



## Reconocimiento de señales de tráfico en Cuba

Claudia Alvarez Martínez      Roger Moreno Gutiérrez  
Kevin Majim Ortega Alvarez      Jan Carlos Perez González

7 de julio de 2024

Facultad de Matemática y Computación.  
Universidad de la Habana

## Resumen

La detección y clasificación de señales de tránsito es una tarea para apoyar al conductor e incluso asistir en la navegación de un automóvil autónomo. El objetivo de este trabajo es presentar una metodología para la detección y clasificación de señales de tráfico cubanas mediante el uso de redes convolucionales. La metodología se divide en cinco etapas, 1-) se realiza la recolección de 615 de señales de tráfico en un ambiente no controlado, 2-) se realiza un proceso manual de marcado para la detección de señales de tráfico en las imágenes, 3-) se entrena una red neuronal (YOLOv8) en el conjunto GTSRB para obtener un conocimiento general de las características de las señales, 4-) se evalúa esa red neuronal en el conjunto de datos creado, 5-) se realiza un proceso de transferencia de conocimiento y aumentado de datos para la clasificación. De acuerdo con los resultados obtenidos de los experimentos se concluye que la metodología propuesta permite reconocer señales de tráfico cubanas con un *mAP@0.5* del 75 %, sin embargo tras la experimentación con filmaciones en tiempo real, se obtienen mejores resultados.

**Palabras clave:** Señales de tráfico, redes convolucionales, imágenes segmentadas, YOLOv8

## 1. Introducción

Las señales de tráfico son los signos visuales utilizados para ofrecerle información a los conductores y peatones que transitan por un camino. Las señales de tráfico se clasifican de acuerdo a sus colores y formas, de manera que sean llamativas para que los automovilistas les presten atención. Muchos países europeos han trabajado para estandarizar las señales de tráfico, lo que dio lugar a la convención de Viena del 8 de noviembre de 1968.

Particularmente en nuestro país, las señales de tránsito se distribuyen en 8 grupos[1], entre las que se encuentran algunas de prohibición, que por lo general son blancas y rojas en los bordes, las de peligro son triángulos amarillos o las de obligación que son redondas y de fondo azul.

### 1.1. Motivación

Es ampliamente reconocido por todos los conductores la importancia de obedecer las leyes de tránsito, prestando especial atención a las señales de tráfico. Las consecuencias de no hacerlo pueden ser desastrosas, especialmente cuando estas señales se encuentran en mal estado debido a las inclemencias del clima, la mala iluminación, o simplemente no se ven a tiempo. Por ello, es crucial desarrollar un sistema automatizado que pueda detectar de forma rápida y eficiente estas señales. Además, este sistema puede ser utilizado como asistencia a conductores, mejorando significativamente la seguridad en las vías.

### 1.2. Problemática

A pesar de los avances en la tecnología de reconocimiento de imágenes, el diseño de un sistema de detección de señales de tráfico que funcione de manera confiable en diversas condiciones sigue siendo un desafío. Las señales de tráfico pueden verse afectadas por múltiples factores, como la variabilidad en la iluminación, las condiciones climáticas adversas, la obstrucción parcial por otros objetos y el desgaste físico. Además, las señales pueden variar significativamente en apariencia dependiendo de la región o país, lo que complica aún más la tarea de desarrollar un sistema universalmente eficaz. En el contexto específico de Cuba, la diversidad de señales y su estado de conservación presenta un desafío adicional para la implementación de estos sistemas.

### 1.3. Objetivos generales y específicos

**Objetivo General:** Desarrollar una metodología para la detección y clasificación de señales de tráfico cubanas utilizando técnicas avanzadas de reconocimiento de imágenes, además de proporcionar una base de datos de señales obtenidas en un entorno no controlado para la evaluación y mejora de los modelos de detección.

### **Objetivos Específicos:**

- Crear una base de datos: Recopilar y etiquetar imágenes de señales de tráfico cubanas en diferentes condiciones ambientales y de iluminación.
- Desarrollar un algoritmo de detección: Entrenar un modelo de aprendizaje automático que pueda detectar y clasificar señales de tráfico con alta precisión.
- Evaluar el desempeño del modelo: Probar el modelo en escenarios reales y comparar su rendimiento con métodos existentes.
- Mejorar la robustez del sistema: Investigar y aplicar técnicas para mejorar la resistencia del sistema a variaciones en la iluminación y las condiciones climáticas adversas.

## 2. Revisión bibliográfica

Dado que las señales de tráfico se clasifican por sus formas y colores, se pueden aplicar técnicas de aprendizaje de máquinas y visión computacional para su reconocimiento [6], aunque las más utilizadas son las redes convolucionales pues estas han demostrado poseer una gran precisión en la detección de señales. A. Hechri y A. Mtibaa [7] realizan una comparación entre dos aproximaciones distintas, una utilizando Maquinas de Soporte Vectorial (SVM por sus siglas en inglés) y la otra utilizando redes convolucionales, además proponen un nuevo modelo a dos fases, la primera de detección y reconocimiento, en la cual utilizan el SVM, y la segunda de clasificación, en la cual utiliza redes convolucionales para esta tarea. Se muestran resultados prometedores para este modelo, y un tanto desalentadores para el SVM.

Divya Elankumaran [3] propone un algoritmo que combina redes residuales (ResNet-50), YOLOv8 (You Only Look Once) y redes neuronales convolucionales profundas (DCNN) durante el entrenamiento para resolver esta tarea. En él, se utiliza YOLOv8 para la detección de las señales, mientras que de la clasificación se utilizan las DCNN, y para mitigar el efecto que pueda tener el trabajo con estas redes profundas se utiliza ResNet-50, dado que como posee capas residuales, permite que el gradiente salte ciertas capas durante el entrenamiento.

Gege Guo y Zhenyu Zhang [4] crean un algoritmo ligero basado en YOLOv5s para la detección de baches en las calles de manera más rápida, y aunque no detecte señales de tránsito, utiliza varias técnicas que son interesantes. Cambian la estructura de YOLOv5s, dado que modifican la columna (Backbone) del mismo utilizando una red MobileNetV3, la cual es una red, eficiente en términos de computación y memoria, lo cual se traduce en una mayor rapidez, aplican el algoritmo de KMeans para determinar automáticamente las mejores anclas (“anchor boxes” son uno de los hiperparámetros de YOLOv5); el suavizado de etiquetas y la reparametrización estructural. en el año 2022, Yanzhao Zhu y Wei Qi Yan [10] realizan una comparación entre los modelos de YOLOv5 y SSD (Single Shot MultiBox Detector), donde los resultados evidencian que YOLOv5 posee métricas mejores para el reconocimiento de señales.

En 2023 Weizhen Song and Shahrel Azmin Suani[9] proponen un nuevo modelo basado en YOLOv4-tiny y que utiliza una versión simplificada CSP-Darknet53 para la extracción de características, Agrupamiento de Pirámides Espaciales Mejorado (spatial-pyramid-pooling) lo cual mejora la capacidad del modelo para procesar objetos de diferentes tamaños; además de utilizar el algoritmo e K-Means para agrupar los datos y encontrar las mejores cajas anclas (anchor boxes).

Para concluir, los métodos basados en aprendizaje profundo, nos ofrecen amplios beneficios, dado su efectividad, precisión y rapidez para detectar y clasificar señales. En particular YOLO, es el que mejores resultados ofrece, además con la llegada de YOLOv8, como son la optimización de bloques residuales y cuellos de botella (*bottleneck blocks*) más eficientes, optimización de anclas con el algoritmo de *K-Means++* mejorando la precisión del modelo, se han mejorado las partes fundamentales de la red (*backbone*) y la cabeza de la red (*head*)

para una mejor extracción de características y detección de objetos; hace que YOLOv8 sea un buen candidato para utilizar en nuestro modelo.

### 3. Nuestro Modelo

Para una mejor comprensión de la solución, vamos dividir el capítulo en 6 temas fundamentales.

- **Línea de trabajo:** Se abordará una explicación general del enfoque de la solución, definiendo todos los procesos que se completaron en la misma.
- **Conjuntos de datos:** Explicaremos el proceso de obtención y procesamiento de los datos que se utilizaron para la solución.
- **Entrenamiento, validación y evaluación:** Abordaremos el proceso de entrenamiento de los diferentes modelos utilizados y se explicarán los resultados obtenidos de los modelos a partir de los datos de validación y evaluación.
- **Pruebas de campo:** Mostraremos resultados de los modelos en las pruebas de campo.

#### 3.1. Línea de trabajo

El enfoque principal consiste en clasificar las señales de tráfico en una de las siguientes cuatro categorías fundamentales:

- Prohibición
- Peligro
- Obligación
- Otra

Para lograr esto, hemos optado por el modelo **YOLOv8n**, dado que frente a las limitaciones de cómputo, es una alternativa ligera, además de su capacidad para mantener un equilibrio óptimo entre velocidad de procesamiento y precisión. Este modelo, previamente entrenado en un conjunto diverso de imágenes, será reentrenado específicamente para identificar y clasificar las señales de tráfico en las calles de Cuba, considerando sus particularidades locales y variabilidad en condiciones ambientales.

### **¿Por qué YOLO?**

YOLOv8 es la última versión de YOLO desarrollada por Ultralytics. YOLOv8 se basa en el éxito de las versiones anteriores, introduciendo nuevas características y mejoras para un rendimiento, flexibilidad y eficiencia mejorados. YOLOv8 admite una amplia gama de tareas de visión artificial, incluidas la detección, segmentación, estimación de poses, seguimiento y clasificación. Esta versatilidad permite a los usuarios aprovechar las capacidades de YOLOv8 en diversas aplicaciones y dominios.

Debido a limitaciones de tiempo en cuanto a la recolección de datos, decidimos utilizar un dataset derivado del (ya reconocido en el campo de Traffic Sign Recognition) German Traffic Sign Recognition Benchmark (GTSRB), con el fin de hacer un entrenamiento preliminar sobre el modelo base **YOLOv8n**. Además se confeccionó un conjunto de imágenes tomadas en La Habana, con el fin de utilizarlos para un entrenamiento mejor adaptado al entorno de la ciudad.

Buscamos acostumbrar un poco el modelo base en cuanto al reconocimiento de señales de tráfico. Para ello decidimos hacer *transfer learning* sobre el mismo, congelando las primeras capas del modelo convolucional. A partir de ese entrenamiento evaluamos sobre el propio conjunto de datos, y sobre el conjunto de datos CTSRD. Tras notar deficiencias en la detección de señales, realizamos un segundo entrenamiento sobre el conjunto de datos de señales tomadas en cuba, organizando los datos de dos formas distintas para lograr mejores resultados. Finalmente evaluamos los modelos obtenidos y hacemos pruebas de campo para visualizar los resultados.

## **3.2. Conjunto de datos**

### **3.2.1. Basado en German Traffic Sign Recognition Benchmark (GTSRB):**

El conjunto de datos almacenado en Kaggle, incluye imágenes junto a las anotaciones apropiadas en formato YOLO, que separa las imágenes en 4 clases, o en 43 casos. Nosotros optamos por el formato de 4 clases coincidentes con las 4 categorías nombradas anteriormente.

### **3.2.2. Cuban Traffic Sign Recognition Dataset (CTSRD)**

Para la creación del conjunto de datos, se registraron múltiples trayectos de vehículos a través de diversas calles de La Habana. A partir de estos registros, se extrajeron imágenes que contenían señales de tráfico. Las grabaciones se realizaron con una cámara Sony IMX 586 de 48 MP y sensor de 1/2", lo que garantiza una calidad en las imágenes obtenidas, suficiente para la tarea.

Debido a que las grabaciones capturan los trayectos del vehículo en diversas condiciones, se obtuvo una amplia variedad de señales de tráfico. Estas incluyen señales en condiciones de buena y mala iluminación, en diferentes momentos del día y bajo diversas condiciones climáticas (figura 1), proporcionando así un

alto grado de realismo a nuestro conjunto de datos. En total, se obtuvieron 614 imágenes con más de 1000 señales de tráfico en total.

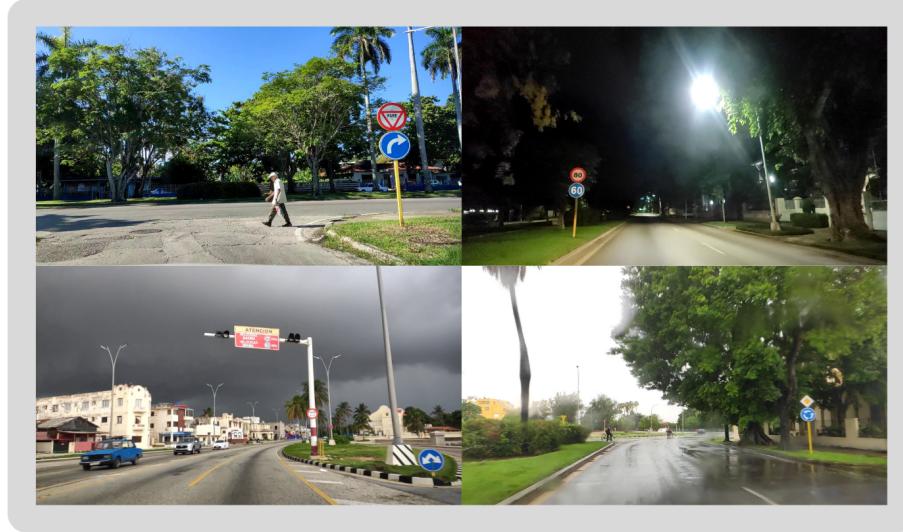


Figura 1: Distintos escenarios de las imágenes

Una vez obtenido el volumen de datos, comenzamos la segmentación de las señales en las imágenes. Para ello utilizamos una herramienta llamada Computer Vision Annotation Tool (CVAT) [2], que nos brinda una interfaz sencilla y eficiente para la anotación, además de tener un formato compatible con YOLO. Las imágenes se separaron en cuatro grupos esenciales, **Prohibición**, **Peligro**, **Obligación**, **Otras**. Del total de imágenes se obtuvieron 1011 señales, 70 de peligro, 398 de prohibición, 87 de obligación y 456 de otros tipos de señales, como se muestra en la imagen 2



Figura 2: Señales separadas por grupos

### ¿Por qué CVAT?

CVAT.ai es una herramienta versátil para la anotación de imágenes y videos, que sirve a la comunidad de *Computer Vision* en todo el mundo. Su objetivo es ayudar a desarrolladores, empresas y organizaciones a nivel global mediante un enfoque de inteligencia artificial centrado en los datos. CVAT Cloud ofrece una plataforma robusta para la colaboración en proyectos de anotación de imágenes y videos, permitiendo a los equipos trabajar juntos de manera eficiente y efectiva, facilitando la colaboración entre diferentes miembros y departamentos.

Tras tener todas las imágenes segmentadas, debemos separar los datos en los conjuntos de entrenamiento, validación y prueba. Consideramos dos formas de separar los datos:

- Aleatoriamente utilizando el 20 % para el conjunto del prueba, y del conjunto restante, el 20 % para validación y el 80 % para entrenamiento.
- Con una idea de separar proporcionalmente los datos, basado en señales que comparten cierta similitud, con el objetivo de hacer la repartición más equitativa para el modelo. Separación que además cumple los porcentajes del tipo de repartición anterior. Por lo que se realizó un conteo de las señales separándolas por dicha similitud, como se aprecia en la figura 3



Figura 3: Cantidad de señales separadas por similitud

En siguiente capítulo que trata sobre el entrenamiento de los modelos se explican las razones y usos de dichas separaciones.

### 3.3. Entrenamiento y resultados del modelo

Como ya abordamos anteriormente, partimos desde el modelo base **YOLOv8n (nano)**. Este modelo es el más ligero de la familia YOLOv8 y consta de 24 capas y 3.2 millones de parámetros. YOLOv8n ha demostrado un comportamiento al nivel del estado del arte en conjuntos de datos populares como COCO y VOC.

YOLO es una excelente opción para transfer learning debido a su arquitectura pre-entrenada que sobresale en la extracción de características. Esto permite que, al aplicarse *Transfer Learning*, el modelo ya cuente con una sólida base de

características previamente aprendidas, lo que acelera y mejora el proceso de entrenamiento en nuevos dominios o con datos específicos.

En la primera etapa de la solución entrenamos el modelo en el conjunto de datos GTSRB, con el objetivo de que aprenda a reconocer algunas señales. Para ello decidimos hacer *Transfer Learning* definiendo las primeras 5 épocas como *épocas de calentamiento* y manteniendo invariables los parámetros de las 10 primeras capas, a las que se le conoce como el *backbone* de YOLOv8n, las cuales se encargan de la extracción de características. Este modelo basado en GTSRB se guarda en el archivo *general\_best.pt*

Una vez entrenado nuestro modelo inicial, evaluamos sus resultados con ambos conjuntos de datos (CTSRD y GTSRB)(figura 4).

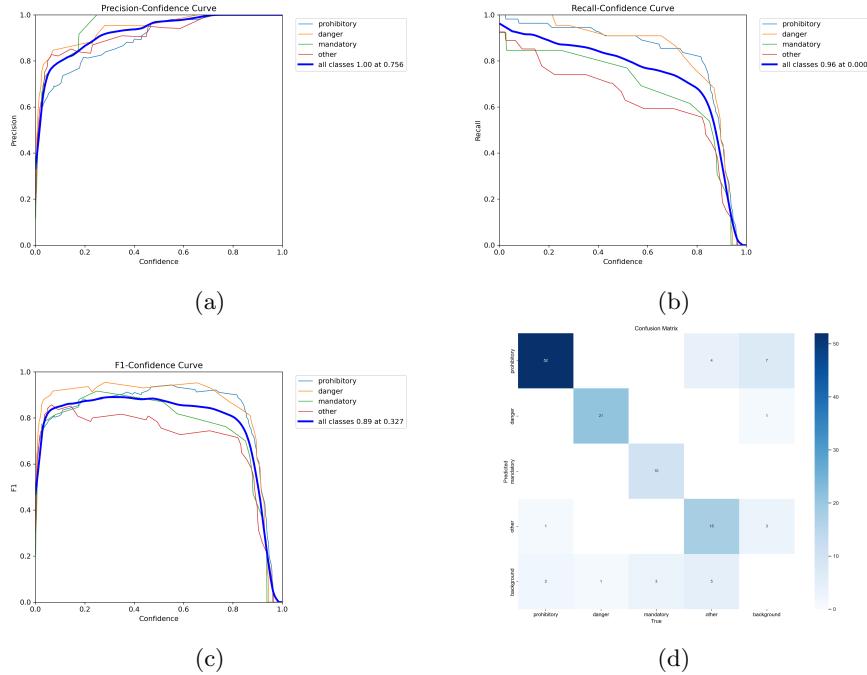


Figura 4: Métricas en el conjunto GTSRB. a) Curva de precisión b) Curva de recobrado c) F1 d) Matriz de confusión

Como se puede apreciar en los gráficos anteriores, podemos concluir que el modelo posee un buen desempeño en las imágenes de prueba del conjunto de imágenes de la misma naturaleza del conjunto donde fue entrenado. Se puede observar como para niveles no muy exigentes de confianza, el modelo obtiene una excelente precisión, además en la gráfica de recobrado obtenemos buenos valores para poca confianza y la pendiente de descenso no es tan pronunciada inicialmente. Además, si analizamos la matriz de confusión, se puede observar que el mayor problema que presenta, es que reconoce señales donde no debe haberlas y reconoce señales donde no las hay. Si analizamos la gráfica de la

métrica F1 se puede apreciar que para los valores centrales de confianza tanto la precisión como el recobrado son bastante altos y estables, solo obteniendo un descenso brusco para valores muy altos (por encima de 0.8) de confianza.

De forma general observamos buenas métricas en la evaluación anterior, sin embargo ¿Qué pasa si evaluamos en los datos de CTSRD?

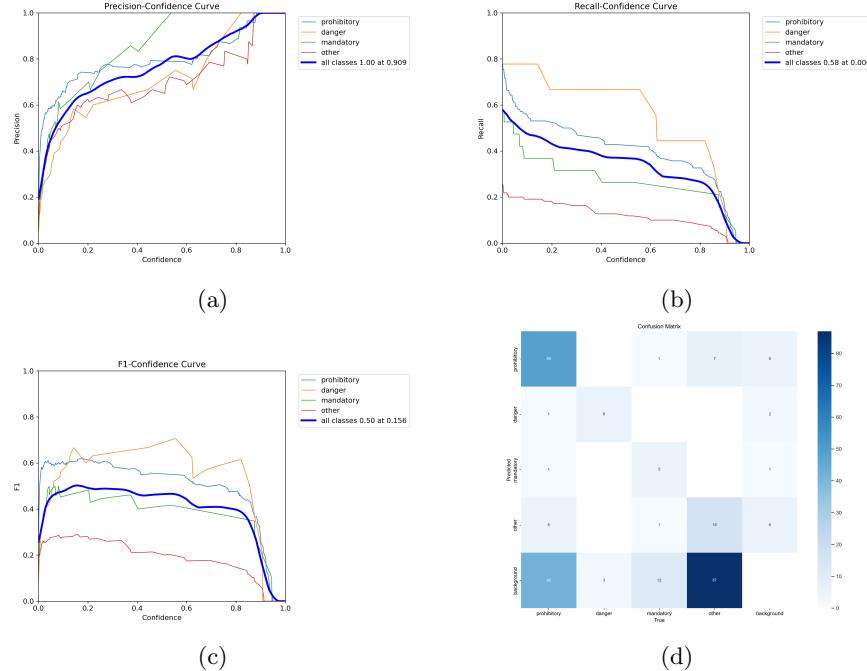


Figura 5: Métricas en el conjunto GTSRB. a) Curva de precisión b) Curva de recobrado c) F1 d) Matriz de confusión

Al observar los resultados (figura 5) en este conjunto de datos, es evidente el mal desempeño del modelo. Una de las razones de este mal desempeño es la heterogeneidad de las características que describen a las señales de ambos conjuntos figura, por no mencionar los metadatos de las imágenes. Lo anterior expuesto nos lleva a la necesidad de hacer un nuevo entrenamiento sobre el modelo *general\_best.pt* con los datos del conjunto CTSRD.



Figura 6: Ejemplos de pare en ambos conjuntos

Para enfrentar esta problemática propusimos dos soluciones de entrenamiento diferenciada por la forma de repartir los datos, explicadas en la sección anterior. La división aleatoria fue la primera aproximación ingenua (modelo *cuban\_rand.pt*), por otra parte, la segunda separación llega como una alternativa más equilibrada para el modelo, en aras de dotar de heterogeneidad en los conjuntos de entrenamiento, validación y prueba (modelo *cuban\_sim.pt*). Comparemos los resultados de ambos modelos

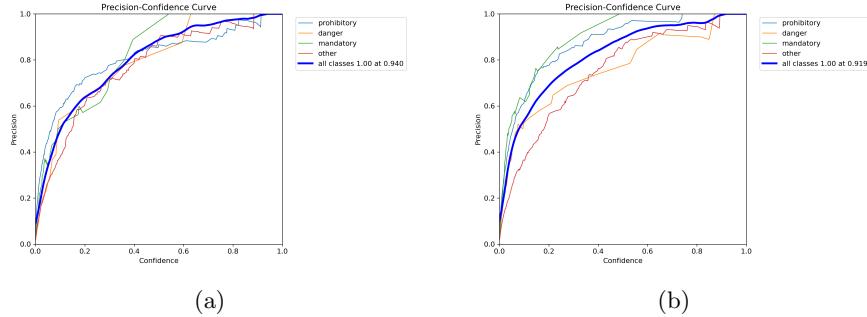


Figura 7: Métricas en el conjunto GTSRB. a) Precisión en el conjunto aleatorio  
b) Precisión en el conjunto ordenado

En las gráficas anteriores (figura 7) observamos una leve mejora en la precisión de las clases **Prohibición** y **Obligación**, además de que se suaviza la curva promedio lo cual sugiere una mayor generalización del modelo.

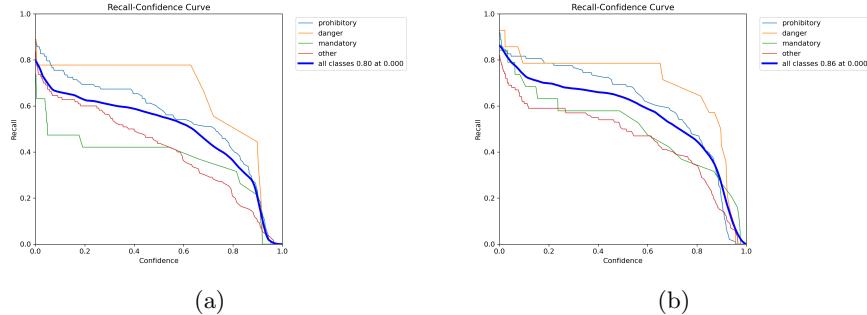


Figura 8: Métricas en el conjunto GTSRB. a) Recobrado en el conjunto aleatorio  
b) Recobrado en el conjunto ordenado

En las gráficas de recobrado (figura 8), se aprecia una mejora en todas las clases. Esto significa que, después de reorganizar los datos de entrenamiento y validación, el segundo modelo muestra una mayor capacidad para identificar correctamente los casos positivos de cada clase. En otras palabras, el segundo modelo es más efectivo en términos de recobrado, ya que es capaz de encontrar y clasificar correctamente una mayor proporción de verdaderos positivos

para todas las categorías evaluadas. Esta mejora en el recobrado sugiere que la nueva organización de los datos ha permitido al modelo aprender patrones más representativos y generalizables, lo cual se refleja en un mejor rendimiento en la detección de todas las clases consideradas.

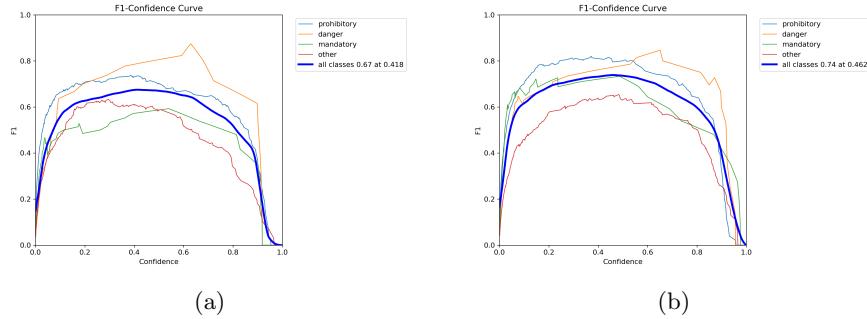


Figura 9: Métricas en el conjunto GTSRB. a) F1 en el conjunto aleatorio b) F1 en el conjunto ordenado

La mejora en las curvas de F1 (figura 9) implica que la reorganización de los datos ha permitido al segundo modelo obtener una representación más efectiva y generalizable de los datos, resultando en un rendimiento superior y más robusto en la clasificación de todas las clases evaluadas.

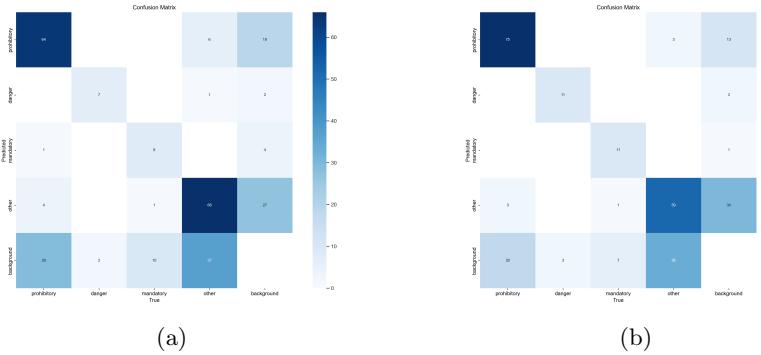


Figura 10: Métricas en el conjunto GTSRB. a) Matriz de confusión del conjunto aleatorio b) Matriz de confusión del conjunto ordenado

Como se puede apreciar en la matriz de confusión (figura 10), el modelo organizado fue capaz de clasificar e identificar mejor cada señal, llevandolo a generalizar mejor.

Considerando que el modelo descrito reconoce únicamente las cuatro clases de señales de tráfico mencionadas anteriormente, su implementación se sugiere como un complemento para aumentar la atención de conductores inexpertos en la vía. Este sistema puede funcionar eficazmente como una alerta preventiva,

especialmente en el caso de señales de prohibición. La tendencia del modelo a detectar señales con mayor frecuencia que a omitirlas, debido a su alta precisión, contribuye a mantener a los conductores en un estado de alerta constante. Esta característica es particularmente beneficiosa para prevenir accidentes, ya que asegura que los conductores sean informados oportunamente sobre la presencia de señales críticas, permitiéndoles tomar las acciones necesarias para evitar situaciones de peligro.

### 3.4. Pruebas de campo

Como parte de las pruebas de campo, hemos acoplado una cámara a un vehículo de forma que grabe el recorrido del mismo por las calles. Tras un análisis estadístico de la distribución de las señales de tránsito en la vía, en diferentes regiones, consideramos óptima la localización de la cámara, a  $5/8$  de proporción horizontal, a la altura de la base del parabrisas (figura 11).

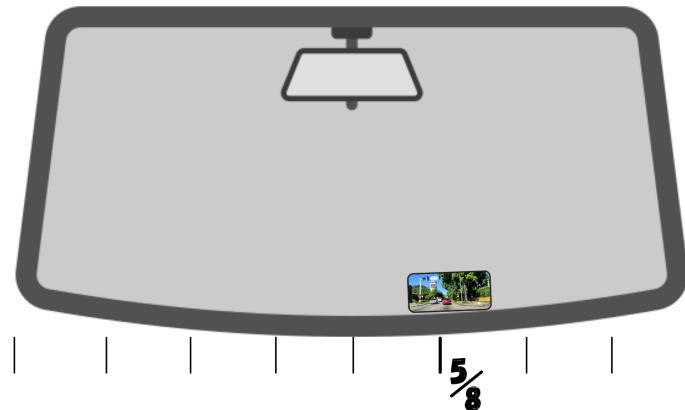


Figura 11: Localización de la cámara

La razón de la elección de esta proporción se basa en que la mayoría de las señales de tránsito en carretera se encuentran cercanas al borde derecho de la calle, sin descuidar las que se encuentran en el lado izquierdo; y en menor cantidad en la parte superior de la calle, junto a semáforos.

Dada la necesidad de obtener resultados certeros en las predicciones de las señales, hemos utilizado un valor de confianza del 70 %. Vale aclarar que como el porpósito final de este modelo es en grabaciones en tiempo real, la cantidad de capturas en poco tiempo es alta, por lo que existe una mayor probabilidad de detectar una señal que existe, si en una captura falla.

Algunas grabaciones obtenidas han sido postprocesadas por nuestro modelo entrenado, obteniéndose los videos que hubiesen sido detectados en tiempo real.

Con el objetivo de observar mejor el resultado de lo que sería una prueba de campo, hemos provisto el archivo *webcam.py*, que consiste en un script de

python que utiliza la webcam integrada en la computadora donde se ejecuta, para obtener predicciones de nuestro modelo en tiempo real.

La frecuencia de capturas depende del procesador de la computadora donde se ejecuta el script.

## 4. Conclusiones

El análisis exhaustivo de señales cubanas mediante técnicas avanzadas de visión computacional revela hallazgos significativos para su aplicación práctica. Los modelos de aprendizaje profundo han demostrado una notable eficacia en la clasificación automática de estas señales, a pesar de desafíos como variaciones en condiciones ambientales y estructurales. Este estudio subraya la viabilidad de implementar sistemas basados en visión computacional para mejorar la seguridad vial y la gestión del tráfico en entornos urbanos cubanos. Sin embargo, se identifica la necesidad urgente de bases de datos anotadas y accesibles, así como la consideración ética en el uso de esta tecnología en espacios públicos. Estos resultados no solo promueven avances en la infraestructura tecnológica local, sino que también abren nuevas perspectivas para la innovación en sistemas de transporte inteligente y conducción autónoma en Cuba.

Se notó, que las imágenes anotadas como **Otras** son muy variadas en cuanto a sus características visuales, por lo que nuestro modelo no desempeña el mejor papel en la identificación de estas. Una posible solución hubiese sido aumentar los datos anotados con esta clasificación. Otra alternativa pudiese ser dividir esta categoría en las 5 que aparecen en la *Ley 109 del código de seguridad vial*[1] asegurando mayores similitudes en su aspecto.

Inicialmente en los objetivos del proyecto estaba hacer un análisis del daño de las señales del tránsito, para así, complementar lo anterior expuesto y proveer un acercamiento al estudio del estado de las señales del tránsito en Cuba. A continuación se ofrecerán una aproximación a posibles soluciones:

- Una solución general hubiese sido, basándonos en aprendizaje supervisado, tener un conjunto de datos con señales dañadas y en buen estado. Esto implicaría la construcción de un nuevo conjunto de datos lo cual es inviable por cuestiones de tiempo y recursos.
- Una posible aproximación también sería la propuesta por Vanessa Dallborgo et al. [8], ya que en Cuba también hay un problema con señales ocultas por vegetación, para ello, se agregaría una nueva clase al modelo, lo que implicaría una segmentación del conjunto de datos en su totalidad, además de requerir más datos de esta clase que se quiere explotar.
- Por último Bubryur Kim et al. [5] proponen un algoritmo para la detección de daño en estructuras de acero, como puentes y vigas de acero, pero desde una distancia más próxima a las mismas. Se podría utilizar esta propuesta

para detectar el daño en las señales de tráfico, por lo anterior expuesto no se puso en practica esta solución

## 5. Glosario

- **Anchor Boxes:** Las anchor boxes son cajas predefinidas de ciertas alturas y anchuras que se utilizan como referencias para predecir las cajas delimitadoras reales alrededor de los objetos.
- **Backbone:** Se refiere a la red de extracción de características que procesa los datos de entrada en una cierta representación de características. Constituido por las primeras capas del modelo.
- **Bloques Residuales:** Su idea principal es introducir conexiones de atajo (skip connections) que permiten el flujo de información sin interrupciones a través de la red, saltando una o más capas
- **Bottleneck blocks:** Tipo de bloque de convolución que se utiliza para reducir la dimensionalidad de las características intermedias dentro de una red neuronal profunda, y luego expandirlas nuevamente
- **Bounding Box:** En la detección de objetos, una bounding box es una caja rectangular que se utiliza para definir la posición y el tamaño del objeto en una imagen.
- **Épocas de calentamiento:** Técnica utilizada durante el entrenamiento de redes neuronales profundas para suavizar la transición hacia un aprendizaje más agresivo.
- **Función SoftMax:** La función softmax, o función exponencial normalizada, se emplea para comprimir un vector K-dimensional de valores reales arbitrarios en un vector K-dimensional, de valores reales en el rango [0, 1].
- **Head:** Parte final de la red que se encarga de realizar la predicción de las salidas deseadas, como las clases de los objetos y sus ubicaciones
- **K-Means++:** Es una mejora del algoritmo KMeans estándar. Mejora la inicialización de los centroides para que el algoritmo converja más rápido y alcance una mejor solución final en términos de calidad del clustering.
- **Transfer Learning:** En lugar de entrenar un modelo desde cero, se aprovecha el conocimiento adquirido por un modelo existente para mejorar la eficiencia y el rendimiento de un nuevo modelo.

## Referencias

- [1] *Ley 109, Código de Seguridad Vial.* Editorial Capitán San Luis, 2011.
- [2] CVAT.ai Corporation. Computer vision annotation tool, 2024.
- [3] Divya Elankumaran. An improved traffic sign recognition and road lane detection for self-driving cars using yolo-v8. *International Journal For Multidisciplinary Research*, 2023.
- [4] Gege Guo and Zhenyu Zhang. Road damage detection algorithm for improved yolov5. *Scientific Reports*, 2022.
- [5] Bubryur Kim; N. Yuvaraj; Hee Won Park; K. Preethaa; R. Pandian; Dong-Eun Lee. Investigation of steel frame damage based on computer vision and deep learning. *Automation in Construction*, 2021.
- [6] Tejal Patil; Sonali Patil; Pawan Nirpal; Govind Madankar; Prof. B Mahalakshmi. A survey on traffic sign recognition techniques. 2022.
- [7] A. Hechri; A. Mtibaa. Two-stage traffic sign detection and recognition based on svm and convolutional neural networks. *IET Image Processing*, 2020.
- [8] Vanessa Dalbongo; Thiago B. Murari; Vinicius S. Madureira; João Gabriel L. Moraes; Vitor Magno O. S. Bezerra; Filipe Q. Santos; Alexandre Silva; and Roberto L. S. Monteiro. Traffic sign recognition with deep learning: Vegetation occlusion detection in brazilian environments. *Sensors*, 2023.
- [9] Weizhen Song and Shahrel Azmin Suand. Tsr-yolo: A chinese traffic sign recognition algorithm for intelligent vehicles in complex scenes. *Italian National Conference on Sensors*, 2023.
- [10] Yanzhao Zhu and Wei Qi Yan. Traffic sign recognition based on deep learning. *Multimedia tools and applications*, 2022.