

# Reconocimiento de Placas Vehiculares Peruanas en Ruta: Dataset Propio y Modelo de Dos Etapas

Joel Ibaceta

Universidad Nacional de Ingeniería

joel.ibaceta@email.com

Segundo Autor

Nombre de su institución

segundo.autor@email.com

Tercer Autor

Otra institución

tercer.autor@email.com

**Abstract**—Los sistemas de reconocimiento automático de placas vehiculares (ALPR) han alcanzado altos niveles de precisión en contextos controlados, pero presentan una significativa degradación de desempeño en entornos reales no estructurados, caracterizados por variabilidad en las condiciones de iluminación, movimiento de cámara y diversidad de configuraciones vehiculares. Esta problemática se agudiza en países como Perú, donde se observan particularidades adicionales como placas laterales pintadas, formatos extendidos y elementos visuales atípicos. En este trabajo se propone un enfoque jerárquico en dos etapas para la detección robusta de placas: primero se segmentan las regiones vehiculares utilizando un modelo YOLOv7 entrenado sobre una adaptación del conjunto de datos KITTI; posteriormente, se ejecuta una detección localizada de placas sobre dichas regiones, restringiendo el contexto visual y reduciendo la tasa de falsos positivos. Adicionalmente, se introduce un conjunto de datos curado manualmente a partir de grabaciones reales, anotado mediante un proceso asistido por OCR. Los resultados experimentales evidencian mejoras sustanciales en escenarios complejos, permitiendo identificar placas poco convencionales y múltiples instancias simultáneas en escenas densas de tráfico.

## I. INTRODUCCIÓN

El reconocimiento automático de placas vehiculares (ALPR) es ampliamente utilizado en control vehicular y seguridad pública. Sin embargo, su desempeño se ve limitado cuando se aplica en contextos distintos al de su entrenamiento original. En el caso peruano, modelos como ALPR-Unconstrained presentan errores al enfrentar placas con elementos particulares como la palabra “PERÚ”, o al confundirse con calcomanías y textos en el parabrisas. Además, muestran dificultades al operar desde una perspectiva frontal y móvil, típica de patrullaje o monitoreo embarcado, a diferencia de la vista elevada de muchos datasets disponibles.

La mayoría de estos modelos han sido entrenados con cámaras fijas y condiciones de iluminación controladas, lo que limita su capacidad de generalización a escenarios dinámicos. No obstante, el contexto local ofrece oportunidades únicas: en Perú, varios vehículos cuentan con placas pintadas en los laterales o parte trasera, que pueden servir como fuente redundante cuando la placa principal no es visible.

Este trabajo propone una solución adaptada a estas condiciones. Se presenta un nuevo dataset capturado desde la perspectiva del conductor, curado manualmente, junto con un modelo de detección entrenado sobre estas imágenes. El enfoque apunta a mejorar la robustez en escenarios reales de

monitoreo en movimiento, donde los sistemas tradicionales fallan.

## II. TRABAJOS RELACIONADOS

Los sistemas de reconocimiento automático de placas vehiculares (ALPR) han sido tradicionalmente entrenados y evaluados sobre datasets tomados desde cámaras fijas, usualmente en altura, bajo condiciones controladas. Ejemplos ampliamente utilizados incluyen CCPD [1], OpenALPR y EasyOCR. Sin embargo, su rendimiento disminuye cuando se aplican en contextos no estructurados o con perspectivas móviles.

Un trabajo destacado en América Latina es el dataset UFPR-ALPR [2], desarrollado en Brasil, el cual contiene imágenes capturadas desde vehículos en movimiento y motivó parte de nuestra propuesta como un equivalente adaptado al contexto peruano. De forma similar, el sistema PatrolVision [3] demuestra la viabilidad de soluciones embarcadas para entornos urbanos dinámicos, utilizando arquitecturas tipo YOLO para ALPR desde patrullas móviles.

Nuestro trabajo se inspira en estos enfoques, pero se enfoca específicamente en las condiciones particulares del Perú: placas con la inscripción “PERÚ”, presencia de placas laterales pintadas, y escenas grabadas desde vehículos en ruta bajo condiciones reales de iluminación, ángulos y tráfico local.

## III. PREPARACIÓN DEL DATASET

### A. Recolección de datos

Para construir un dataset representativo de las condiciones reales, se seleccionaron tres videos públicos grabados desde el interior de un vehículo en movimiento. Estos registros abarcan distintos contextos de conducción —tráfico urbano y carretera— y franjas horarias (mañana, tarde y noche), con variaciones naturales de iluminación, clima y dinámica vehicular.

La elección de fuentes reales responde a la necesidad de capturar imágenes con ruido, ángulos obstruidos y condiciones no controladas, características típicas de un entorno operativo como el patrullaje móvil.

### B. Procedimiento de construcción

Cada cuadro fue procesado mediante un sistema de detección de vehículos utilizando el modelo YOLOv7 [4], previamente entrenado sobre el dataset KITTI [5]. Esta etapa

TABLE I  
RESUMEN DE LA RECOLECCIÓN DE DATOS

Detalle	Valor
Total de videos utilizados	3
Duración acumulada	75 minutos
Frecuencia de muestreo	5 cuadros por segundo
Total de frames procesados	22,500

permitió generar recortes individuales de cada automóvil visible en la escena. El objetivo de esta segmentación previa fue acotar espacialmente el área de búsqueda del sistema OCR, lo que introdujo una restricción semántica basada en contexto visual. De esta manera, se redujeron significativamente los falsos positivos causados por textos ajenos al dominio, como anuncios, carteles, o señalización vial, y se disminuyó la complejidad del entorno visual a procesar, favoreciendo la especialización del sistema de reconocimiento de placas.

Sobre cada recorte vehicular se aplicó PaddleOCR [?], un sistema robusto de reconocimiento óptico de caracteres que devuelve múltiples candidatos textuales por imagen. A diferencia de los enfoques tradicionales que abordan la detección de placas como un problema de segmentación de objetos, esta propuesta invierte la lógica convencional: se parte del texto detectado y, si alguno de los candidatos coincide con patrones sintácticos predefinidos (por ejemplo, combinaciones alfanuméricas como ABC-123 o EV-123), se infiere que el área que lo rodea corresponde a una posible placa vehicular.

Una vez validado el patrón textual, se toma su área de bounding box y se genera un recorte expandido mediante padding, ajustando su proporción a las dimensiones habituales de una matrícula. Este procedimiento fue automatizado para garantizar consistencia visual en los recortes generados, los cuales conforman el conjunto final de imágenes del dataset.

Finalmente, se implementó un pequeño script de revisión asistida para facilitar la curaduría manual de los recortes. Este permitió validar visualmente los textos detectados y su correspondencia con la imagen recortada, descartando falsos positivos y garantizando la calidad del conjunto de entrenamiento.

### C. Resultados y almacenamiento

Para evitar el sobreentrenamiento en placas repetidas, se impuso una política de exclusión que limita a tres el número máximo de veces que una misma placa puede ser incluida en el dataset. Esta regla se implementó mediante un sistema de conteo por hash del texto reconocido, complementado con una revisión manual durante la curaduría asistida.

Dado que el modelo YOLOv7 requiere un formato específico para su entrenamiento —consistente en imágenes con sus respectivas anotaciones en archivos de texto plano donde cada línea representa una clase, coordenadas normalizadas del bounding box (centro, ancho y alto)— se diseñó una rutina de conversión automática que traduce cada recorte detectado al formato esperado por el modelo. Esto asegura compatibilidad directa con los procesos de entrenamiento y evaluación

del pipeline de YOLOv7 sin necesidad de transformaciones adicionales.

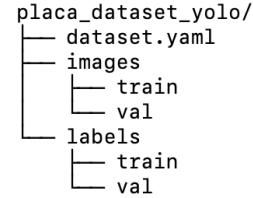


Fig. 1. Estructura de directorios del dataset generado. Se organiza en carpetas de entrenamiento y validación, junto a sus respectivas anotaciones en formato YOLO.

La estructura generada se muestra en la Figura 1, donde se aprecia la separación entre datos de entrenamiento y validación, junto a sus anotaciones correspondientes.

## IV. ENTRENAMIENTO DE LOS MODELOS

Para validar la viabilidad del enfoque propuesto, se llevaron a cabo dos etapas de entrenamiento utilizando modelos basados en YOLOv7. La primera etapa consistió en entrenar un modelo segmentador de vehículos, y la segunda en entrenar un modelo segmentador de placas utilizando los datos generados automáticamente mediante el método descrito anteriormente.

### A. Entrenamiento del modelo de segmentación de vehículos

Para la detección de vehículos en escenas urbanas, se entrenó un modelo YOLOv7 [4] empleando el conjunto de datos KITTI [?], el cual incluye anotaciones para objetos como car, van y truck. Estas tres clases fueron consideradas de forma independiente durante el entrenamiento, aunque en etapas posteriores del pipeline serán tratadas como una única categoría general (vehículo) para efectos de segmentación y recorte.

El entrenamiento se llevó a cabo durante 100 épocas utilizando la arquitectura base yolov7.pt, con tasa de aprendizaje definida en hyp.scratch.p5.yaml, y un tamaño de lote de 16. El proceso se realizó sobre GPU y fue monitoreado mediante la plataforma Weights & Biases (W&B), lo que permitió registrar y analizar la evolución de las principales métricas y funciones de pérdida.

Los resultados obtenidos reflejan una convergencia adecuada del modelo.

TABLE II  
MÉTRICAS DE DESEMPEÑO DEL MODELO YOLOV7 SOBRE EL CONJUNTO DE VALIDACIÓN (DETECCIÓN DE VEHÍCULOS).

Métrica	Valor
Precisión	0.96
Recall	0.97
mAP@0.5	0.98
mAP@0.5:0.95	0.84

## B. Entrenamiento del Segmentador de Placas (YOLOv7 sobre Dataset Generado)

Para detectar placas dentro de recortes vehiculares, se entrenó un modelo YOLOv7 utilizando un dataset personalizado generado mediante una estrategia de detección de texto, según el procedimiento descrito en la sección anterior. Este dataset incluye imágenes de placas peruanas con características particulares, como la palabra "PERÚ" y placas pintadas en los laterales de algunos vehículos.

El modelo fue entrenado durante 100 épocas utilizando la arquitectura `yolov7.pt`, imágenes de  $640 \times 640$  píxeles, tasa de aprendizaje basada en `hyp.scratch.p5.yaml`, batch de tamaño 16 y cacheo de imágenes activado. El entrenamiento fue monitorizado con W&B.

TABLE III

MÉTRICAS DE DESEMPEÑO DEL MODELO YOLOV7 SOBRE EL CONJUNTO DE VALIDACIÓN (DETECCIÓN DE PLACAS).

Métrica	Valor
Precisión	0.980
Recall	0.992
mAP@0.5	0.988
mAP@0.5:0.95	0.846

Las gráficas obtenidas evidencian un comportamiento adecuado del modelo:

train/box\_loss

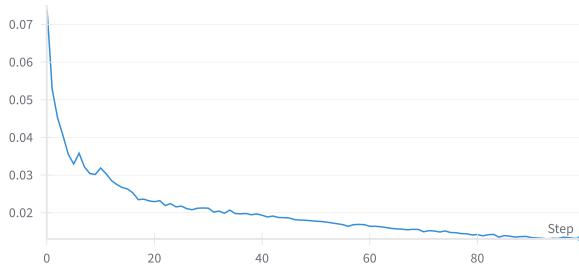


Fig. 2. Curva de pérdida de las cajas durante el entrenamiento (train/box\_loss).

`train/box_loss`: muestra una disminución sostenida, lo que indica un aprendizaje efectivo en la regresión de *bounding boxes* (ver Figura 2).

val/box\_loss

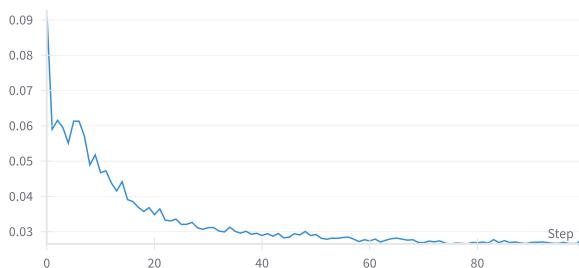


Fig. 3. Curva de pérdida de las cajas en validación (val/box\_loss).

`val/box_loss`: desciende progresivamente, lo que sugiere una correcta generalización en el conjunto de validación (ver Figura 3).

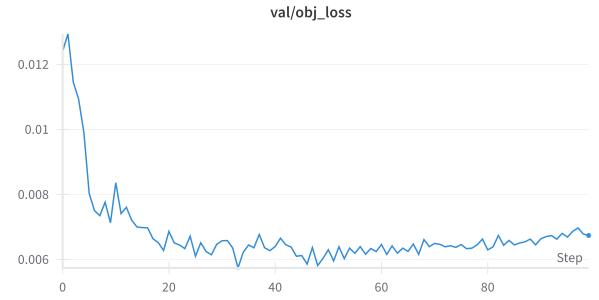


Fig. 4. Curva de pérdida del objeto en validación (val/obj\_loss).

`val/obj_loss`: decrece con ciertas oscilaciones y se estabiliza, lo que puede reflejar la variabilidad en las posiciones y tamaños de las placas (ver Figura 4).

La Figura 5 presenta una muestra de las detecciones generadas automáticamente por la plataforma W&B al finalizar el entrenamiento. Esta visualización corresponde a una inferencia masiva del modelo segmentador sobre imágenes del conjunto de validación, permitiendo verificar de forma rápida y estructurada el rendimiento cualitativo del detector.



Fig. 5. Grid automático de validación generado por W&B tras la última época de entrenamiento. Se muestran predicciones sobre el conjunto de validación con sus respectivas confidencias.

## V. RESULTADOS CUALITATIVOS

Además de las métricas cuantitativas, se presentan a continuación varios casos que ilustran el comportamiento del sistema propuesto ante situaciones reales representativas del contexto vial peruano. Estos ejemplos fueron seleccionados por su relevancia práctica y por evidenciar escenarios en los que los enfoques tradicionales suelen fallar, como la presencia de múltiples placas en un solo vehículo, placas pintadas en lugar de metálicas, o ubicaciones no convencionales.

A través de estas muestras, se busca evidenciar no solo la capacidad del modelo para detectar correctamente las placas en condiciones visuales variadas, sino también su robustez

semántica al evitar falsos positivos en textos adyacentes, y su capacidad para resolver ambigüedades utilizando información redundante disponible en los vehículos (como placas duplicadas o versiones pintadas de la misma).



Fig. 6. Placa pintada detectada en un camión de carga. Aunque la placa metálica inferior es ilegible, el sistema logra identificar correctamente la versión pintada en la parte superior.

La Figura 6 muestra un caso frecuente en vehículos pesados del contexto peruano: la presencia de una placa pintada legible y otra metálica oscurecida o deteriorada. El enfoque propuesto permite segmentar ambas regiones, y la validación OCR prioriza la que presenta mayor confianza, contribuyendo a la robustez del sistema, es importante destacar que se discrimina la región derecha de la imagen, que corresponde a un texto no relacionado con una placa vehicular a pesar de que el estilo sea similar.



Fig. 7. Vehículo con doble placa (metálica y pintada). El sistema detecta ambas instancias y verifica que contienen el mismo texto, validando así la consistencia de los resultados.

La Figura 7 evidencia un caso donde un vehículo cuenta con dos instancias de la placa trasera, una pintada y una metálica. El modelo segmentador detecta ambas de forma independiente, y el OCR confirma su correspondencia, reforzando la fiabilidad del proceso de extracción.



Fig. 8. Camión recolector de residuos con placa en una ubicación no convencional.

Por último, la Figura 8 ilustra un caso poco convencional: un camión recolector de residuos cuya placa se encuentra colocada en una posición no estándar, alejada del centro inferior visual típico. A pesar del emplazamiento atípico se observa que el modelo logra detectar la placa correctamente.



Fig. 9. Robustez del sistema ante múltiples condiciones urbanas. Se detectan correctamente varias placas en distintos planos, horarios y tipos de vehículos, incluyendo tráfico nocturno y vehículos con placas pintadas.

La Figura 9 ilustra la capacidad del sistema propuesto para operar en entornos urbanos reales bajo diferentes condiciones. Se presentan escenas diurnas y nocturnas, con tráfico denso, occlusiones parciales y variedad de vehículos, incluyendo SUV, sedanes y camiones.

## VI. CONCLUSIONES

Este trabajo presentó un enfoque robusto para el reconocimiento de placas vehiculares peruanas capturadas en condiciones reales de tránsito desde una perspectiva móvil. A diferencia de muchos modelos entrenados sobre datasets idealizados, nuestro sistema demuestra una alta capacidad de generalización ante escenas con múltiples vehículos, variabilidad en iluminación, ángulos no convencionales, y presencia de placas pintadas.

La estrategia basada en dos etapas —detección de vehículos seguida de segmentación de placas— permitió reducir significativamente los falsos positivos asociados a textos no vehiculares. La generación automatizada del dataset a partir de detección OCR validada sintácticamente, complementada con curaduría manual, garantizó la calidad del entrenamiento sin requerir costosas anotaciones desde cero.

Los resultados alcanzados por el modelo segmentador de placas, con una precisión superior al 97%

Este trabajo podría sentar bases para futuras líneas de desarrollo que incluyan la incorporación de reconocimiento OCR end-to-end en tiempo real y la expansión del dataset con imágenes de otras regiones del país que presenten particularidades locales usando la misma técnica de recolección y curaduría.

## REFERENCES

- [1] Z. Xu, B. Yang, G. Zhou, X. Wang, and Y. Wu, “Towards end-to-end license plate detection and recognition: A large dataset and baseline,” *arXiv preprint arXiv:1802.10130*, 2018.
- [2] G. P. Gonçalves, S. M. Silva, R. Laroca, E. Severo, and D. Menotti, “UFPR-ALPR dataset: A new dataset for Brazilian license plate recognition,” *Iberoamerican Conference on Pattern Recognition (CIARP)*, Springer, 2020, pp. 400–408.
- [3] W. Liu, H. Zhang, K. Tan, and C. S. Foo, “PatrolVision: Real-Time Vehicle License Plate Recognition System for Urban Traffic Management,” *arXiv preprint arXiv:2504.10810*, 2024.
- [4] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,” *arXiv preprint arXiv:2207.02696*, 2022.
- [5] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354–3361.
- [6] PaddlePaddle Team, “PaddleOCR: An ultra lightweight OCR system, developed by PaddlePaddle,” *GitHub Repository*, 2020. [Online]. Available: <https://github.com/PaddlePaddle/PaddleOCR>. [Accessed: 05-Jun-2025].