

Computer Vision System for License Plate Recognition and Reading

Helguero Velásquez, Hernán Luis¹[0000-0002-3952-697X] and Ancalle Yucra Diego Orlando², Guzmán Gonzales Fernando², Gonzales Casanova Miguel Ángel²

¹ Facultad Nacional de Ingeniería, Universidad Técnica de Oruro, Oruro, Bolivia
hernan.helguero@sistemas.edu.bo

² Facultad Nacional de Ingeniería, Universidad Técnica de Oruro, Oruro, Bolivia

Abstract. This research paper presents the development of an automatic license plate recognition (ANPR) system based on computer vision and deep learning. The system uses the YOLOv8 model in segmentation mode, trained on a Bolivian license plate dataset, to detect and extract license plates from frames captured by an IP camera. Image processing with OpenCV is then applied to correct license plate orientation and perspective, improving character reading accuracy. For optical character recognition (OCR), the Microsoft TrOCR model is implemented, demonstrating high efficiency even with low-resolution images and without intensive preprocessing. The system was developed and tested on a laptop with modest hardware specifications, showing promising results that could be improved with greater computational power. The results indicate a significant improvement in license plate detection and recognition compared to traditional methods.

Keywords: Recognition, License Plates, Computer Vision.

Sistema de Visión por Computador para el Reconocimiento y Lectura de Placas Vehiculares

Resumen. En este trabajo de investigación se presenta el desarrollo de un sistema de reconocimiento automático de placas vehiculares (ANPR) basado en visión por computador y aprendizaje profundo. El sistema utiliza el modelo YOLOv8 en modo segmentación, entrenado con un conjunto de datos de placas bolivianas, para detectar y extraer las placas vehiculares desde frames capturados por una cámara IP. Posteriormente, se aplica un procesamiento de imágenes con OpenCV para corregir la orientación y perspectiva de las placas, mejorando la precisión en la lectura de caracteres. Para el reconocimiento óptico de caracteres (OCR), se implementa el modelo TrOCR de Microsoft, el cual demuestra una alta eficacia incluso con imágenes de baja resolución y sin procesamiento intensivo. El sistema fue desarrollado y probado en una laptop con especificaciones de hardware modestas, mostrando resultados prometedores que po-

drían mejorar con mayor potencia computacional. Los resultados obtenidos indican una mejora significativa en la detección y reconocimiento de placas vehiculares en comparación con métodos tradicionales.

Palabras clave: Reconocimiento, Placas Vehiculares, Visión por Computador.

1 Introducción

El reconocimiento automático de placas vehiculares (ANPR, por sus siglas en inglés) es una tecnología clave en aplicaciones como control de tráfico, seguridad vial, gestión de estacionamientos y cobro automatizado de peajes (Hrabina, 2018; Du et al., 2013; Robinson y Venter, 2019). Los sistemas ANPR enfrentan desafíos significativos debido a variaciones en condiciones de iluminación, ángulos de captura, diversidad en los formatos de las placas y limitaciones de hardware (Kang et al., 2018; Gai- kwad et al., 2014; Álvarez y Salas, 2011).

Con el avance del aprendizaje profundo y las técnicas de visión por computador, se han desarrollado sistemas más robustos y precisos (Du et al., 2013; Salma et al., 2021; Silva y Jung, 2020). En particular, la utilización de modelos de detección y segmentación avanzados, combinados con OCR basados en redes neuronales profundas, ha mostrado mejoras notables en la precisión y eficiencia de los sistemas ANPR (Zhai et al., 2012; Li et al., 2021; Silva y Jung, 2020).

En este trabajo, se presenta un sistema ANPR desarrollado con el microframework Flask para Python, integrando módulos de gestión de usuarios, conductores, cámaras, detecciones y registro de actividad de usuarios. El enfoque principal se centra en la detección y lectura de placas vehiculares utilizando YOLOv8 en modo segmentación y el modelo TrOCR para OCR. Las contribuciones principales de este trabajo son: la implementación de un modelo de segmentación basado en YOLOv8 entrenado con un dataset de placas bolivianas, desarrollo de un proceso de corrección de perspectiva y orientación de las placas utilizando OpenCV, integración del modelo TrOCR para mejorar la precisión en la lectura de caracteres en placas vehiculares, y evaluación del rendimiento del sistema en hardware de recursos limitados y discusión sobre escalabilidad con hardware más potente.

El resto del artículo está organizado de la siguiente manera: en la Sección II se revisan los trabajos relacionados; la Sección III describe la metodología propuesta; en la Sección IV se presentan los resultados obtenidos; la Sección V discute los hallazgos y considera las implicaciones del hardware utilizado; y, finalmente, en la Sección VI se ofrecen las conclusiones y perspectivas futuras.

2 Trabajos Relacionados

Numerosos estudios han abordado el reconocimiento de placas vehiculares utilizando diversas técnicas de visión por computador y aprendizaje automático. En Kang et al. (2018), proponen un sistema ANPR basado en una sola cámara para mejorar el reconocimiento de placas en carreteras de múltiples carriles, utilizando técnicas de procesamiento de imágenes y binarización adaptativa. Frydson y Gutiérrez (2011) aplican

visión por computador para el reconocimiento de placas utilizando OCR convencionales como Tesseract, enfrentando desafíos en condiciones no controladas.

Hrabina (2018) compara tecnologías de identificación vehicular como RFID y ANPR, destacando las ventajas y desventajas de cada una en aplicaciones de peaje y estacionamiento. Du et al. (2013) presentan una revisión exhaustiva de los sistemas ALPR, analizando las etapas clave y los desafíos asociados.

Tripathi et al. (2021) y Gaikwad et al. (2014) implementan sistemas ANPR basados en procesamiento de imágenes con OpenCV y OCR tradicionales, pero enfrentan limitaciones en precisión bajo condiciones variables. Males y Rojas (2015) desarrollan un sistema de reconocimiento de matrículas en un entorno universitario utilizando visión artificial y hardware de bajo costo.

Negri et al. (2010), proponen un sistema LPR robusto que combina la estructura morfológica, gradientes y un análisis espectral, para detectar placas vehiculares en imágenes fijas. Su enfoque es eficaz, detectando en placas deterioradas.

La mayoría de estos trabajos resaltan las limitaciones en la lectura de caracteres debido a rotaciones, baja resolución o condiciones de iluminación adversas (Álvarez y Salas, 2011; Zhai et al., 2012; Silva y Jung, 2020). Salma et al. (2021) proponen un marco ANPR para placas paquistaníes utilizando detección de objetos y OCR, mostrando mejoras significativas en precisión al utilizar modelos de aprendizaje profundo.

Silva y Jung (2020) presentan un sistema de detección y reconocimiento de placas en tiempo real utilizando redes neuronales convolucionales profundas, lo que demuestra mejoras notables en precisión y eficiencia.

En este trabajo, el enfoque se diferencia al utilizar el modelo YOLOv8 en modo segmentación y el OCR TrOCR de Microsoft (Li et al., 2021), que ha mostrado alta eficacia sin necesidad de preprocesamiento intensivo, incluso en imágenes de baja calidad y rotadas.

3 Metodología

3.1 Configuración del hardware

El sistema fue desarrollado y probado en una laptop HP con un Procesador Intel Core i7-8550U @ 1.80GHz (8 CPUs), una Memoria RAM de 16 GB, GPU NVIDIA GeForce MX150 con 2 GB de VRAM y con un Sistema Operativo: Windows 11 de 64 bits. Esta configuración representa un hardware de gama media, lo que demuestra la viabilidad del sistema en entornos con recursos limitados. Se anticipa que, con hardware más potente, el rendimiento del sistema podría mejorar significativamente (Gurney et al., 2012).

3.2 Dataset y Entrenamiento del Modelo

Se recopiló un dataset de placas vehiculares bolivianas, capturando imágenes en diversas condiciones de iluminación, ángulos y tipos de vehículos, siguiendo metodolo-

gías de Miranda (2019), Álvarez y Salas (2011) y Rojas y Aracena (2013). El dataset fue anotado manualmente para incluir las máscaras de segmentación de las placas.

El dataset está compuesta por 508 imágenes de los cuales 60% son imágenes reales de placas, 30% son datos aumentados en base a las imágenes originales, 10% son datos sintéticos. En la Fig. 1 se muestra algunas placas del dataset.



Fig. 1. Muestra de imágenes del dataset.

El modelo YOLOv8 en modo segmentación fue entrenado utilizando este dataset. Se utilizó una partición de 80% para entrenamiento y 20% para validación. El entrenamiento se realizó utilizando la librería de Ultralytics para Python, y se aprovechó la aceleración por GPU proporcionada por Google Colaboratory.

Se utilizó YOLOv8n, una versión nano optimizada para operar en dispositivos con recursos limitados. El modelo fue entrenado durante 250 épocas, utilizando imágenes redimensionadas a 640x640 píxeles. Se empleó un tamaño de lote de 100, el cual permite procesar múltiples imágenes por iteración, optimizando el uso de los recursos disponibles. Además, se habilitó el almacenamiento en memoria (cache = True) para agilizar el acceso a los datos durante el entrenamiento, pero aumentando el consumo de la memoria RAM. La tarea final es que el modelo aprenda a identificar el objeto dentro de la imagen, en este caso para que detecte la placa.

El tiempo del entrenamiento duró 2.148 horas (2 horas, 8 minutos y 8 segundos) y el modelo fue guardado como "best.pt".

3.3 Captura y Preprocesamiento de Imágenes

El sistema captura frames en tiempo real desde una cámara IP, siguiendo prácticas descritas en Navacerrada (2017) y De la Rosa Martín (2021). Cada frame es enviado al modelo YOLOv8 para detectar la presencia de placas vehiculares. Si se detecta una placa, se procede al procesamiento de la imagen.

En la figura Fig. 2 se muestra el Flujo del proceso de detección de placas, donde se captura la imagen del automóvil en movimiento, se envía al modelo, y se debe detectar la placa. Si fuese así, se procesa la imagen, almacenando los datos necesarios y luego se continúa con el mismo proceso.

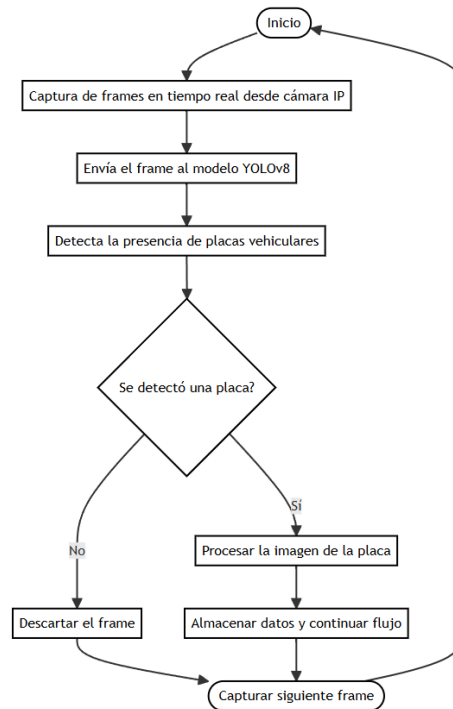


Fig. 2. Flujo básico de detección de placas.

En la Fig. 3 se muestran las detecciones que realiza el modelo YOLOv8n sobre las imágenes obtenidas.



Fig. 3. Detección de la placa con el modelo de segmentación YOLOv8n.

3.4 Corrección de Orientación y Perspectiva

Las placas detectadas pueden presentar rotaciones o perspectivas que afectan la precisión del OCR (Álvarez y Salas, 2011; Rojas y Aracena, 2013). Para corregir este problema, se utiliza el resultado de segmentación obtenido mediante YOLOv8n, del cual se extrae el contorno de la placa. Luego se calcula el rectángulo de área mínima que encierra dicho contorno utilizando la función `cv2.minAreaRect()`, lo que permite iden-

tificar sus cuatro vpertrices principales. Finalmente, se aplica una transformación de perspectiva con la función `cv2.warpPerspective()`, alineando la imagen de la placa y generando una representación rectangular corregida más adecuada, siguiendo las recomendaciones de Álvarez y Salas (2011) y Rojas y Aracena (2013).

En la Fig. 4 se muestra la aplicación de proceso de corrección de perspectiva de la placa.



Fig. 4. Proceso de corrección de perspectiva

3.5 Procesamiento para OCR

Aunque el modelo TrOCR demostró eficacia sin preprocesamiento intensivo, se aplicaron técnicas básicas de OpenCV para mejorar la calidad de la imagen. Inicialmente, la imagen se convirtió a escala de grises mediante la función `cv2.cvtColor()`, luego se aplicó un desenfoque Gaussiano con `cv2.GaussianBlur()`, lo que permitió suavizar el fondo y resaltar las regiones de interés, finalmente, se realizó una binarización utilizando Otsu a través de la función `cv2.threshold()`, que segmenta la imagen en blanco y negro, ajustando el umbral según la intensidad. Este proceso contribuye a resaltar los caracteres y reducir el ruido, mejorando la precisión del reconocimiento óptico de caracteres, siguiendo recomendaciones de Talledo y Gutiérrez (2011) y Navacerrada (2017).



Fig. 5. procesamiento de la imagen para lectura del OCR

En la Fig. 5 se muestra los detalles del procesamiento OCR, primero en la imagen se tiene la imagen con ajuste de perspectiva, luego la siguiente aplicado la escala de

grises, luego tiene aplicado un blur y finalmente la binarización de la imagen junto con otsu.

3.6 Implementación del OCR con TrOCR

Se utilizó el modelo trocr-base-printed de Microsoft, integrado a través de la librería Hugging Face Transformers. TrOCR es un modelo basado en Transformer que combina técnicas de atención para el reconocimiento de texto en imágenes (Li et al., 2021). Su capacidad para manejar imágenes de baja resolución y rotadas lo hace adecuado para este sistema.

TrOCR tiene el siguiente funcionamiento:

- Imagen de entrada: La imagen manuscrita se recibe como entrada.
- División en parches: Se divide en bloques pequeños (ej. 16x16 píxeles).
- Embedding de parches: Cada parche se transforma en un vector utilizando embeddings de posición y características.
- Codificador de imágenes: Utiliza un Transformer basado en el modelo BEiT para generar una representación codificada.
- Representación codificada: Es el resultado del procesamiento del codificador, que resume la información relevante de la imagen.
- Decodificador de texto: Utiliza un Transformer basado en RoBERTa para generar texto de manera autoregresiva.
- Texto transcrito: Salida final del modelo, que es la transcripción de la imagen.

En la Fig. 6 se muestra el flujo de básico de procesamiento. La principal ventaja radica en el uso de mecanismos de atención dentro del Transformer, los cuales permiten al sistema “enfocarse” en distintas partes de la imagen durante la codificación. Esto habilita la detección de detalles significativos, incluso en condiciones de poca iluminación o con cambios de orientación. Además, el uso de embeddings de posición contribuye a preservar la estructura espacial de los caracteres, lo que mejora la robustez del modelo frente a rotaciones y deformaciones leves.

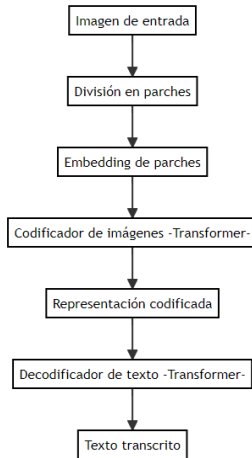


Fig. 6. Flujo básico del procesamiento de TrOCR.

3.7 Métricas de Evaluación

Para evaluar el rendimiento del modelo, se utilizan las métricas estándar de visión por computador, tales como precisión, recall, F1-score y mAP@0.5 (mean Average Precision a un IoU de 0.5). La precisión mide la proporción de verdaderos positivos sobre el total de predicciones positivas, el recall evalúa la capacidad del modelo para identificar correctamente los objetos. (Du et al., 2013) El F1-score combina ambas métricas en una media armónica que refleja el equilibrio entre precisión y recall. (Silva y Jung, 2020) Y, el mAP@0.5 calcula el promedio de precisión entre las predicciones y las etiquetas reales (considerando una superposición mínima del 50% - IoU mayor a 0.5).

3.8 Arquitectura del Sistema

El sistema está desarrollado con Flask y cuenta con módulos para: **Gestionar usuarios** que permite Registrar, Editar, Eliminar y Administrar las cuentas de usuarios que tienen acceso al sistema, definiendo roles y permisos específicos; **Gestionar cámaras** que Facilita la configuración, activación, desactivación y monitoreo de cámaras IP integradas al sistema para la captura de imágenes y videos; **Gestionar conductores** para Administrar la información de los conductores detectados (nombre, número de licencia y vehículos asociados); **Ver detecciones** donde Muestra un registro en tiempo real o histórico de las placas vehiculares detectadas, incluyendo detalles como fecha, hora, imagen y cámara utilizada; **Administración del panel de control** para Centralizar el acceso a funcionalidades principales del sistema, brindando una vista general de cámaras activas, estadísticas rápidas y acciones administrativas; **Mostrar reportes** Genera, visualiza reportes detallados sobre detecciones, usuarios y actividades del sistema en formatos descargables, como PDF o Excel; **Mostrar estadísticas** Presenta gráficos y métricas claves relacionadas a las detecciones, cámaras y actividades del sistema para análisis y toma de decisiones; y **Ver actividad de usuarios** que Registra y muestra un historial de acciones realizadas por los usuarios, como inicios de sesión, configuraciones de cámaras y generación de reportes.

En la Fig. 7 se muestra la vista general del sistema.



Fig. 7. Vista general del sistema.

La arquitectura general se ilustra en la Fig. 8.

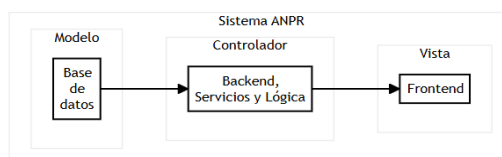


Fig. 8. Diagrama de bloques de la arquitectura del sistema ANPR desarrollado.

4 RESULTADOS

4.1 Evaluación del Modelo YOLOv8

El modelo entrenado alcanzó una precisión promedio (mAP@0.5) del 95% en la detección y segmentación de placas. En la Fig. 9 se muestran ejemplos de detección en diferentes condiciones.



Fig. 9. Detecciones en diversas condiciones.

Para evaluar el rendimiento del modelo YOLOv8, se analizaron las curvas de métricas clave relacionadas con la detección de placas vehiculares con el uso de TensorBoard.

La Fig. 10 muestra el equilibrio entre precisión y recall en función del nivel de confianza. El modelo alcanzó un F1-score de **0.98** con un nivel de confianza óptimo de **0.299**, lo que indica un balance efectivo entre la minimización de falsas detecciones y la maximización de aciertos.

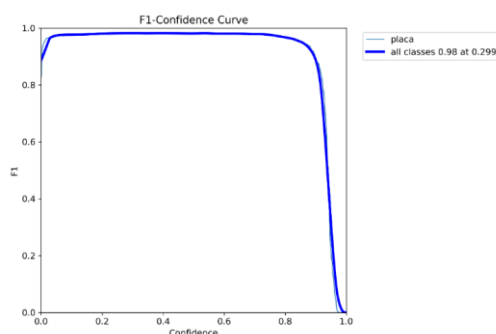


Fig. 10. Curva F1 – Curva de confianza.

La Fig. 11 presenta la variación de la precisión con respecto al nivel de confianza. El modelo logró una precisión del 100% con un nivel de confianza de 0.925, evidenciando su robustez en tareas de detección bajo diferentes niveles de certeza.

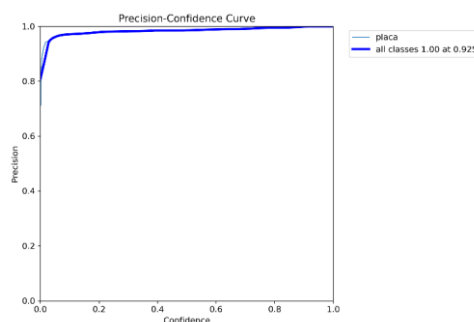


Fig. 11. Curva de precisión-confianza

La Fig. 12 ilustra la relación entre la precisión y el recall, obteniendo un valor mAP@0.5 de 0.991. Este resultado confirma la capacidad del modelo para realizar detecciones precisas mientras minimiza errores en escenarios complejos. A pesar de que el modelo tiene buenos resultados, no es tan efectivo en algunas condiciones como se muestra en la anterior Fig. 9, esto debido a que en el dataset para el entrenamiento no contenía imágenes de placas en esas condiciones, caso contrario pudo tener un mejor desempeño.

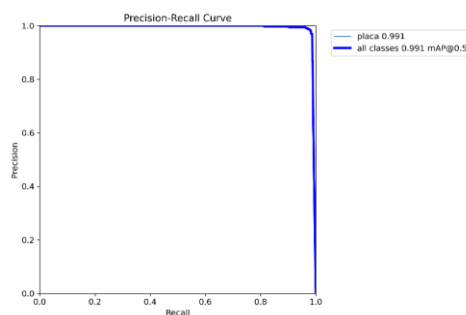















Fig. 12. Curva recall-precisión.

4.2 Desempeño del OCR

Para evaluar el desempeño de distintos motores de reconocimiento óptico de caracteres (OCR), se llevó a cabo una prueba de detección de texto en placas cuyas condiciones de lectura resultan particularmente desafiantes (baja iluminación, escasa resolución, inclinaciones, entre otros). En la Tabla 1 se comparan los resultados obtenidos con Pytesseract, EasyOCR y TrOCR, incluyendo el número de muestra, la imagen de la placa, sus dimensiones en píxeles (variando entre 50x31 y 436x232 píxeles, con un promedio aproximado de 186x120 píxeles) y la lectura final proporcionada por cada modelo.

Tabla 1. Comparación de OCRs

n°	placa	dimension (píxeles)	pytesseract	easyocr	trocr
1		50x31	-	[sooohr"	500onyr
2		82x70	-	hzmumha	3007hac
3		93x83	-	572ixz8	572xx2n
4		90x75	-	bo e 129zeee	129zeee
5		90x75	-	-	630 zcr
6		78x55	-	6244526	624676
7		118x72	-	270 pt	1270ptr
8		284x188	-	s115h8]	5765rnk]
9		436x232	-	b 0 l iv 5608uxr	5608uxr
10		354x220	-	5215 w	5215 dun
11		340x224	-	od 2267rfg	9267reg
12		264x177	-	j989h2c	4989hzc
13		321x182	-	bolivia 6272gyu	6272670
aciertos			0	3	9
porcentaje			0%	23%	69%

Esta prueba se realizó sobre un conjunto de imágenes de baja calidad, con el fin de resaltar las fortalezas y debilidades de cada OCR. En tales condiciones adversas, TrOCR mostró un mejor desempeño general, brindando lecturas más precisas que sus contrapartes.

Se observó, además, que al incrementar la resolución de las imágenes, todos los modelos mejoran de forma significativa su precisión de detección, lo que indica la importancia de contar con una adecuada calidad de imagen para optimizar el proceso de OCR.

5 Discusión

Los resultados demuestran que la combinación de YOLOv8 en modo segmentación y TrOCR mejora significativamente la precisión en el reconocimiento de placas vehiculares. A diferencia de métodos tradicionales que requieren preprocesamiento intensivo (Talledo y Gutiérrez, 2011; Gaikwad et al., 2014; Tripathi et al., 2021), nuestro enfoque maneja eficazmente imágenes con rotaciones y de baja resolución.

El rendimiento del sistema en hardware de gama media es prometedor. Sin embargo, se anticipa que, con hardware más potente, como GPUs de mayor capacidad como se muestra en la Fig 1., el tiempo de procesamiento podría reducirse significativamente, permitiendo aplicaciones en tiempo real en entornos más exigentes (Zhai et al., 2012).

Comparado con trabajos anteriores como Kang et al. (2018) y Du et al. (2013), nuestro sistema presenta ventajas en la precisión del OCR y en la capacidad de manejar condiciones variables sin preprocesamiento extensivo. Además, la utilización de modelos de aprendizaje profundo avanzados contribuye a la robustez del sistema.

Se identificaron limitaciones en condiciones de iluminación extremadamente adversas y en placas con daños físicos severos. Estos desafíos podrían abordarse mediante la incorporación de técnicas de mejora de imagen más avanzadas y la ampliación del dataset con más ejemplos de estas condiciones (Álvarez y Salas, 2011; Silva y Jung, 2020; Rojas y Aracena, 2013; Van de Vyvere y Colpaert, 2022).

6 Conclusiones y Trabajo Futuro

Se desarrolló un sistema ANPR eficiente y preciso, integrando técnicas avanzadas de visión por computador y aprendizaje profundo. El uso de TrOCR resultó clave para mejorar la lectura de caracteres en placas vehiculares, incluso en imágenes de baja calidad y rotadas.

El sistema demuestra viabilidad en hardware de gama media, y se anticipa que su rendimiento mejoraría con mayor potencia computacional. Esto abre posibilidades para su implementación en entornos donde los recursos de hardware son limitados.

Como trabajo futuro, se plantea:

Ampliar el Dataset: Incluir más imágenes en diversas condiciones para mejorar la robustez del modelo.

Optimización del Tiempo de Procesamiento: Implementar técnicas de paralelización y aprovechar hardware más potente.

Despliegue en Dispositivos Embebidos: Explorar la implementación en dispositivos como NVIDIA Jetson para aplicaciones en tiempo real (Quezada y Romero, 2016).

Mejoras en el Preprocesamiento: Incorporar técnicas avanzadas de mejora de imagen para condiciones de iluminación adversas (Rojas y Aracena, 2013; Van de Vyvere y Colpaert, 2022). Otra mejora podría ser el de eliminar la parte superior de la placa que contiene el texto “BOLIVIA”, ya que éste no forma parte del número de la placa, y puede interferir en la precisión del OCR, especialmente en placas tomadas con resoluciones bajas o donde pueda presentarse caracteres superpuestos.

Anonimización y Gestión de Datos: Considerar la anonimización de datos para cumplir con regulaciones de privacidad y explorar aplicaciones en análisis de tráfico (Van de Vyvere y Colpaert, 2022; Robinson y Venter, 2019).

Validación de la estructura de la placa: como se conoce que las placas bolivianas siguen un formato predefinido por cuatro números y tres letras, se puede incorporar una validación del OCR, que permitiría corregir errores y mejorar la precisión del reconocimiento, por ejemplo, si en los primeros cuatro caracteres reconoce una “O”, se debería cambiar por “0”.

References

- Kang, S.-I., Byun, S.-C., Yoon, Y.-H., & Park, Y. S. (2018). A study on single camera based ANPR system for improvement of vehicle number plate recognition on multi-lane roads. *Invention Journal of Research Technology in Engineering & Management (IJRTEM)*, 2(1), 11–16. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3267290
- Frydson Talledo, M. F., & Gutiérrez Pilay, R. A. (2011). Aplicación de visión por computador para el reconocimiento del número de placa de vehículos usando modelos de aprendizaje (OCR convencionales) [Tesis de grado, Escuela Superior Politécnica del Litoral]. <https://www.dspace.espol.edu.ec/bitstream/123456789/20617/1/T-83130%20FRYDSON-GUTIERREZ.pdf>
- Hrabina, M. (2018). Automated vehicle identification for tolling and parking: RFID vs ANPR [Whitepaper]. FEIG Electronics Inc. https://www.rfidreadernews.com/wp-content/uploads/2018/05/FEIG_AVI_WhitePaperFinal-1.pdf
- Du, S., Ibrahim, M., Shehata, M., & Badawy, W. (2013). Automatic license plate recognition (ALPR): A state of the art review. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(2), 311–325. <https://ieeexplore.ieee.org/document/6213519>
- Gaikwad, D. Y., Maiti, S., & Borole, P. B. (2014). Automatic number plate recognition system (ANPR system). *International Journal of Engineering Research & Technology (IJERT)*, 3(7), 1446–1450. <https://www.academia.edu/download/64349379/automatic-number-plate-recognition-system-IJERTV3IS071132.pdf>
- Tripathi, S., Shetty, S., Jain, S., & Sharma, V. (2021). Automatic number plate recognition system (ANPR): The implementation. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 10(8), 1–5. https://www.researchgate.net/publication/352561304_Automatic_Number_Plate_Recognition_System_ANPR_The_Implementation
- Negri, P., Tepper, M., Acevedo, D., Jacobo, J., & Mejail, M. (2010). Multiple clues for license plate detection and recognition. In I. Bloch & R.M. Cesar Jr. (Eds.), *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (CIARP 2010)* (pp. 269–276). Springer. https://doi.org/10.1007/978-3-642-16687-7_38
- Males Chalán, J. A., & Rojas Livisaca, R. F. (2015). Desarrollo de un sistema basado en visión artificial para el reconocimiento de matrículas vehiculares en el AEIRNNR de la UNL [Te-

- sis de grado, Universidad Nacional de Loja]. <https://dspace.unl.edu.ec/jspui/handle/123456789/11402>
- Salma, M., Shah, H., Arif, M., Khan, M., Khan, Z. A., & Khan, M. Z. (2021). Development of ANPR framework for Pakistani vehicle number plates using object detection and OCR. *Complexity*, 2021, Article ID 5597337, 1–14. <https://doi.org/10.1155/2021/5597337>
- Miranda León, R. A. (2019). Diseño de un sistema de asistencia vehicular avanzado [Tesis de grado, Instituto Politécnico Nacional]. https://tesis.ipn.mx/bitstream/handle/123456789/27149/Miranda%20Le%C3%B3n%2C%20Rafael%20Asrael_Tesis.pdf
- Choez Álvarez, C. L., & Salas Guerrero, S. F. (2011). Normalización de imágenes de placas vehiculares a través de corrección geométrica [Tesis de grado, Escuela Superior Politécnica del Litoral]. <https://www.dspace.espol.edu.ec/handle/123456789/45490>
- Zhai, Z., Bensaali, F., & Sotudeh, R. (2012). OCR-based neural network for ANPR. En 2012 IEEE International Conference on Imaging Systems and Techniques Proceedings (pp. 393–397). <https://doi.org/10.1109/IST.2012.6295581>
- Silva, S. M., & Jung, C. R. (2020). Real-time license plate detection and recognition using deep convolutional neural networks. *Journal of Visual Communication and Image Representation*, 71, 102773. <https://doi.org/10.1016/j.jvcir.2020.102773>
- Rojas Henríquez, J., & Aracena Pizarro, D. (2013). Segmentación de patentes vehiculares mediante técnicas de agrupamiento en ambientes externos. *Ingeniare. Revista Chilena de Ingeniería*, 21(2), 172–184. <https://doi.org/10.4067/S0718-33052013000200002>
- Navacerrada, J. (2017). Sistema de detección de matrículas con OpenCV [Trabajo de Fin de Grado, Universidad Politécnica de Madrid]. <https://oa.upm.es/51869/>
- De La Rosa Martín, T. (2021). Sistema de identificación de placas automotrices para la Universidad Metropolitana de Ecuador. *Revista Universidad y Sociedad*, 13(1), 242–252. http://scielo.sld.cu/scielo.php?pid=S2218-36202021000100242&script=sci_arttext
- Quezada Martínez, G., & Romero Martínez, E. J. (2016). Sistema embebido para el reconocimiento de placas vehiculares [Tesis de grado, Instituto Politécnico Nacional]. <https://tesis.ipn.mx/handle/123456789/22093>
- Gurney, R., Rhead, M., Lyons, V., & Ramalingam, S. (2012). The effect of ANPR camera settings on system performance. *School of Engineering & Technology, University of Hertfordshire*. <https://doi.org/10.1049/ic.2013.0276>
- Li, M., et al. (2021). TrOCR: Transformer-based optical character recognition with pre-trained models. *arXiv preprint, arXiv:2109.10282*. <https://arxiv.org/abs/2109.10282>
- Van de Vyvere, B., & Colpaert, P. (2022). Using ANPR data to create an anonymized linked open dataset on urban bustle. *European Transport Research Review*, 14(17), 1–16. <https://doi.org/10.1186/s12544-022-00538-1>
- Robinson, A., & Venter, C. (2019). Validating traffic models using large-scale automatic number plate recognition (ANPR) data. *Journal of the South African Institution of Civil Engineering*, 61(3), 45–57. <http://dx.doi.org/10.17159/2309-8775/2019/v61n3a5>