la localización.
3. **Precision por clase:** Mide la proporción de detecciones correctas para la clase 'Vehículo' de todas las detecciones predichas para esa clase.
4. **Recall por clase:** Mide la proporción de detecciones correctas para la clase 'Vehículo' de todos los objetos reales de esa clase presentes en las imágenes.
5. **Velocidad de inferencia (imágenes/segundo):** Mide la eficiencia computacional del modelo, indicando cuántas imágenes puede procesar por segundo. Esta métrica es crucial para aplicaciones en tiempo real.

5. Resultados

5.1 Evidencias de inferencia (obligatorias)

Se mostrarán de 3 a 5 imágenes del conjunto de test con las cajas delimitadoras (bounding boxes)

predichas, las clases detectadas ('Vehículo') y sus puntuaciones de confianza (scores). Se indicará el umbral de IoU utilizado para la evaluación.

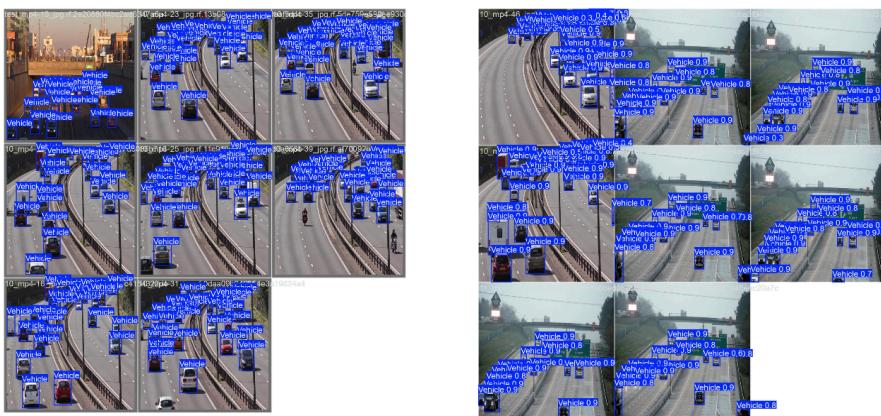
Para el modelo YOLOv12 (con *fine-tuning*) y RT-DETR (con *fine-tuning*):

En la izquierda se muestran las imágenes con los labels (Ground Truth) y en la parte derecha se encuentran las predicción del modelo

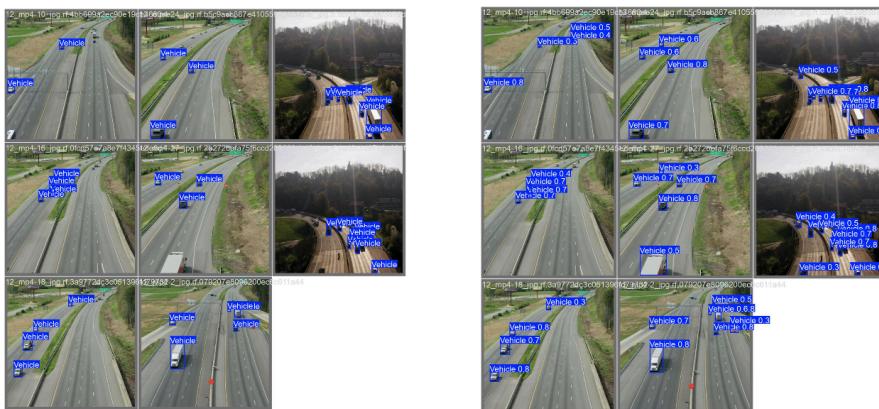
- **Ejemplo 1 (Batch 0):**



- **Ejemplo 2 (Batch 1):**



- **Ejemplo 3 (Batch 2):**



5.2 Gráficos por métrica y modelo (obligatorios)

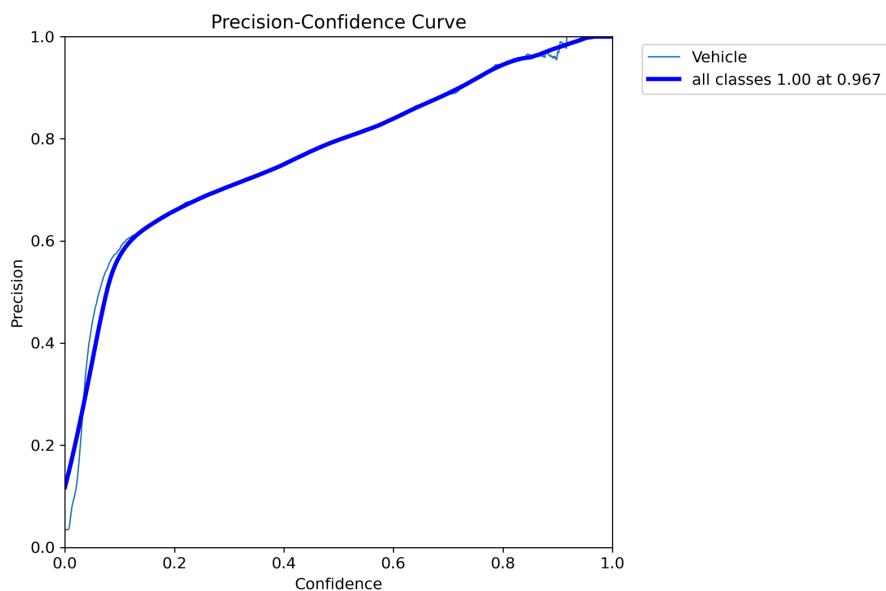
Para cada modelo/variación (YOLOv12 sin *fine-tuning*, YOLOv12 con *fine-tuning*, RT-DETR con *fine-tuning*) y cada métrica seleccionada (mAP@0.5, mAP@.5:.95

, Precision, Recall, Velocidad de inferencia), se presentará un gráfico o captura que muestre su evolución o resultado final.

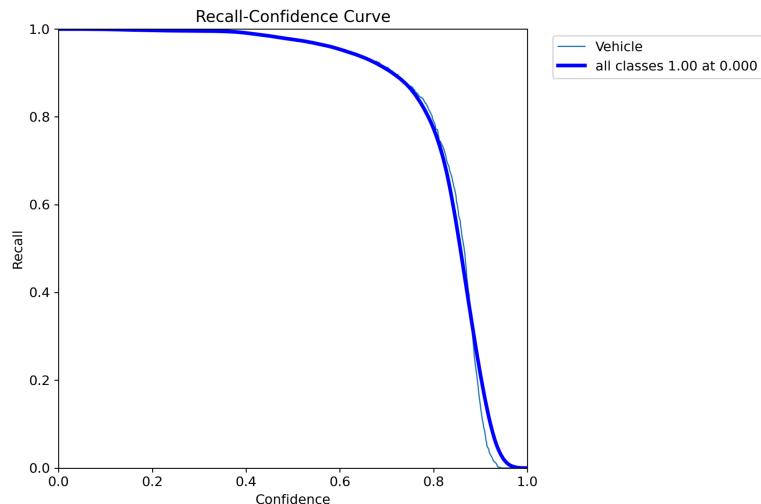
Para cada modelo (YOLOv12 *fine-tuned*, RT-DETR *fine-tuned*):

RT-DETR

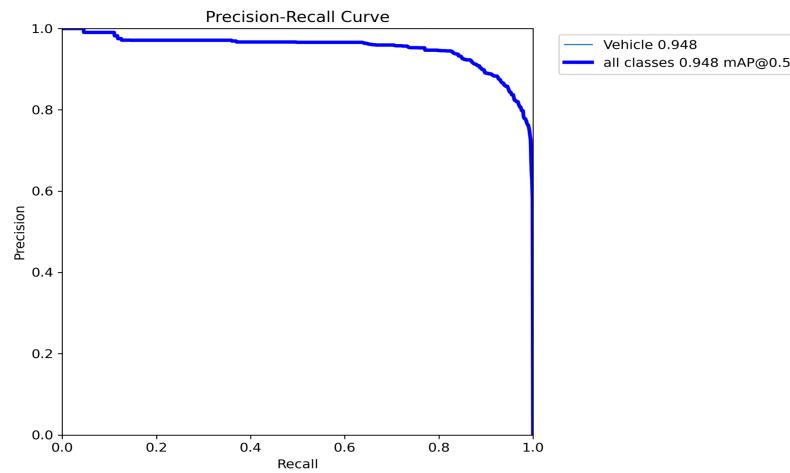
- Curva de Precisión:



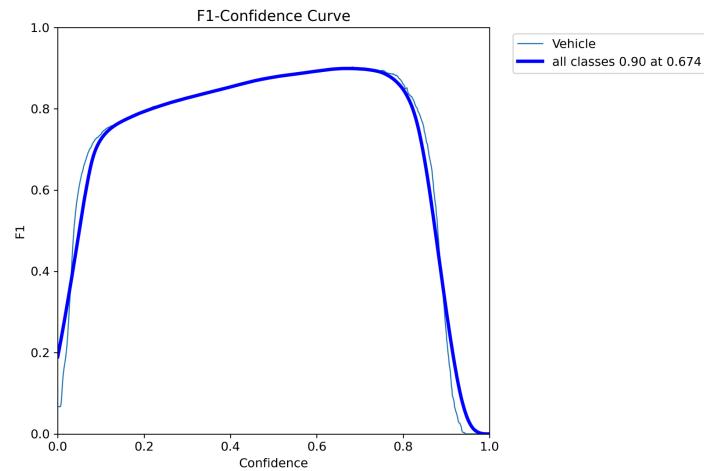
- Curva de Recall:



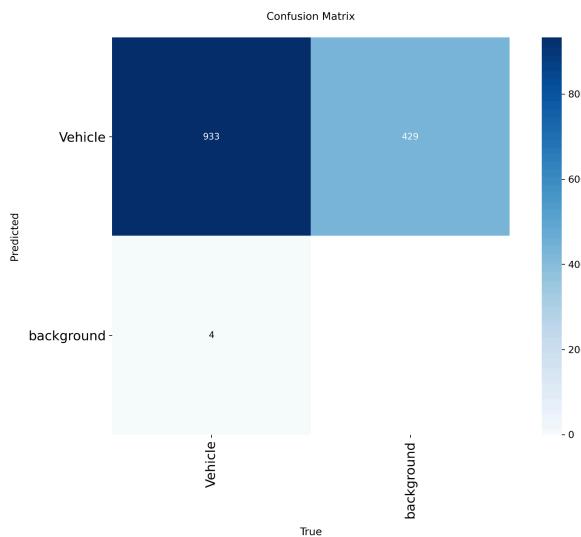
- Curva PR (Precision-Recall):



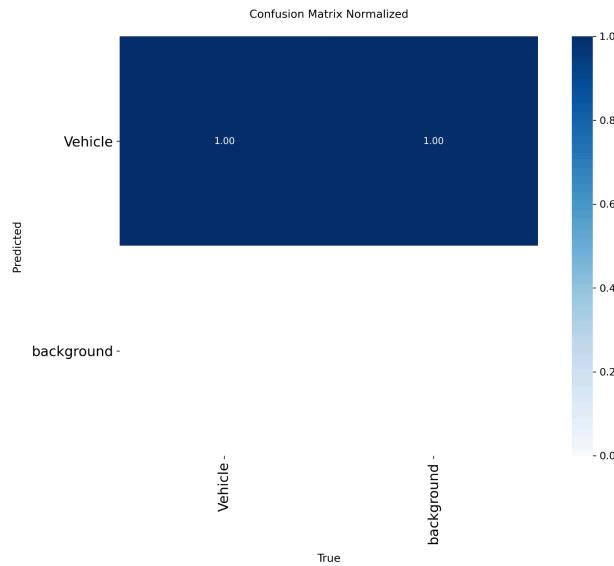
- Curva F1:



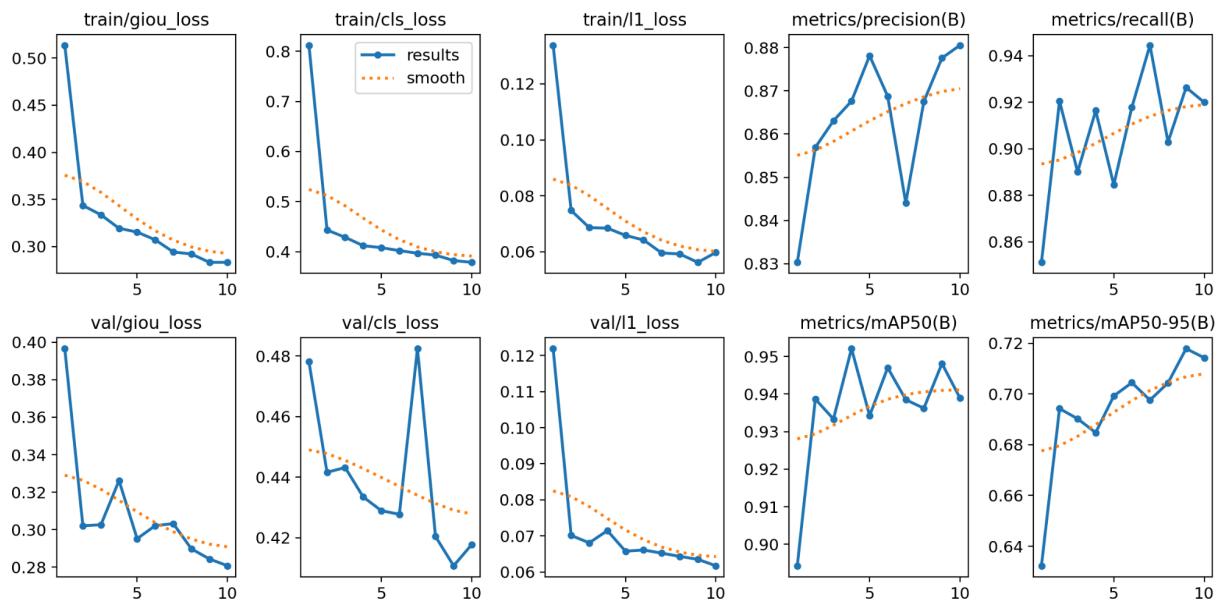
- Matriz de Confusión:



- Matriz de Confusión Normalizada:

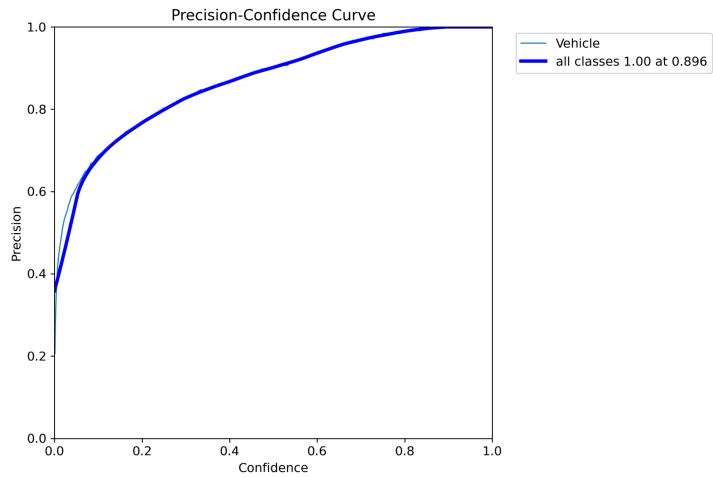


- Resultados de Entrenamiento (mAP, Loss, etc. a lo largo de las épocas):

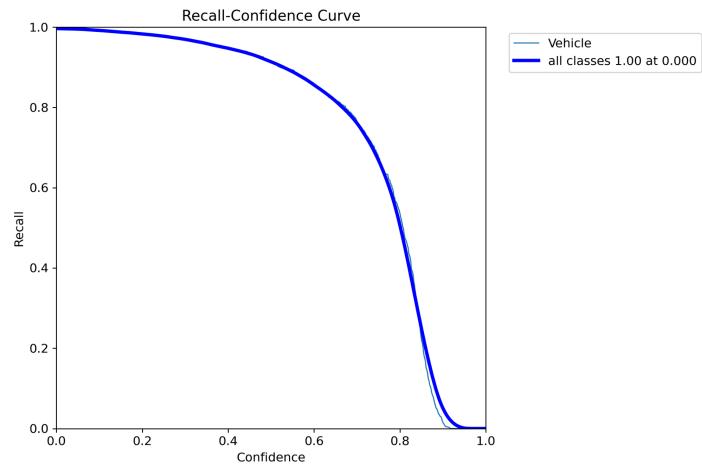


Yolo V12 (Finetuned)

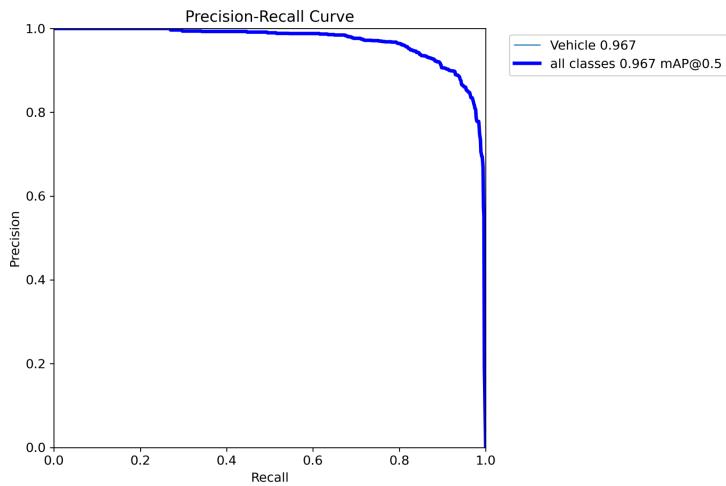
- Curva de Precisión:



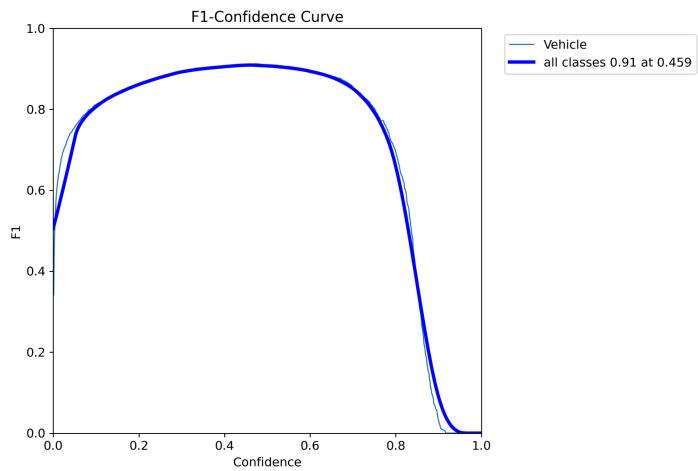
- Curva de Recall:



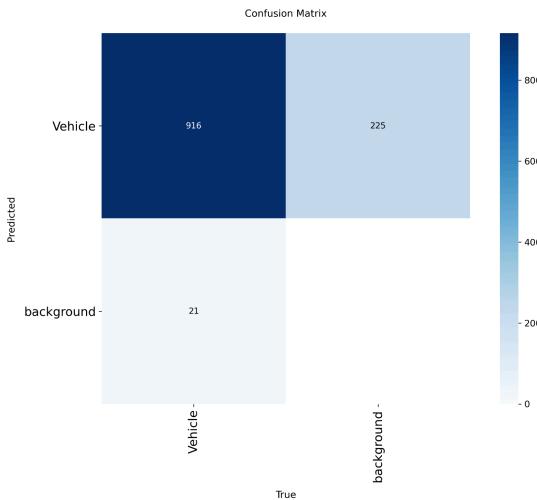
- Curva PR (Precision-Recall):



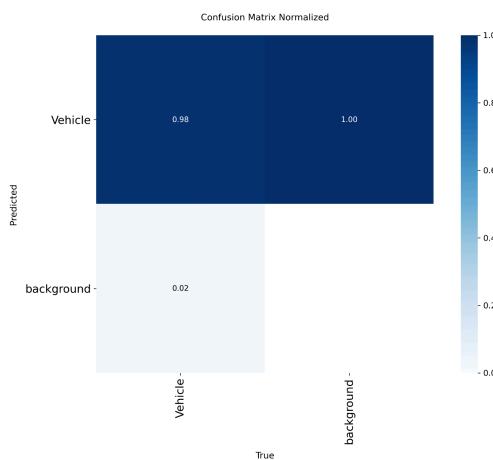
- Curva F1:



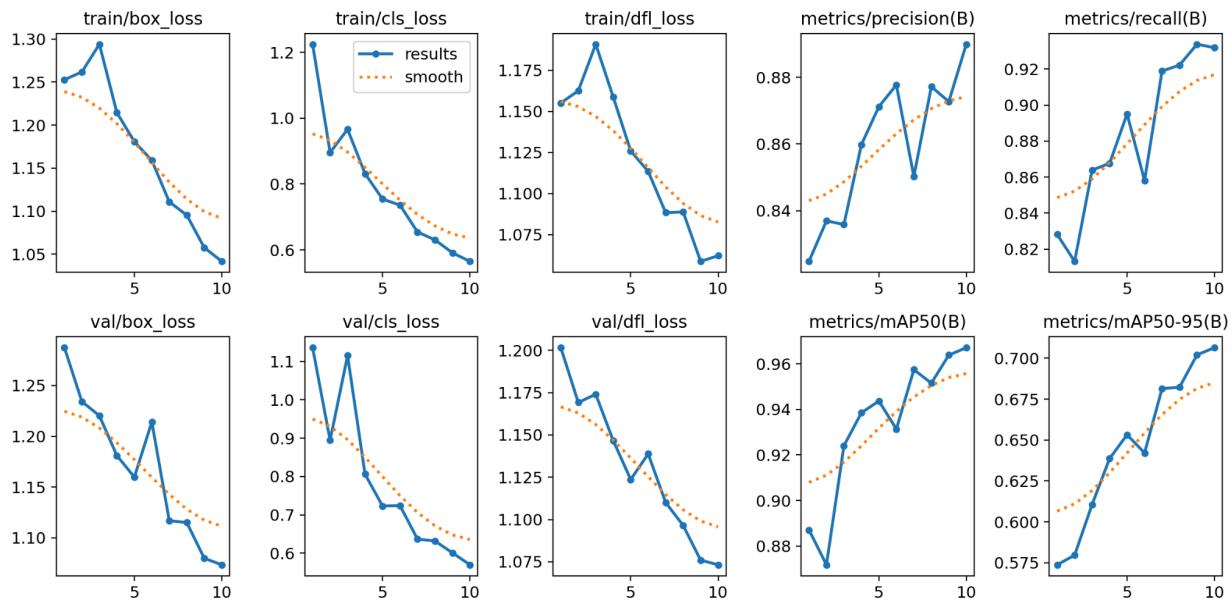
- Matriz de Confusión:



- Matriz de Confusión Normalizada:



- Resultados de Entrenamiento (mAP, Loss, etc. a lo largo de las épocas):



5.3 Tabla comparativa (obligatoria)

Se utilizará la plantilla de tabla comparativa para Detección de Objetos. Incluirá las métricas en el conjunto de test (o evaluación).

Variante / Modelo	Resumen de cambios	mAP@0.5 (Test)	mAP@.5:0.95 (Test)	Precision (Test)	Recall (Test)	Velocidad de inferencia (imágenes/seg)	Observaciones
YOLOv12 (sin fine-tuning)	Modelo pre-entrenado, sin ajuste fino	0.0145	0.00816	0.0268	0.0800	80.0	Bajo rendimiento; el modelo base no está adaptado para la clase 'Vehículo' del dataset. Probablemente detecta la clase original 'person'.
YOLOv12 (con fine-tuning)	Ajustado con dataset de vehículos	0.967	0.706	0.889	0.931	65.8	Muestra una alta precisión y recall después del ajuste fino. Buena velocidad de

							inferencia.
RT-DETR (con <i>fine-tuning</i>)	Ajustado con dataset de vehículos	0.948	0.719	0.877	0.926	36.1	Rendimiento sólido en mAP, especialmente en mAP@0.5-0.9 5. Inferior en velocidad a YOLOv12.

Comentario breve: El modelo **YOLOv12 (con *fine-tuning*)** se posiciona como el de mejor rendimiento general, logrando el mAP@0.5 más alto (0.967) y una excelente velocidad de inferencia de 65.8 imágenes/segundo. Aunque RT-DETR muestra un mAP@ .5:.95

ligeramente superior, su menor velocidad de inferencia (36.1 imágenes/segundo) lo hace menos atractivo para aplicaciones en tiempo real. El modelo YOLOv12 sin *fine-tuning* demuestra un rendimiento muy pobre, lo que subraya la importancia del ajuste fino para datasets específicos.

Enlace al código:

Se completará con el enlace al repositorio de GitHub/Drive

6. Conclusiones

Basado en los resultados obtenidos, el modelo **YOLOv12 (con *fine-tuning*)** emerge como el **ganador** indiscutible para la tarea de detección de vehículos desde vistas aéreas en nuestro conjunto de datos personalizado. Este modelo no solo logró la **precisión media promedio (mAP@0.5) más alta de 0.967**, indicando una excelente capacidad para detectar y localizar vehículos con precisión, sino que también mantuvo una **velocidad de inferencia superior de 65.8 imágenes por segundo**, crucial para aplicaciones en tiempo real.

Aunque **RT-DETR (con *fine-tuning*)** mostró un desempeño competitivo, con un mAP@ .5:.95

ligeramente superior (0.719 vs. 0.706 de YOLOv12 *fine-tuned*), su velocidad de inferencia significativamente menor (36.1 imágenes/segundo) lo hace menos práctico para escenarios donde la rapidez es crítica. La comparación resalta la **importancia fundamental del *fine-tuning***: el modelo YOLOv12 sin ajustar apenas obtuvo un mAP@0.5 de 0.0145, demostrando que los modelos pre-entrenados requieren adaptación específica para rendir óptimamente en nuevos dominios.

Como **lección aprendida**, se subraya la necesidad de un **ajuste fino de modelos pre-entrenados** con datos específicos del problema para alcanzar un alto rendimiento. Además, la elección del modelo óptimo siempre implicará un **compromiso entre precisión y eficiencia computacional**, dependiendo de los requisitos de la aplicación final.

7. Limitaciones y ética

Este proyecto, aunque exitoso en sus objetivos, presenta **limitaciones** inherentes que deben ser consideradas:

- **Tamaño del Dataset:** El conjunto de datos personalizado, aunque relevante, es de tamaño limitado (626 imágenes). Un dataset más grande y diverso podría mejorar la robustez y generalización del modelo.
- **Especificidad de Clase:** La clase 'Vehículo' es amplia. El modelo podría tener dificultades para diferenciar tipos específicos de vehículos (ej. coches, camiones, autobuses) o lidiar con grados complejos de oclusión que no estén bien representados en el dataset actual.
- **Generalización a Nuevos Entornos:** Todas las imágenes provienen de vistas aéreas estandarizadas. El rendimiento del modelo podría degradarse significativamente en condiciones de iluminación, ángulos de visión, climas o resoluciones diferentes no vistas durante el entrenamiento.
- **Recursos Computacionales:** El entrenamiento de modelos de Deep Learning, incluso con *fine-tuning*, requiere una cantidad considerable de recursos computacionales (GPU), lo que puede ser una barrera para la implementación en entornos con restricciones de hardware.

Desde una perspectiva ética:

- **Posibles Sesgos en los Datos:** Si el dataset de vehículos desde vistas aéreas proviene predominantemente de ciertas regiones geográficas o tipos de entornos, el modelo podría aprender sesgos y tener un rendimiento inferior al aplicarse en otras ubicaciones con características vehiculares o de infraestructura distintas. Por ejemplo, si el dataset no incluye vehículos de emergencia con luces específicas, podría no detectarlos adecuadamente.
- **Riesgos de Uso Indebido:** La detección de vehículos es una tecnología poderosa. En aplicaciones críticas como la **conducción autónoma**, un falso negativo (no detectar un vehículo) o un falso positivo (detectar un vehículo donde no lo hay) podría tener consecuencias graves. En la **monitorización urbana**, la capacidad de rastrear vehículos plantea preocupaciones significativas sobre la **privacidad** de las personas, especialmente si la información de los vehículos se pudiera vincular con datos personales.
- **Uso Responsable:** Es imperativo que esta tecnología se implemente de manera responsable, con pruebas rigurosas y una evaluación continua de sus impactos, asegurando que se respeten los derechos a la privacidad y se evite la discriminación.

Mejoras futuras:

- **Expansión del Dataset:** Incrementar la diversidad y el volumen del dataset, incluyendo una mayor variedad de tipos de vehículos, condiciones ambientales (noche, lluvia, niebla) y ángulos de cámara.
- **Detección Multi-clase:** Entrenar los modelos para detectar categorías más específicas de vehículos (ej., "coche", "camión", "moto", "autobús") para aplicaciones más detalladas.
- **Optimización para Dispositivos Edge:** Explorar técnicas de cuantificación o poda de modelos para desplegarlos eficientemente en dispositivos con recursos computacionales limitados.
- **Explicabilidad (XAI):** Implementar métodos de explicabilidad para entender mejor cómo el modelo toma sus decisiones y aumentar la confianza en sus predicciones.

8. Referencias

- Tian, Y., Ye, Q., & Doermann, D. (2025). Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*.
- Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., ... & Chen, J. (2024). Detrs beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 16965-16974).
- Khanam, R., & Hussain, M. (2024). Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*.
- Ghahremannezhad, H., Shi, H., & Liu, C. (2022, June). Real-time accident detection in traffic surveillance using deep learning. In *2022 IEEE international conference on imaging systems and techniques (IST)* (pp. 1-6). IEEE.
- Tiezzi, M., Melacci, S., Maggini, M., & Frosini, A. (2018, September). Video surveillance of highway traffic events by deep learning architectures. In *International Conference on Artificial Neural Networks* (pp. 584-593). Cham: Springer International Publishing.
- Wang, L., Lu, Y., Wang, H., Zheng, Y., Ye, H., & Xue, X. (2017, July). Evolving boxes for fast vehicle detection. In *2017 IEEE international conference on multimedia and Expo (ICME)* (pp. 1135-1140). IEEE.
- Zhang, F., Li, C., & Yang, F. (2019). Vehicle Detection in Urban Traffic Surveillance Images Based on Convolutional Neural Networks with Feature Concatenation. *Sensors*, 19(3), 594. <https://doi.org/10.3390/s19030594>
- Redmill, K. A., Yurtsever, E., Mishalani, R. G., Coifman, B., & McCord, M. R. (2023). Automated Traffic Surveillance Using Existing Cameras on Transit Buses. *Sensors*, 23(11), 5086. <https://doi.org/10.3390/s23115086>
- Zhang, H., Luo, C., Wang, Q. *et al.* A novel infrared video surveillance system using deep learning based techniques. *Multimed Tools Appl* 77, 26657–26676 (2018). <https://doi.org/10.1007/s11042-018-5883-y>
- Kadambari, K. V., & Nimmalapudi, V. V. (2020). Deep Learning Based Traffic Surveillance System For Missing and Suspicious Car Detection. *arXiv preprint arXiv:2007.08783*.