

Sistema de Detección de Placas Vehiculares

Aplicado a monitoreo inteligente de estacionamientos y movilidad urbana

Julian Gutierrez y Felipe Idarraga



El Problema: Desafíos en la Gestión Vehicular Urbana

Congestión y Tiempo Perdido

La búsqueda de aparcamiento y la gestión ineficiente del tráfico generan pérdidas significativas de tiempo y productividad en las ciudades.

Impacto Ambiental

La congestión vehicular contribuye directamente al aumento de emisiones contaminantes y a una menor calidad del aire en entornos urbanos.

Seguridad y Monitoreo

La falta de un sistema de monitoreo eficaz dificulta la seguridad en aparcamientos y la identificación de vehículos en situaciones críticas.

Nuestra solución aborda estos desafíos mediante la lectura automática de placas, facilitando el control de acceso y la supervisión en centros comerciales, universidades y calles.

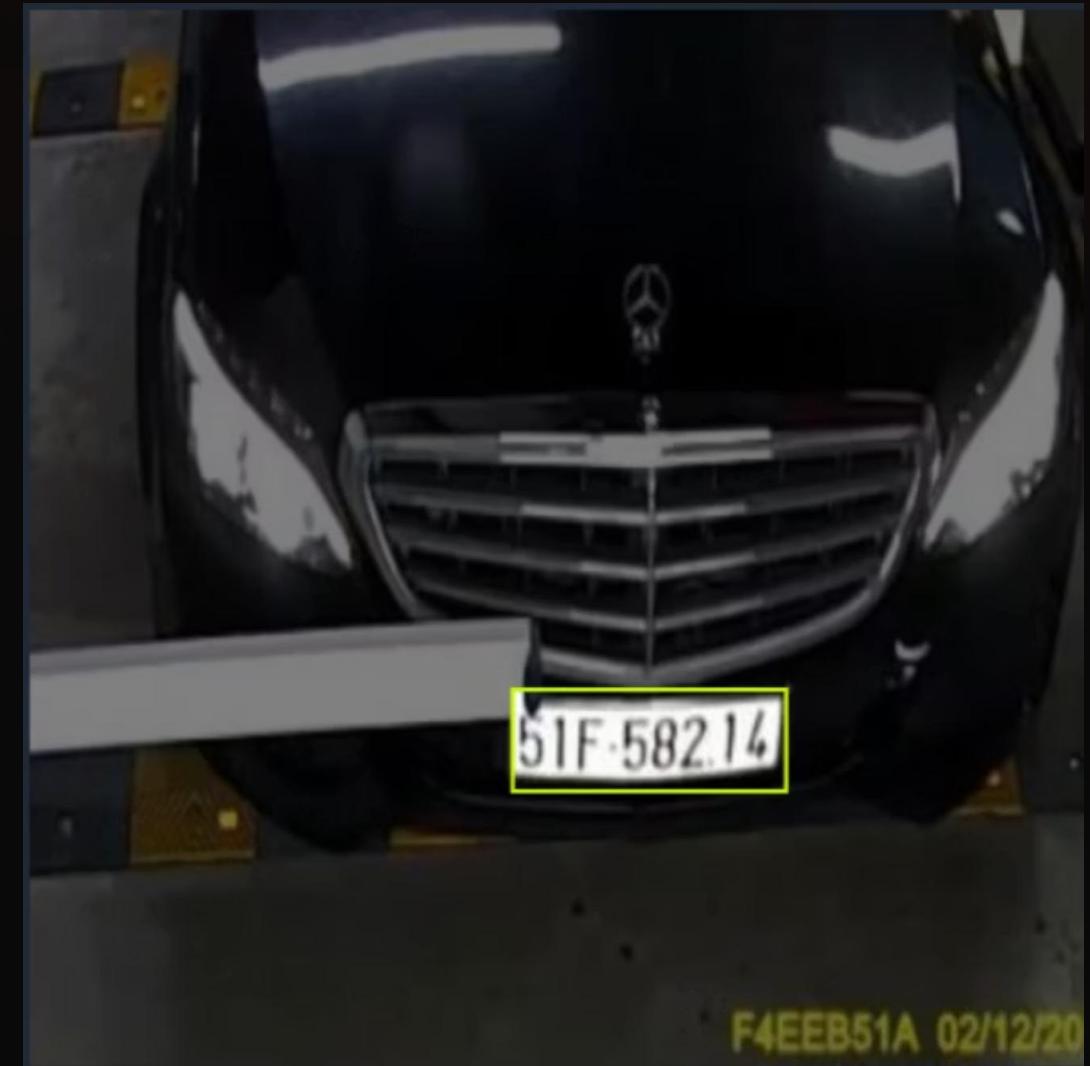


Nuestra Solución: Detección de Placas con Deep Learning

Detección de Objetos con Bounding Boxes

Implementamos una tarea de **detección de objetos** para identificar y localizar placas vehiculares mediante **bounding boxes**. Este enfoque nos permite precisar la ubicación de cada placa en tiempo real.

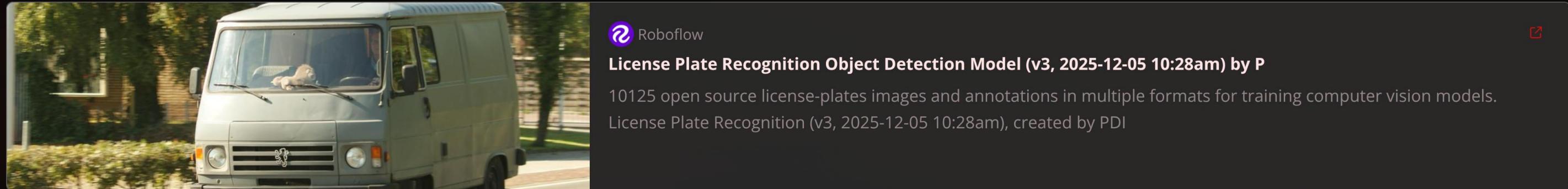
- **Automatización:** Elimina la necesidad de intervención manual en la identificación de vehículos.
- **Precisión:** Alta fiabilidad en el reconocimiento, incluso en condiciones variables.
- **Escalabilidad:** Adaptable a diferentes entornos y volúmenes de tráfico, desde pequeños aparcamientos hasta grandes infraestructuras urbanas.



Este enfoque facilita una gestión vehicular más eficiente y segura, sentando las bases para sistemas de movilidad urbana más inteligentes.

El Corazón del Modelo: Nuestro Dataset

Para entrenar nuestro modelo, utilizamos un dataset robusto de **Roboflow**, fundamental para la precisión del sistema.



Origen

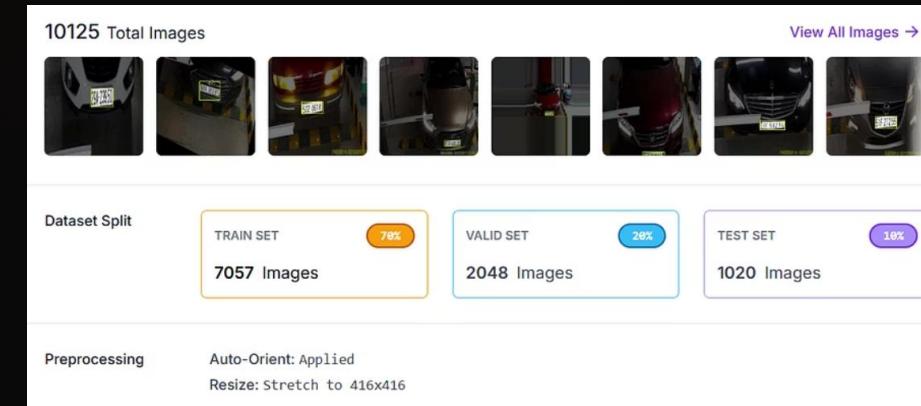
Dataset de Roboflow, específicamente de la categoría **License Plate Recognition**.

Volumen

Consta de **10125 imágenes** cuidadosamente seleccionadas y anotadas, proporcionando una base sólida para el entrenamiento.

Anotaciones Clave

Cada imagen contiene anotaciones detalladas de la clase **LicensePlate**, permitiendo al modelo aprender a identificar las placas con gran exactitud.



Este dataset ha sido fundamental para desarrollar un modelo robusto y preciso.

Arquitectura y Entrenamiento del Modelo

Nuestro modelo se basa en una adaptación optimizada para la detección de placas.

01

Modelo Base

Implementamos un modelo de detección propio en PyTorch, derivado de una versión ligera de YOLOv3 Mini, mejorado para la especificidad de las placas vehiculares.

02

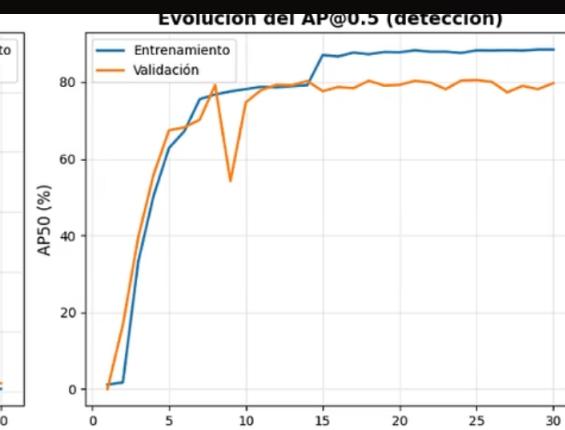
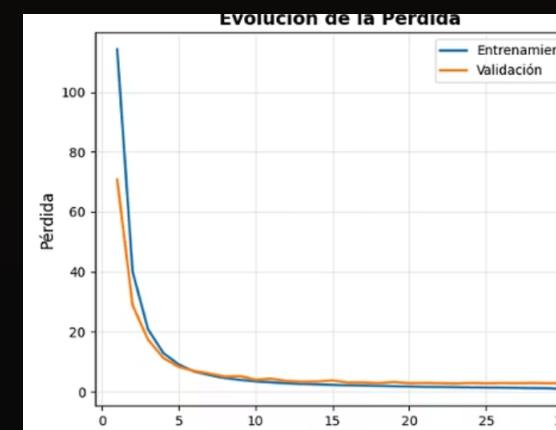
Proceso de Entrenamiento

El entrenamiento se realizó en múltiples **épocas**, ajustando los parámetros para minimizar la **función de pérdida** y optimizar el rendimiento del modelo.

03

Métricas Clave

Monitorizamos métricas como la **pérdida de entrenamiento vs. validación** y el **AP@0.5** para asegurar la estabilidad y precisión del modelo.



DETECTION METRICS			
=====			Average Loss: 2.5890
=====			AP@0.5: 81.09%
PER-CLASS METRICS			
Class	Precision	Recall	F1-Score
LicensePlate	91.90%	89.95%	90.92%
Mean	91.90%	89.95%	90.92%

DETAILED PER-CLASS METRICS			
Class	TP	FP	FN
LicensePlate	976	86	109
Mean			

Optimización y Exportación del Modelo

La eficiencia del modelo es clave para su despliegue en entornos reales.

Conversión a TorchScript / LiteRT

Para maximizar la eficiencia y la compatibilidad, convertimos el modelo entrenado a formatos optimizados como **TorchScript** y **LiteRT**. Estos formatos permiten un despliegue más ágil y ligero.

- **Portabilidad:** Facilita la integración en diversas plataformas y sistemas operativos.
- **Velocidad:** Mejora significativamente los tiempos de inferencia.
- **Despliegue en Dispositivos Pequeños:** Ideal para hardware con recursos limitados, como cámaras de tráfico o dispositivos móviles.

Comparación de Tamaños en Disco

- Modelo original (.pth) y Modelo TorchScript (.pt) igual tamaño

```
 Tamaño best_plate_detector.pth : 38.08 MB  
 Tamaño plate_detector_ts_cpu.pt : 38.27 MB  
  
Benchmarking modelos...  
  
 PyTorch (eager): 299.799 ± 4.276 ms por inferencia (n=50)  
 TorchScript (.pt): 303.697 ± 8.123 ms por inferencia (n=50)  
  
Resumen tiempos:  
→ PyTorch (loaded): 299.799 ms  
→ TorchScript (.pt): 303.697 ms  
Speedup aproximado: 0.99x (si >1, TS es más rápido)
```

Se probó la cuantización dinámica al convertir a TorchScript, pero al ser un modelo ya ligero y con pocas capas lineales, la mejora en tamaño y velocidad fue despreciable, por lo que se mantuvo solo la versión TorchScript sin cuantización.

Inferencia en Python: Ejecución en Tiempo Real

Desarrollamos un script de inferencia en Python para demostrar la capacidad del modelo en acción.

El script permite cargar el modelo optimizado y procesar imágenes o flujos de video para detectar placas en tiempo real.

- **Facilidad de Uso:** Interfaz sencilla para la ejecución de pruebas.
- **Resultados Visuales:** Genera imágenes con las placas detectadas y sus respectivos bounding boxes.
- **Análisis de Rendimiento:** Calcula los **tiempos promedio de inferencia**, vital para evaluar la eficiencia en escenarios reales.

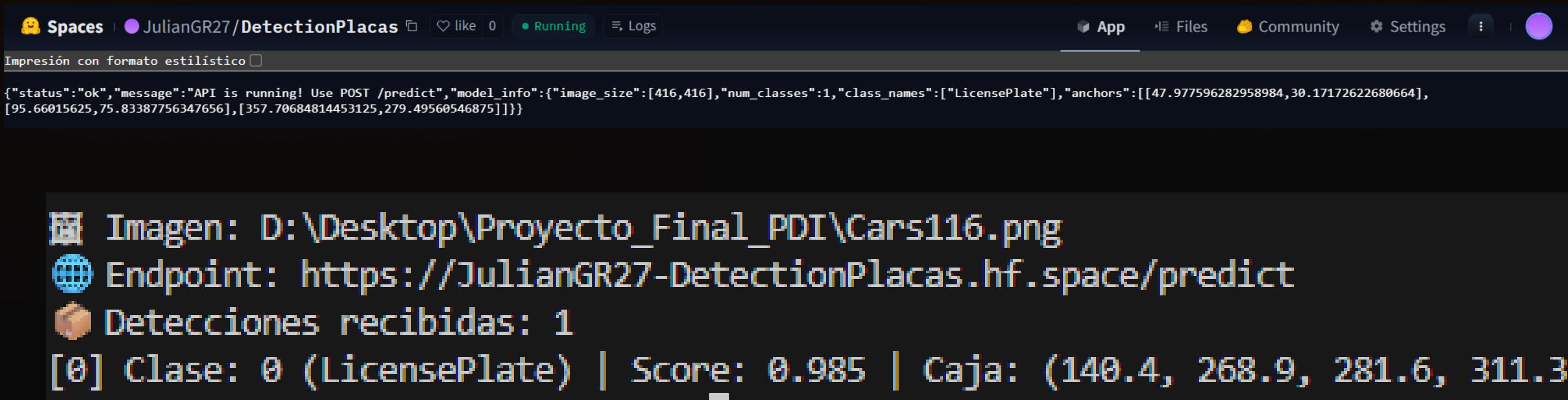
Este componente es crucial para la verificación y ajuste del modelo antes de su implementación definitiva.

Despliegue Interactivo en HuggingFace Spaces

Para facilitar el acceso y la demostración del modelo, lo hemos desplegado en una interfaz interactiva.

Interfaz Intuitiva

Desarrollamos una interfaz de usuario en **HuggingFace Spaces** que permite cargar imágenes y ver la detección de placas en tiempo real, sin necesidad de configuraciones complejas.



The screenshot shows the HuggingFace Spaces interface for a model named "JulianGR27/DetectionPlacas". The top navigation bar includes "Spaces", the model name, "like 0", "Running", and "Logs" buttons, along with "App", "Files", "Community", and "Settings" tabs. A purple profile picture is on the far right. Below the header, a message box displays the API status: {"status": "ok", "message": "API is running! Use POST /predict", "model_info": {"image_size": [416, 416], "num_classes": 1, "class_names": ["LicensePlate"], "anchors": [[47.977596282958984, 30.17172622680664], [95.66015625, 75.83387756347656], [357.70684814453125, 279.49560546875]]}}. The main content area shows a preview of a car image with a detected license plate and the following JSON output:

```
[{"Imagen": "D:\\Desktop\\Proyecto_Final_PDI\\Cars116.png", "Endpoint": "https://JulianGR27-DetectionPlacas.hf.space/predict", "Detecciones recibidas": 1, "[0] Clase": "0 (LicensePlate) | Score: 0.985 | Caja: (140.4, 268.9, 281.6, 311.3)"}
```

¡Prueba nuestro sistema en vivo! HuggingFace Space

Resultados, Limitaciones y Futuras Mejoras

El modelo muestra un rendimiento prometedor, pero identificamos áreas clave para optimización.

Rendimiento en Validación

Los resultados obtenidos en las fases de validación demuestran la alta precisión del modelo en la detección de placas bajo condiciones óptimas.



Casos de Fallo Identificados

- **Ángulos Extremos:** Dificultad en detectar placas con inclinaciones o perspectivas muy pronunciadas.
- **Iluminación Deficiente:** La baja luz o los reflejos intensos pueden afectar la visibilidad de la placa.
- **Placas Pequeñas o Pixeladas:** Imágenes de baja resolución o placas muy alejadas representan un desafío.

Ideas de mejora futura: Ampliar el dataset con más variedad de condiciones, implementar algoritmos de preprocessamiento de imagen avanzados y explorar arquitecturas de modelo más robustas.

Impulsando la Movilidad Urbana con Visión por Computador

La gestión de la movilidad urbana y los aparcamientos se enfrenta a desafíos crecientes. Nuestro proyecto ha explorado cómo la visión por computador puede transformar estos problemas en soluciones eficientes y escalables. A continuación, presentamos los hallazgos clave de nuestra investigación y desarrollo.

Detección de Placas: Una Solución Efectiva

El sistema de detección de placas vehiculares ha demostrado ser una herramienta formidable para resolver problemas críticos en la movilidad urbana. Permite la reducción de los tiempos de búsqueda de aparcamiento y optimiza la monitorización en centros comerciales y otros espacios privados.

Flujo de Trabajo de Visión por Computador Moderno

Desde la creación del dataset hasta la evaluación y exportación del modelo, hemos validado un flujo de trabajo de visión por computador eficiente y adaptable. Este enfoque nos permite desarrollar soluciones robustas y versátiles para diversas plataformas y entornos.

Portabilidad Optimizada con TorchScript

La conversión del modelo a TorchScript ha mejorado significativamente su portabilidad, facilitando su ejecución en dispositivos de bajo consumo y sistemas embebidos. Esto asegura un rendimiento competitivo sin comprometer la eficiencia.

Oportunidades de Mejora

Aunque los resultados bajo condiciones controladas son excelentes, hemos identificado limitaciones en escenarios con iluminación deficiente, ángulos extremos o placas parcialmente visibles. Estas áreas representan claras oportunidades para futuras mejoras.

Demostración Interactiva en HuggingFace Spaces

El despliegue en HuggingFace Spaces ha permitido ofrecer una demostración interactiva y accesible, confirmando la viabilidad de integrar nuestro modelo en aplicaciones reales basadas en la nube y entornos IoT.

En conclusión, la visión por computador aplicada al reconocimiento de placas vehiculares se perfila como una solución escalable, automatizada y de alto impacto para los sistemas inteligentes de estacionamiento y la gestión vehicular del futuro.