

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería



**Diseño e Implementación de un Sistema de Asistencia de  
Tránsito a través de modelos de Redes Neuronales y Visión  
por Computador**

Trabajo de graduación presentado por José Ricardo Méndez González  
para optar al grado académico de Licenciado en Ingeniería en Ciencias  
de la Computación y Tecnologías de la Información

Guatemala,

2025







UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería



**Diseño e Implementación de un Sistema de Asistencia de  
Tránsito a través de modelos de Redes Neuronales y Visión  
por Computador**

Trabajo de graduación presentado por José Ricardo Méndez González  
para optar al grado académico de Licenciado en Ingeniería en Ciencias  
de la Computación y Tecnologías de la Información

Guatemala,

2025



Vo.Bo.:

(f)   
MSc, MBA Luis Alberto Suriano Saravia

Tribunal Examinador:

(f)   
MSc, MBA Luis Alberto Suriano Saravia

(f) \_\_\_\_\_  
Alexander Bolaños

Fecha de aprobación: Guatemala, 12 de noviembre de 2025.



---

## Prefacio

---

En los últimos años las estadísticas de siniestros vehiculares han ido en aumento. Han llegado un punto donde reflejan un peligro diario al ciudadano común. El objetivo de este trabajo es poder brindar una herramienta que sirva para poder enfocar de mejor manera el cumplimiento de la nortamiva, con el fin de disminuir estas desafortunadas estadísticas.

Este trabajo se lo dedico a Dios, que me ha brindado la oportunidad de formarme en esta universidad y vivir hasta el día en que redacto este texto.

Quiero agradecer a mi familia, especialmente a mis padres, tía y hermana: Ricardo Méndez Tello, Lilian Magaly González, Rosa Celia González y Lilian Nohemy Méndez, quienes me apoyaron en todo momento y han sido pilares fundamentales en mi formación personal. A mis amigos de etapas previas y a aquellos que la universidad puso en mi camino, quienes fueron una constante fuente de inspiración y apoyo durante estos años de formación, especialmente en esta etapa final.



---

## Índice

---

<b>Prefacio</b>	<b>v</b>
<b>Lista de figuras</b>	<b>ix</b>
<b>Resumen</b>	<b>xI</b>
<b>Abstract</b>	<b>xIII</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Justificación</b>	<b>3</b>
2.1. Justificación . . . . .	3
<b>3. Objetivos</b>	<b>5</b>
3.1. Objetivo general . . . . .	5
3.2. Objetivos específicos . . . . .	5
<b>4. Marco teórico</b>	<b>7</b>
4.1. Visión por computadora y aprendizaje profundo . . . . .	7
4.1.1. Historia y evolución . . . . .	7
4.2. Infracciones de tránsito y normativas . . . . .	13
4.2.1. Reglamento de tránsito y sanciones . . . . .	13
4.3. Aplicaciones y estado del arte . . . . .	14
4.3.1. Sistemas actuales de detección de infracciones . . . . .	14
4.3.2. IA y ciudades inteligentes . . . . .	15
4.4. Aspectos técnicos relevantes . . . . .	16
4.4.1. Preprocesamiento de imágenes . . . . .	16
4.4.2. Métricas de desempeño en detección y clasificación . . . . .	19
4.4.3. Desafíos de implementación en tiempo real . . . . .	19
4.5. Consideraciones éticas y sociales . . . . .	19
4.5.1. Impacto de la IA en la seguridad vial . . . . .	20
4.5.2. Privacidad y uso de cámaras en espacios públicos . . . . .	20
4.5.3. Limitaciones y riesgos en detección automática . . . . .	20

<b>5. Metodología</b>	<b>23</b>
5.1. Metodología general . . . . .	23
5.1.1. Análisis de datos disponibles y selección de artículos . . . . .	23
5.1.2. Búsqueda y construcción de dataset propio . . . . .	23
5.1.3. Anotación de datos . . . . .	24
5.1.4. Entrenamiento de redes YOLO . . . . .	24
5.1.5. Implementación del algoritmo SORT . . . . .	27
5.1.6. Evaluación del sistema . . . . .	27
<b>6. Resultados</b>	<b>29</b>
6.1. Resultados . . . . .	29
6.1.1. Vehículos . . . . .	29
6.1.2. Señales de Tránsito . . . . .	31
6.1.3. Semáforos . . . . .	32
6.1.4. Sistema completo . . . . .	34
<b>7. Discusión de resultados</b>	<b>37</b>
7.1. Discusión de resultados . . . . .	37
<b>8. Conclusiones</b>	<b>39</b>
8.1. Conclusiones . . . . .	39
<b>9. Recomendaciones</b>	<b>41</b>
9.1. Recomendaciones generales . . . . .	41
9.1.1. Recomendaciones complementarias . . . . .	41
<b>10. Bibliografía</b>	<b>43</b>

---

## Lista de figuras

---

1.	Arquitectura de LeNet-5 . . . . .	9
2.	Arquitectura de GoogleNet . . . . .	10
3.	Arquitectura de VGG . . . . .	10
4.	Arquitectura de ResNet50 . . . . .	11
5.	Arquitectura de Region-based CNN . . . . .	11
6.	Arquitectura de You Only Look Once . . . . .	12
7.	Ejemplos de detección y clasificación de vehículos . . . . .	30
8.	Métricas de YOLO en detección y clasificación de automóviles. . . . .	30
9.	Ejemplos de detección y clasificación de señales de tránsito. . . . .	31
10.	Métricas de YOLO en detección y clasificación de señales de tránsito. . . . .	32
11.	Ejemplos de detección y clasificación de semáforos. . . . .	33
12.	Métricas de YOLO en detección y clasificación de semáforos. . . . .	33
13.	Ejemplo del sistema completo en funcionamiento para Semáforos en Rojo. . .	34
14.	Ejemplo del sistema completo en funcionamiento para señal de Prohibido el Giro en U. . . . .	34
15.	Ejemplo del sistema completo en funcionamiento para señal de Alto. . . . .	35



---

## Resumen

---

El presente trabajo describe el diseño e implementación de un sistema de asistencia de tránsito basado en redes neuronales y visión por computador. El objetivo principal fue desarrollar un agente capaz de detectar vehículos, semáforos y señales de tránsito en imágenes, videos y transmisiones en tiempo real, con el fin de identificar infracciones viales de forma automática. Para ello, se emplearon modelos de detección de objetos de la familia YOLOv8 y el algoritmo DeepSORT, los cuales permiten combinar reconocimiento y seguimiento de múltiples elementos en entornos urbanos dinámicos.

El sistema fue entrenado utilizando conjuntos de datos públicos como UA-DETRAC y LISA, además de un dataset propio de señales de tránsito guatemaltecas. Se aplicaron técnicas de aumento de datos y ajuste de hiperparámetros para optimizar la precisión y evitar sobreajuste. La validación se realizó en un entorno virtual de alta fidelidad generado con Grand Theft Auto V, que permitió simular infracciones como el irrespeto a la luz roja, la señal de alto y el giro prohibido en U.

Los resultados demuestran un desempeño aceptable y estable en tiempo real, con detecciones precisas y una adecuada capacidad de generalización en diferentes condiciones de iluminación y fondo. El sistema constituye una base sólida para futuras aplicaciones en monitoreo vial, automatización de reportes de infracciones y desarrollo de soluciones orientadas a ciudades inteligentes.

Palabras clave: Visión por computador, YOLOv8, DeepSORT, detección de infracciones, inteligencia artificial, tránsito vehicular.



---

## Abstract

---

This work presents the design and implementation of a traffic assistance system based on neural networks and computer vision. The main objective was to develop an intelligent agent capable of detecting vehicles, traffic lights, and road signs in images, videos, and real-time streams to automatically identify traffic violations. The system integrates YOLOv8 object detection models with the DeepSORT tracking algorithm, enabling simultaneous recognition and tracking of multiple elements in complex urban environments.

Public datasets such as UA-DETRAC and LISA were used for training, complemented by a custom dataset of Guatemalan traffic signs. Data augmentation and hyperparameter optimization techniques were applied to enhance accuracy and reduce overfitting. Validation was performed using the Grand Theft Auto V virtual environment, which allowed the controlled simulation of violations such as running a red light, ignoring a stop sign, and making illegal U-turns.

Results show that the system operates efficiently in real time, achieving high detection precision and stability under various lighting and background conditions. The developed model provides a strong foundation for future implementations in traffic monitoring, automatic infraction reporting, and smart city applications.

**Keywords:** Computer vision, YOLOv8, DeepSORT, traffic violation detection, artificial intelligence, intelligent transportation.



# CAPÍTULO 1

---

## Introducción

---

El desarrollo de sistemas automatizados basados en visión por computadora ha adquirido un papel central en la modernización de la gestión del tránsito y la seguridad vial. La capacidad de los modelos actuales para interpretar escenas complejas, detectar objetos relevantes y analizar comportamientos en tiempo real ha sido posible gracias a avances significativos en aprendizaje profundo, técnicas de clasificación y algoritmos de seguimiento. Desde los primeros descubrimientos sobre el procesamiento visual en el cerebro hasta la consolidación de arquitecturas como LeNet, AlexNet, VGG, GoogleNet, ResNet y YOLO, la evolución del campo ha permitido trasladar principios biológicos y computacionales hacia aplicaciones prácticas de alto impacto.

En el contexto guatemalteco, el control del tránsito depende en gran medida de la observación directa del personal autorizado, lo cual limita la cobertura, la consistencia y la capacidad de respuesta ante infracciones recurrentes. La normativa vigente describe con detalle las sanciones aplicables, pero su cumplimiento operativo requiere mecanismos de vigilancia más robustos, capaces de operar continuamente y reducir la intervención manual. Esta necesidad contrasta con soluciones internacionales que integran análisis visual automatizado para optimizar la circulación vehicular, identificar zonas de riesgo y reducir incidentes derivados de infracciones.

Los trabajos recientes en detección de infracciones mediante inteligencia artificial muestran que la combinación de modelos de detección de objetos y algoritmos de seguimiento multi-objeto constituye un enfoque eficaz para monitorear intersecciones y vías urbanas. Investigaciones que emplean YOLO, SORT, DeepSORT o StrongSORT evidencian que es posible detectar violaciones a semáforos, señales y normas operativas con precisión significativa, incluso en entornos urbanos complejos. Estas propuestas sirven como referencia directa para el presente proyecto, guiando la selección de arquitecturas, la construcción de datasets, el diseño del flujo de procesamiento y la definición de criterios de evaluación.

El objetivo central de este trabajo fue diseñar e implementar un sistema capaz de detectar vehículos, señales de tránsito y semáforos, dar seguimiento a los objetos relevantes e identificar automáticamente infracciones específicas establecidas en el reglamento nacional.

Para ello se integraron técnicas de preprocesamiento de imágenes, construcción y anotación de conjuntos de datos, entrenamiento de modelos YOLOv8 para detección y clasificación, y aplicación del algoritmo DeepSORT para seguimiento persistente. La validación del sistema se realizó mediante videos generados en un entorno sintético de alta fidelidad, lo que permitió evaluar su funcionamiento bajo escenarios controlados.

Asimismo, se consideran los desafíos éticos y sociales asociados al uso de cámaras y algoritmos de visión en espacios públicos, así como los riesgos derivados de errores de detección, sesgos en los datos o interpretaciones incorrectas de la señalización vial. El proyecto incorpora estas consideraciones con el fin de proponer un sistema técnicamente viable, pero también responsable y alineado con principios de protección de datos y equidad.

En conjunto, este trabajo busca aportar una base técnica y metodológica para el desarrollo de sistemas automatizados de análisis del tránsito, evaluando su desempeño, sus limitaciones y su potencial para integrarse en soluciones reales de control y monitoreo vial. El análisis de resultados y las recomendaciones presentadas establecen un marco orientado a futuras mejoras, incluyendo la incorporación de datos capturados en entornos reales y la optimización del sistema para despliegues permanentes.

# CAPÍTULO 2

---

## Justificación

---

### 2.1. Justificación

En Guatemala, los accidentes de tránsito constituyen un problema creciente para la seguridad vial y la movilidad urbana. De acuerdo con datos del Instituto Nacional de Estadística (INE), durante el año 2023 se registraron **11,198 víctimas de accidentes de tránsito**, lo que representa un **incremento del 4.4 %** en comparación con el año anterior. De estas víctimas, el **20.3 % fallecieron** y el **79.7 % resultaron lesionadas**, evidenciando un impacto significativo tanto en la salud pública como en la eficiencia del sistema de transporte.[1]

Este incremento en la siniestralidad resalta la necesidad de implementar métodos más eficientes de monitoreo y asistencia vial, capaces de detectar infracciones y comportamientos de riesgo de manera automática. Un sistema basado en visión por computador y modelos de inteligencia artificial permitiría identificar infracciones en tiempo real o de forma asincrónica, contribuyendo a la reducción de accidentes y a la mejora de la seguridad en las carreteras.

Asimismo, el proyecto se alinea con las iniciativas tecnológicas actuales de la Municipalidad de Guatemala, que buscan modernizar la gestión del tránsito mediante la implementación del *plan de Ciudades Inteligentes*. Este programa contempla la sustitución de los semáforos convencionales en más de 500 intersecciones por dispositivos inteligentes conectados a plataformas como *Waze for Cities*. En este contexto, el desarrollo de un sistema automatizado de detección de infracciones representa una herramienta complementaria y de alto valor, que podría integrarse a la infraestructura existente para fortalecer la toma de decisiones y la seguridad vial en entornos urbanos.[2]



# CAPÍTULO 3

---

## Objetivos

---

### 3.1. Objetivo general

Desarrollar un agente de inteligencia artificial capaz de procesar imágenes y vídeo de modo que permita conocer si algún vehículo involucrado ha cometido alguna infracción de tránsito.

### 3.2. Objetivos específicos

- Detectar vehículos, vehículos pesados y motocicletas en imágenes, metrajes asíncronos o en tiempo real.
- Identificar semáforos y señales de tránsito en imágenes, metrajes asíncronos o en tiempo real.
- Desarrollar un sistema que analice los resultados por medio de visión por computador dando un veredicto sobre las infracciones cometidas.



# CAPÍTULO 4

---

## Marco teórico

---

### 4.1. Visión por computadora y aprendizaje profundo

#### 4.1.1. Historia y evolución

En esta sección se presenta de forma detallada la historia de la visión por computadora y parte de inteligencia artificial con aprendizaje profundo.

Además, se profundiza sobre la evolución de los distintos modelos del ámbito, resaltando características importantes sobre los modelos seleccionados para llevar a cabo el presente trabajo de graduación.

#### Descubrimientos en los campos de biología y fisiología

Los primeros experimentos y descubrimientos del campo se dieron en los finales de la década de los 50. Varios experimentos realizados, principalmente por los campos de medicina y fisiología, dieron las bases a lo que hoy es el procesamiento visual artificial. Estos experimentos se centraron en conocer y entender cómo el sistema neuronal de los seres vivos captaba y procesaba la información visual de su entorno.[3]

En 1959 se realizaron y presentaron los resultados de un experimento donde colocaron electrodos en la corteza estriada de un gato. El objetivo era capturar las señales eléctricas causadas por la estimulación visual. El estudio demostró la existencia de campos receptivos.[4] Algunos reaccionan a la intensidad y posición de la luz, mientras que otros al movimiento horizontal y vertical por separado. También se detectó una mejor reacción a estímulos en movimiento en vez de estímulos estacionarios.[3]

En seres con visión binocular (es decir, con dos ojos) las regiones receptivas fueron ubicadas de la misma manera o muy similar tanto en tamaño, forma, orientación y con

las mismas respuestas ante estímulos. Además, el *efecto de sumación*<sup>1</sup> se vio incluso en situaciones donde en cada ojo se buscaba estimular una región diferente.[5]

Otro experimento importante fueron los avances de Kirsch sobre algoritmos de digitalización en torno al escáner. En 1957 publicó un estudio donde detalla experimentos de computadoras que fueron usadas para procesar información visual. El escáner fue un aporte vital para los objetivos de este estudio. Este era un dispositivo capaz de digitalizar información de una imagen. El sistema funcionaba a través de índices de luz. La imagen era rotada varias veces por un tambor, y el escáner detectaba pequeñas áreas de 0.25<sup>2</sup>mm (el área de un píxel). Si el índice de luz reflejado por el área era menor a un límite ajustado, el sistema le brindaba un valor de 1. Si se escaneaba suficiente espacio blanco o el índice reflejado era mayor al límite, se le daba un valor de 0. Se mencionan también avances sobre la detección de bordes e identificación de objetos.[6]

Más adelante, luego del descubrimiento de los campos receptivos, David Marr realizó un experimento centrado en el movimiento visual. Introdujo un *framework*<sup>2</sup> que él nombró *State-Motion-State*. Esta estructura redefine un movimiento complejo al descomponerlo en un grupo de movimientos elementales sencillos de analizar. El trabajo de Marr fue fundamental, pues no solo sentó los principios fisiológicos, sino que también introdujo principios matemáticos y computacionales.[7]

## Reconocimiento de patrones y reconocimiento óptico de caracteres

Una vez fueron publicados los avances sobre digitalización de imágenes, Kirsch tuvo una nueva idea. Desarrolló un sistema capaz de interpretar texto en inglés y patrones en imágenes. A pesar de que el texto fue altamente teórico, varios lenguajes de programación dedicados al procesamiento de imágenes comenzaron a surgir. Durante la época, el objetivo principal era el desarrollo del reconocimiento óptico de caracteres (OCR por sus siglas en inglés) debido a su significado en la industria; las computadoras serían capaces de "leer". Esto significaría que los procesos tardados, como el registro de datos de documentos o tarjetas perforadas, podrían automatizarse.[8]

Los primeros modelos de máquinas que realizaban el proceso de OCR datan desde 1929 en Alemania y 1933 en Estados Unidos. En ambos casos, el objetivo principal era poder leer una tarjeta perforada y transmitir la información obtenida. Para ello se utilizaba una técnica de identificación llamada *mask matching*. Se refiere a cuando se usa una plantilla fija para la comparación y reconocimiento de caracteres. Era un acercamiento útil y fácil de implementar en imprenta, pero tenía varias fallas cuando se empleaba en escritos a mano. En este caso existen infinitas 'fuentes' y se debería tener una plantilla para cada una. Este era un gran problema y, para solucionarlo, surgieron los métodos de análisis estructural (por ejemplo, el análisis de trazos). Funcionaban de ambas formas, solos o como complemento.

Existen cuatro generaciones de OCR donde la primera es la implementación de *mask*

<sup>1</sup>El efecto de sumación en percepción visual se refiere a la integración de estímulos múltiples que generan una respuesta neuronal mayor que la obtenida con cada estímulo por separado. Este fenómeno se observa en la corteza visual primaria (V1).

<sup>2</sup>Un framework es un conjunto estandarizado de conceptos, prácticas y criterios que sirve como referencia para abordar y resolver problemas dentro de un dominio específico.[7]

*matching*. Luego se tiene el primer acercamiento al reconocimiento de escritos a mano y su posterior mejora. Finalmente, está la extracción y entendimiento de formas más complejas, como documentos que contienen tablas y símbolos.

Durante este período, el procesamiento de imágenes también tuvo avances significativos gracias a la disponibilidad de computadoras. Procedimientos tales como segmentación y detección de objetos comenzaron a tener sus primeros acercamientos, bastante similares a lo que hoy en día realizan las redes neuronales. Desde este momento se hacía notar el costo computacional del procesamiento de imágenes y se teorizaba sobre la paralelización de sus tareas para fines de convolución.[9]

## Desarrollos tempranos de Machine y Deep Learning

El primer trabajo centrado en visión por computadora utilizando técnicas de ML es la tesis doctoral de Lawrence Roberts. Roberts desarrolló un método capaz de obtener las propiedades de un objeto 3D a partir de una serie de entradas en 2D. También incluyó programas capaces de transformar de forma bidireccional entre 2D y 3D a través de manipulaciones en matrices computacionales.[10]

En 1980 se presenta el modelo de redes neuronales denominado como *neocognitron*. Fukushima lo describe en su estudio como una estructura de múltiples capas capaz de reconocer patrones visuales independientemente de su posición espacial.[11]

## Algoritmos de clasificación

El modelo de clasificación de imágenes *LeNet-5*[12] fue publicado en 1998, sentando las bases para el uso de redes neuronales convolucionales (CNN por sus siglas en inglés) en tareas de clasificación. Este estudio introdujo el concepto de *backpropagation*<sup>3</sup> y demostró avances en el uso de OCR para el procesamiento de documentos.

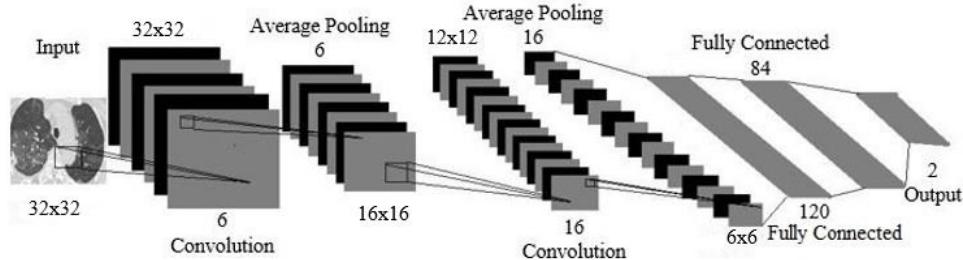


Figura 1: Arquitectura de LeNet-5  
[14]

*Alexnet* fue presentado en 2012 durante un desafío de visión por computadoras propuesto y organizado por Microsoft.[15] Siendo una versión mejorada de LeNet-5 capaz de soportar

---

<sup>3</sup> Backpropagation es un algoritmo de entrenamiento en redes neuronales que ajusta los pesos calculando el error de salida y propagándolo hacia atrás a través de la red, usando derivadas para actualizar los parámetros y minimizar la función de pérdida.[13]

60 millones de parámetros y entrenada con tarjeta gráfica o de video (GPU), demostró la importancia de arquitecturas CNN aplicadas a la visión por computador.

En el año 2014 son presentadas dos nuevas arquitecturas diseñadas para la clasificación de imágenes, *Visual Geometry Group* (VGG por sus siglas) y *GoogleNet*. El estudio de VGG demostró que, al aumentar la profundidad de una CNN, su desempeño incrementa de manera significativa. Sin embargo, el costo es muy alto, pues la complejidad y ocupación en memoria se elevan.[16] GoogleNet usó los resultados de este estudio para buscar una estructura capaz de ser profunda a un bajo costo. Los resultados fueron satisfactorios, obteniendo una estructura de 22 capas (a comparación de las 18 de VGG) a un bajo costo.[17]

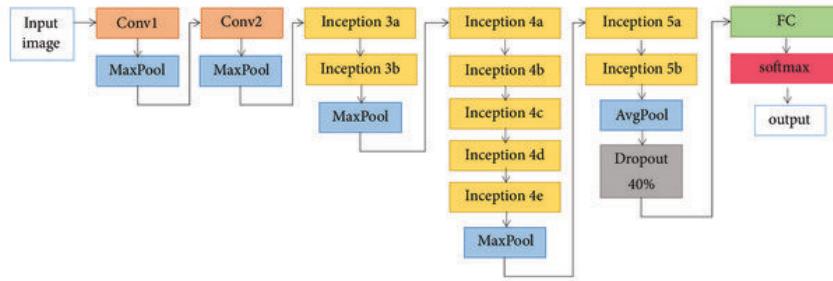


Figura 2: Arquitectura de GoogleNet  
[18]

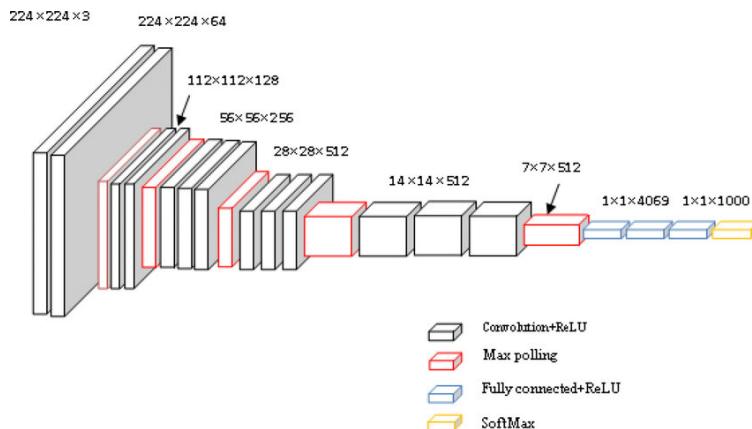


Figura 3: Arquitectura de VGG  
[19]

*ResNet* fue presentada en 2015. Producto del estudio que se planteó la pregunta: *is learning better networks as easy as stacking up layers?*[20] Durante su estudio detallan que existe un problema de pérdida de precisión mientras más aumenta la profundidad de una CNN. Para solventar el problema, se propuso el enfoque de *deep residual learning*. En esta solución, en lugar de aproximar toda la transformación desde cero, el agente toma la identidad y le añade el residuo o "lo que falte" para que el valor se approxime a la transformación deseada.

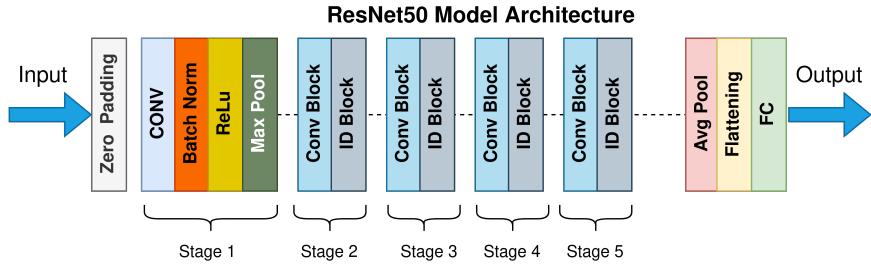


Figura 4: Arquitectura de ResNet50  
[21]

## Algoritmos de detección

Como OCR tiene su propio foco independiente, poco a poco fue surgiendo la investigación de modelos enfocados en otro tipo de objetos de la vida real. Uno de los primeros estudios, el de Viola y Jones, introdujo un algoritmo para el fin de la detección de objetos. Su framework les permitía la detección de objetos en tiempo real y con gran precisión. Su arquitectura se basa en el uso de características tipo Haar calculadas eficientemente mediante la imagen integral, un proceso de selección y combinación de clasificadores débiles a través de AdaBoost, y una cascada de clasificadores que descarta rápidamente regiones no relevantes.[22]

En 2014 fue publicada la arquitectura Region-based CNN. Opuesta a la arquitectura de Viola y Jones, esta empleaba redes neuronales para el fin de detección.[3]

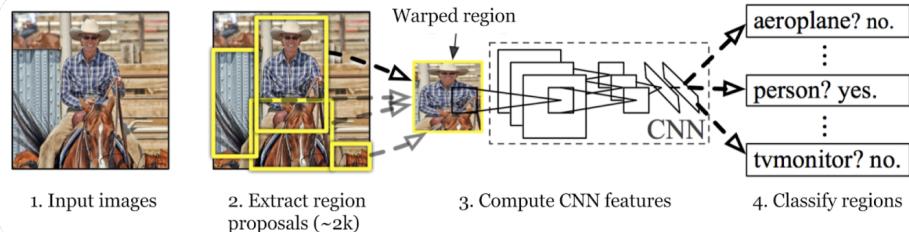


Figura 5: Arquitectura de Region-based CNN  
[23]

El modelo *You Only Look Once* (YOLO por sus siglas) es presentado en 2016. Su estructura consiste en una entrada de imagen en formato de matriz  $S \times S$ . Cada celda predice un número fijo de cajas de contorno, sus confianzas y clase. Finalmente, se combinan las predicciones para localizar y clasificar objetos en un solo paso por la red. La clave principal de YOLO es que no tiene la necesidad de separar la detección de la región y la clasificación, usando la misma etapa para ambas. Esto le otorga una gran velocidad y un estándar de precisión razonable para varios fines, tales como el análisis de imágenes u objetos en tiempo real.[24]

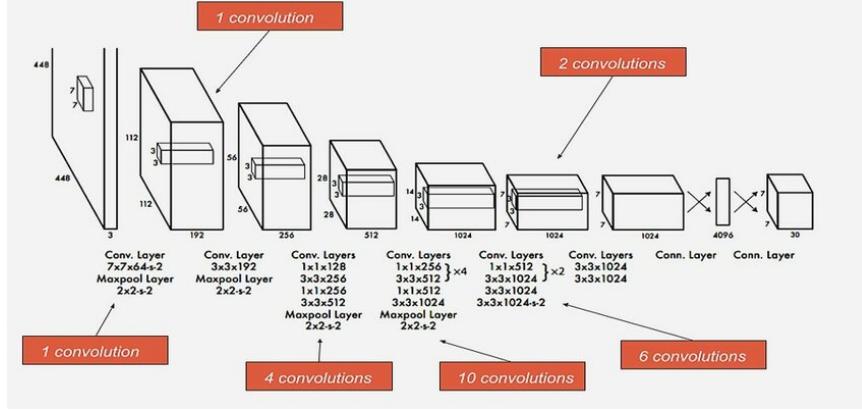


Figura 6: Arquitectura de You Only Look Once [25]

## Algoritmos de seguimiento

El algoritmo *Simple Online and Realtime Tracking* (SORT) fue propuesto con el objetivo de ofrecer un método de seguimiento de objetos eficiente, simple y en tiempo real. SORT se basa en el uso del filtro de Kalman para predecir la posición de los objetos entre cuadros consecutivos y del algoritmo de asignación de Hungría para asociar las nuevas detecciones con las trayectorias previamente estimadas. Gracias a su bajo costo computacional, este método logró una excelente relación entre precisión y velocidad, lo que permitió su integración con detectores de objetos como YOLO o Faster R-CNN.[26]

Posteriormente, en 2017 se introdujo *DeepSORT*, una extensión del algoritmo original que incorporó técnicas de aprendizaje profundo para mejorar la asociación de objetos a lo largo del tiempo. DeepSORT añadió un modelo de extracción de características visuales (*Re-Identification* o ReID) basado en redes neuronales convolucionales, lo cual permitió mantener la identidad de los objetos incluso ante occlusiones parciales o cambios en la apariencia visual. Esta mejora incrementó significativamente la robustez del sistema en entornos complejos, especialmente en el seguimiento de peatones y vehículos.[27]

En 2022, se presenta *StrongSORT*, una versión mejorada de DeepSORT que optimizó tanto el modelo de ReID como los mecanismos de asociación de trayectorias. StrongSORT incorporó redes de características más avanzadas, como OSNet, además de módulos adicionales como el filtro de movimiento adaptativo (*Motion Compensation Module*) y el ajuste de asociación por tiempo (*Time-aware Update*). Estas modificaciones aumentaron la precisión del seguimiento en escenarios con alta densidad de objetos y occlusiones prolongadas.[28]

Actualmente, los algoritmos derivados de la familia SORT continúan siendo ampliamente utilizados como base para sistemas de seguimiento en tiempo real, debido a su equilibrio entre velocidad, precisión y facilidad de implementación. Plataformas como Ultralytics incorporan variantes modernas de esta familia y las presentan como opciones listas para su uso en tareas de seguimiento, entre ellas *OC-SORT*, *ByteTrack* y *BoT-SORT*. Estas implementaciones integran estrategias avanzadas de asociación y detección múltiple que mejoran la estabilidad del seguimiento en entornos dinámicos y elevan el rendimiento de los sistemas basados en detección y seguimiento.[29]

## **4.2. Infracciones de tránsito y normativas**

En esta sección se detalla con claridad cómo se encuentra distribuido el reglamento de tránsito. El fin en mente es transmitir la estructura y definir los artículos que fueron considerados para el presente trabajo.

### **4.2.1. Reglamento de tránsito y sanciones**

En Guatemala, el reglamento de tránsito se encuentra en la *Cartilla de Infracciones y Sanciones de Tránsito*, con su última actualización realizada en 2020 y distribuida por el Departamento de Tránsito de la Policía Nacional Civil. El documento se organiza en las siguientes secciones:

#### **Verificación básica**

Describe la revisión inicial que puede realizar el personal autorizado sobre cualquier vehículo (automóvil o motocicleta). Consta de ocho directrices, destacando la prohibición de circulación de vehículos con timón en el lado derecho.[30]

#### **Boleta de aviso**

Contiene recomendaciones para el llenado de la boleta de aviso, así como el procedimiento para requerimiento de pago o citación. Incluye diez buenas prácticas a seguir.[30]

#### **Artículos y sanciones**

Detalla los artículos 180 a 186, indicando las posibles infracciones y sus respectivas sanciones. Las multas pueden variar entre Q 100 y Q 25,000, dependiendo del artículo infringido y la gravedad de la infracción.[30]

El foco del trabajo serán los siguientes artículos:

- Artículo 180. Multa de Q. 100.00.
- 5) Por no utilizar las señales de tránsito correspondientes al virar o girar, cambiar de sentido, cambiar de carril, desacelerar y retroceder.<sup>4</sup>
- 14) Por no respetar las señales de tránsito siguientes:
  - A. Alto
  - B. Alto, del semáforo

---

<sup>4</sup>Este inciso fue seleccionado como parte de un grupo de incisos con el fin de proporcionar normativa a la señal “Prohibido girar en U”, debido a que su norma es ambigua o no definida. Por este motivo el inciso no será aplicado en su totalidad.

- Artículo 181. Multa de Q. 200.00.
- 15) Por circular en contra de la vía señalizada o autorizada.<sup>4</sup>
- 20) )Por virar o girar sin observar las normas de posicionamiento y maniobra reglamentarias.<sup>4</sup>

## 4.3. Aplicaciones y estado del arte

### 4.3.1. Sistemas actuales de detección de infracciones

En esta sección se presentan algunos de los trabajos más actuales en el ámbito. El fin en mente de la misma es poder notar similitudes y diferencias entre el presente y antiguos sistemas de detección de infracciones de tránsito. Además, se analiza cómo cada sistema puede impactar o guiar el desarrollo del presente trabajo de graduación, ya sea en la selección de algoritmos, diseño de arquitectura o definición de métricas de validación.

#### Vigilancia del tráfico de última generación

Un trabajo que propone un acercamiento con base en inteligencia artificial y visión por computadora como respuesta a los accidentes de tráfico. Plantean una arquitectura que utiliza *YOLOv5* para la detección de vehículos, peatones, semáforos y señales de tránsito. Luego, para garantizar el correcto seguimiento de la trayectoria del vehículo, se implementó *strongSORT*<sup>5</sup>. El modelo final mostró resultados satisfactorios, siendo capaz de detectar hasta seis infracciones comunes distintas, siendo estas[31]:

- Violaciones a la señal del semáforo en rojo
- Uso indebido del carril de emergencia
- Mal manejo de la distancia de seguridad entre vehículos
- Obstrucción del paso peatonal
- Aparcamiento en lugares indebidos
- Aparcamiento obstruyendo el paso peatonal

Este trabajo es particularmente relevante para el presente proyecto de graduación porque demuestra la efectividad de combinar detección de objetos y seguimiento multi-objeto en entornos urbanos complejos. Su arquitectura y elección de algoritmos orienta decisiones sobre qué técnicas usar para detectar infracciones de tránsito y cómo manejar la oclusión o pérdida de objetos en secuencias de video.

---

<sup>5</sup> StrongSORT es una mejora de DeepSORT que refuerza el seguimiento multi-objeto en escenas complejas mediante varias innovaciones: incorpora un modelo de movimiento optimizado en el filtro de Kalman, un módulo de compensación de movimiento de cámara y un módulo de interpolación de trayectorias que reduce pérdidas por oclusión.[28]

## Road-Eye

Un trabajo que detalla un acercamiento que combina CNNs y visión por computadora para la detección de ciertas infracciones de tránsito. Este sistema presenta un acercamiento único, ya que el foco del trabajo fue la falta de seguridad de peatones y pilotos. Para ello, su sistema se enfoca en reconocer y detectar placas de vehículo, pasos peatonales y pilotos de motocicleta sin casco. El fin en mente es poder detectar situaciones de riesgo, pilotos que violan la norma de seguridad y la posibilidad de vincular infracciones con el propietario.[32]

La influencia de este sistema en el trabajo de graduación radica en su enfoque en riesgos específicos y la vinculación de infracciones con propietarios. Permite considerar la inclusión de métricas de seguridad y trazabilidad de infracciones, además de reforzar la importancia de segmentar correctamente los objetos de interés en entornos con múltiples actores.

### 4.3.2. IA y ciudades inteligentes

Esta sección analiza proyectos oficiales de implementación de sistemas inteligentes en ciudades, con enfoque en seguridad vial y mejoras de tránsito.

#### Proyecto Green Light

En 2023, Google anuncia el *Proyecto Green Light*. El objetivo principal es atacar directamente el problema de las emisiones de vehículos, el cambio climático y problemas de tráfico detenido en distintas ciudades. Para ello se disponen a analizar distintas intersecciones en las ciudades voluntarias.[33] El proceso consta de cuatro fases:

- Entendimiento de la intersección

El equipo revisa la intersección usando su mapa desarrollado durante décadas. Esto les permite inferir distintos parámetros de los semáforos como el tiempo por ciclo, tiempo de transición, coordinación con otros semáforos, funcionalidad de sensores, entre otros.

- Mediciones y tendencias del tráfico

El equipo diseña un modelo para simular y entender como fluye el tráfico en dicha intersección. Les permite entender y analizar patrones de tráfico como tiempos de espera, coordinación con intersecciones adyacentes, cambios en los semáforos durante el día y otros.

- Desarrollo de recomendaciones para aliviar el problema

Usando un modelo de inteligencia artificial, el equipo identifica las potenciales mejoras que alivian el mayor porcentaje del tráfico. Se ponen en contacto con la ciudad para transmitirles estas recomendaciones y la ciudad debe considerar su aplicación. Un gran beneficio para la ciudad es que el equipo Green Light evita recomendar cambios en infraestructura. Sus recomendaciones siempre se encuentran enfocadas en el sistema de semáforos y señales de tránsito existentes. Es decir, sus recomendaciones son optimizaciones en los tiempos de cada semáforo durante el día, un ajuste que no significa ningún costo para la ciudad.

- Análisis del impacto de las recomendaciones

Una vez el equipo tiene confirmación de la ciudad sobre los cambios aplicados, miden cuántas paradas han evitado para los pilotos durante el día y su aporte en el patrón de tráfico. Luego usan los estándares de la industria para realizar cálculos sobre cuántas emisiones evitaron, traduciéndolo en impacto al cambio climático.

## Iniciativa SMART MODES

La ciudad de Detroit, en 2025, presentó una iniciativa orientada a mejorar la seguridad en intersecciones de alto riesgo mediante el uso de datos en tiempo real y la integración de soluciones tecnológicas. El proyecto abarca actualmente 19 intersecciones estratégicas. A través de Smart Modes, el sistema unifica y procesa información procedente de sensores y cámaras para obtener una caracterización continua del entorno vial, incluyendo patrones de tráfico, velocidades, presencia de obstáculos y factores de riesgo derivados de infracciones recurrentes.

Smart Modes funciona como una capa de análisis que combina distintas fuentes de información para generar una representación coherente del estado de cada intersección. Aunque los detalles específicos de su funcionamiento no son públicos, el sistema provee un conjunto de capacidades que permiten interpretar condiciones del entorno urbano en tiempo real. Entre estas se incluyen la detección de cambios relevantes en el flujo vehicular, la identificación de maniobras peligrosas y el reconocimiento de situaciones que podrían derivar en incidentes. Esta integración permite que la información obtenida sea más útil que la generada por sensores individuales operando de manera aislada.

Otra característica importante es la capacidad del sistema para priorizar zonas de riesgo. Al monitorear simultáneamente varias intersecciones, Smart Modes identifica patrones que evidencian comportamientos peligrosos o condiciones anómalas, como incrementos súbitos en la velocidad promedio, congestión inesperada o repetición de infracciones similares en la misma área. Esta priorización facilita la toma de decisiones por parte de los operadores urbanos, quienes pueden dirigir recursos, ajustar la programación semafórica o intervenir físicamente en puntos críticos.[34]

## 4.4. Aspectos técnicos relevantes

### 4.4.1. Preprocesamiento de imágenes

En esta sección se exploran algunas técnicas para el preprocesamiento de imágenes, siendo el foco aquellas necesarias para los modelos utilizados en el presente trabajo.

#### Redimensionamiento

Existen dos técnicas comúnmente usadas: *Bilinear Interpolation* y *Nearest Neighbor*.

- Bilinear Interpolation

Toma los valores de los píxeles adyacentes al objetivo y usa su promedio. El resultado visual de este proceso es satisfactorio en la mayoría de casos pero tiene un costo computacional significativo.

- Nearest Neighbor

Toma el valor más cercano al píxel objetivo. Es un proceso mucho más rápido y barato. Su desventaja es que visualmente es fácil distinguir los bloques creados por el algoritmo ya que carece de transiciones entre colores.[35]

## Normalización

Existen dos técnicas comúnmente usadas: *Min-Max Scaling* y *Z-Score*.

- Min-Max Scaling

Escala los píxeles en valores de 0 y 1. Es fácilmente interpretable, mantiene la distribución original y es útil en la mayoría de casos. Su desventaja es que es sensible a valores atípicos, ya que puede mantener a los datos comunes en valores muy bajos.

- Z-Score

Escala los píxeles usando el valor de la media y su desviación estándar. La idea es mantener a los valores comunes cercanos a 0 y los valores atípicos cercanos a 1. Este enfoque es menos sensible a los valores atípicos. Su desventaja es que la escala resultante puede ser menos intuitiva y no mantiene un rango fijo.[35]

## Consideraciones para datasets

Una vez las imágenes fueron preprocesadas, es momento de manejar el dataset. Esto suele significar la división del dataset. La norma es utilizar 80 % para entrenamiento y 20 % para pruebas si no se planea tener una etapa de validación. En caso de que se busque tener una etapa de validación, la distribución común es 70 % para entrenamiento, 10 % para pruebas y 20 % para validación. YOLO es la excepción a la norma, ya que su arquitectura no suele tener etapa de "test" sino únicamente "train" y "val", usualmente aplicando el 80 % - 20 %.

Con el dataset dividido, es necesario realizar un análisis de los datos. Esto significa revisar si las clases se encuentran correctamente balanceadas. Los porcentajes pueden variar según los tipos de datos y el fin del modelo. En caso de que exista una clase dominante, se pueden considerar las técnicas *undersampling*, la cual trata de reducir las muestras de dicha clase u *oversampling*, la cual trata de aumentar las muestras de la clase minoritaria.[35]

## Data Augmentation

Se denomina *Data Augmentation* al proceso de incrementar de manera artificial la cantidad y diversidad de muestras disponibles en un conjunto de datos. En el caso de la visión por computador, esto se logra aplicando transformaciones controladas sobre las imágenes

originales, tales como variaciones en el color, brillo, saturación, iluminación, orientación, escala, rotación, traslación, desenfoque o incluso la introducción de ruido digital.

El propósito principal de esta técnica es mejorar la capacidad de generalización del modelo al exponerlo a versiones alteradas de las mismas imágenes, simulando condiciones más variadas de captura del entorno. De esta forma, el modelo aprende a reconocer los objetos de interés sin depender excesivamente de condiciones específicas como la iluminación, el ángulo de la cámara o la resolución del video.

Además, el uso de esta técnica reduce el riesgo de sobreajuste (*overfitting*), especialmente en conjuntos de datos limitados, ya que obliga al modelo a adaptarse a una representación más amplia del espacio de posibles entradas. En contextos de tráfico vehicular, estas técnicas resultan particularmente útiles, pues las condiciones ambientales pueden cambiar drásticamente y afectar la apariencia de los objetos detectados.

En síntesis, *Data Augmentation* constituye una etapa esencial en el preprocesamiento de datos para modelos de *Deep Learning*, al aumentar la diversidad del conjunto de entrenamiento y fortalecer la robustez del sistema frente a escenarios reales.[35]

#### **4.4.2. Métricas de desempeño en detección y clasificación**

La evaluación de modelos de visión por computadora requiere métricas específicas que permitan cuantificar la precisión y confiabilidad de las predicciones. Para tareas de detección de objetos con YOLO, las métricas más utilizadas incluyen Intersection over Union (IoU), que mide el solapamiento entre las predicciones y las anotaciones reales; mean Average Precision (mAP), que resume la precisión promedio de detección a través de diferentes umbrales de IoU; y las tasas de precisión y recall, que indican la proporción de detecciones correctas respecto al total de predicciones y al total de objetos reales, respectivamente. Estas métricas permiten evaluar tanto la capacidad del modelo para localizar objetos como para minimizar falsos positivos y falsos negativos.

Para tareas de clasificación de imágenes con ResNet, las métricas de desempeño típicas incluyen precisión, recall, F1-score y accuracy, que reflejan la habilidad del modelo para asignar correctamente las imágenes a sus categorías correspondientes. La combinación de estas métricas proporciona un panorama completo del rendimiento del modelo, permitiendo comparar diferentes configuraciones y ajustes de hiperparámetros, así como evaluar su robustez frente a clases desbalanceadas o condiciones de iluminación y ocultación variables. [36]

#### **4.4.3. Desafíos de implementación en tiempo real**

La implementación de modelos de detección en tiempo real presenta múltiples desafíos que van más allá de alcanzar una alta precisión. Uno de los principales retos es mantener un equilibrio entre velocidad de inferencia y rendimiento, dado que los modelos más profundos suelen ofrecer mayor exactitud a costa de una latencia inaceptable en aplicaciones críticas. Según Wang y sus compañeros, este compromiso se hace evidente en entornos donde cada milisegundo resulta crucial, como la conducción autónoma o la videovigilancia. Además, la limitación de recursos en dispositivos embebidos obliga a diseñar arquitecturas ligeras y optimizadas, capaces de ejecutar de manera eficiente sin comprometer la robustez del modelo. Además, factores externos como el ancho de banda y la variabilidad en la calidad de los datos en tiempo real imponen restricciones adicionales que deben considerarse en el despliegue práctico.[37]

### **4.5. Consideraciones éticas y sociales**

El desarrollo de sistemas de visión por computadora basados en inteligencia artificial para el tránsito no sólo plantea retos técnicos, sino también dilemas éticos y sociales. Estas tecnologías impactan directamente en la seguridad vial, la privacidad de los ciudadanos y en la toma de decisiones automatizadas que pueden tener consecuencias significativas. A continuación, se analizan tres dimensiones clave.[38]

#### **4.5.1. Impacto de la IA en la seguridad vial**

El uso de algoritmos de IA en sistemas avanzados de asistencia al conductor (ADAS) y en vehículos autónomos promete reducir accidentes ocasionados por errores humanos, que constituyen más del 90 % de los siniestros viales según la Organización Mundial de la Salud (OMS, 2018). La capacidad de los modelos de visión artificial para detectar peatones, ciclistas y obstáculos en tiempo real mejora la toma de decisiones y la anticipación de riesgos en entornos urbanos complejos.

Sin embargo, la implementación de estas tecnologías no elimina completamente los accidentes. Casos documentados, como los fallos de sistemas de conducción autónoma en pruebas en Estados Unidos, evidencian que aún existen limitaciones técnicas y que el despliegue prematuro puede generar una falsa percepción de seguridad. Por lo tanto, es esencial garantizar fases de prueba extensivas, supervisión humana y regulaciones claras que prioricen la protección de la vida humana.

#### **4.5.2. Privacidad y uso de cámaras en espacios públicos**

La adopción de cámaras en semáforos, intersecciones y vehículos conectados plantea preocupaciones sobre la privacidad de los ciudadanos. Estos sistemas recopilan grandes volúmenes de datos visuales que pueden contener información personal identificable, como rostros y placas vehiculares. El Reglamento General de Protección de Datos (GDPR) de la Unión Europea establece que el tratamiento de imágenes debe cumplir con principios de proporcionalidad y minimización de datos.

En América Latina, el despliegue de sistemas de videovigilancia para el tránsito se ha acelerado, pero muchas veces sin marcos regulatorios sólidos que garanticen transparencia y uso legítimo de la información. Esto abre la discusión sobre quién controla los datos, cómo se almacenan, por cuánto tiempo y con qué fines adicionales podrían utilizarse (por ejemplo, vigilancia policial, control migratorio o publicidad dirigida). El reto ético consiste en equilibrar la seguridad vial con la protección de las libertades civiles.

El presente proyecto no recopila, almacena ni transforma ningún tipo de información personal identificable. La propuesta desarrollada se limita a ejecutar el proceso descrito y a proporcionar el resultado correspondiente. La responsabilidad sobre el manejo de la información recae exclusivamente en el usuario que utilice el sistema.

#### **4.5.3. Limitaciones y riesgos en detección automática**

Aunque los modelos de visión por computadora han alcanzado altos niveles de precisión, todavía presentan sesgos y limitaciones. Estudios recientes demuestran que el rendimiento de los algoritmos puede disminuir bajo condiciones climáticas adversas, en escenarios nocturnos o en presencia de objetos atípicos. Asimismo, existen evidencias de que los sesgos en los datos de entrenamiento afectan el reconocimiento de peatones de distintos grupos demográficos, lo cual podría derivar en desigualdades en la seguridad vial.

Otro riesgo ético relevante es la atribución de responsabilidad en caso de error. Si un

sistema automatizado falla al detectar la correcta señalización o la jerarquía de las mismas en un punto específico, es posible que ciudadanos se vean afectados con infracciones injustas. Esto puede ocurrir en situaciones donde un mismo punto de tránsito cuenta con múltiples señales que podrían entrar en conflicto o tener prioridades distintas. Por ejemplo, si en una intersección el semáforo está en verde pero también existe una señal de ALTO, un sistema que no interprete correctamente la jerarquía de señales podría registrar a todos los conductores que avanzan como infractores, generando sanciones injustas. Este tipo de errores no solo afecta la percepción de justicia de los usuarios, sino que también plantea dudas sobre la responsabilidad legal: ¿debe responder el conductor, el operador del sistema, o la entidad que implementó la tecnología? La correcta resolución de estas situaciones requiere un diseño que considere la complejidad del entorno vial y mecanismos claros de apelación y revisión de infracciones.



# CAPÍTULO 5

---

## Metodología

---

### 5.1. Metodología general

El presente proyecto tuvo como objetivo principal el diseño e implementación de un sistema de visión por computadora capaz de detectar y dar seguimiento a señales de tránsito en distintos entornos viales, con el propósito de evaluar su aplicabilidad en escenarios reales. Para alcanzar este objetivo, se llevó a cabo una investigación preliminar sobre bibliografía y trabajos previos en el área de detección y clasificación de objetos, así como una revisión de las arquitecturas más utilizadas en el estado del arte, entre ellas las familias de redes YOLO y ResNet.

#### 5.1.1. Análisis de datos disponibles y selección de artículos

En la etapa inicial se realizó una revisión bibliográfica y técnica de investigaciones previas en el campo de la visión por computadora aplicada al tráfico y la detección de señales. El objetivo fue identificar enfoques metodológicos relevantes, métricas de evaluación y técnicas de preprocesamiento o entrenamiento que pudieran servir como referencia. Asimismo, se examinaron diversos conjuntos de datos públicos con el fin de evaluar su calidad, variedad y aplicabilidad al problema planteado. Finalmente, se seleccionaron los conjuntos *UA-DETRAC* y *LISA* para abordar la detección de vehículos y semáforos, respectivamente. En cuanto a las señales de tránsito, se optó por construir un conjunto de datos propio.[39][40]

#### 5.1.2. Búsqueda y construcción de dataset propio

Dado que los conjuntos de datos existentes no cubrían completamente las necesidades del proyecto, se procedió a construir un dataset propio enfocado en señales de tránsito de Guatemala. Este proceso implicó recopilar imágenes representativas de distintas situaciones

de tráfico a partir de diversas fuentes, como *Google Images*, *Google Street View* y capturas fotográficas propias.

Debido a la limitada cantidad de datos disponibles, fue necesario aplicar técnicas de *Data Augmentation*. Estas consisten en modificar imágenes existentes (mediante variaciones de brillo, rotación, escala, entre otros) para generar nuevas muestras. De esta forma se incrementa artificialmente el tamaño del conjunto de datos, reduciendo la probabilidad de sobreajuste (*overfitting*).

Posteriormente, se realizó el proceso de anotación de datos, descrito en la siguiente sección.

### 5.1.3. Anotación de datos

La anotación de datos constituyó una fase crítica para el entrenamiento supervisado de los modelos. Este proceso consiste en delimitar manualmente las regiones de las imágenes o de los fotogramas de video donde se encuentran los objetos de interés; en este caso, las señales de tránsito. Aunque existen diversos formatos para representar las anotaciones, en este proyecto se empleó el formato de YOLO, que utiliza una línea de texto por objeto con información normalizada según la siguiente estructura:

Identificador de clase Centro en X Centro en Y Ancho Alto

El procedimiento se realizó mediante la herramienta *labelImg*, desarrollada en Python.[41] Esta aplicación ofrece una interfaz gráfica que permite trazar y visualizar de inmediato las cajas delimitadoras sobre cada imagen.

Cabe mencionar que la herramienta carece de soporte oficial y presenta ciertos fallos que debieron ser corregidos manualmente durante la fase de anotación. Por este motivo, su uso no es recomendable sin experiencia previa en depuración y resolución de errores en Python, dado que puede requerir ajustes técnicos adicionales.

### 5.1.4. Entrenamiento de redes YOLO

Con el dataset preparado, se entrenó un primer modelo basado en la arquitectura YOLOv8 de *Ultralytics*, diseñada para la detección en tiempo real de objetos. Se definieron los siguientes hiperparámetros iniciales:

VEHICLES

```
learning\_rate = 0.001
batch\_size = 8
epochs = 15
workers = 4
```

```

image\_size = 512
optimizer = "SGD"

LIGHTS

learning\_rate = 0.001
batch\_size = 8
epochs = 50
workers = 4
image\_size = 512
optimizer = "SGD"

```

La selección de estos parámetros respondió al objetivo de aprovechar al máximo la capacidad del equipo empleado en el entrenamiento. Sus especificaciones fueron: Ryzen 5 8645H, 8 GB de RAM y 4 GB de VRAM. Se priorizó el uso del optimizador “SGD” por su rapidez y estabilidad, permitiendo ciclos más extensos con un tamaño de imagen elevado.

Tras la evaluación inicial de resultados, se realizaron ajustes aprovechando las funcionalidades de *Data Augmentation* integradas en la biblioteca *Ultralytics*. Asimismo, el entrenamiento se repitió en un equipo de mayor capacidad (Ryzen 7 7700X, 32 GB RAM, 8 GB VRAM), lo que permitió modificar y ampliar los hiperparámetros utilizados:

```

#VEHICLES#
epoch=15
imgsz=512
batch=32
lr0=0.001
cos\lr=True
warmup\_epochs=3
weight\_decay=5e-4
patience=5
device=0
box=8.0
df1=2.0
cls=0.3
kobj=0.5
# lista de aumentaciones
mosaic=1.0
mixup=0.15
hsv\_h=0.015
hsv\_s=0.7
hsv\_v=0.4
degrees=10.0
translate=0.1
scale=0.5
shear=2.0

```

```

fliplr=0.5

#LIGHTS#
epochs=25
imgsz=512
batch=32
lr0=0.001
cos\_lr=True
warmup\_epochs=3
weight\_decay=5e-4
patience=5
device=0
box=10.0
dfl=2.0
cls=0.1
kobj=0.5
# lista de aumentaciones
mosaic=1.0
mixup=0.1
hsv\_h=0.015
hsv\_s=0.6
hsv\_v=0.4
degrees=5.0
translate=0.05
scale=0.3
shear=1.0
fliplr=0.5

#SIGNS#
epochs=30
imgsz=512
batch=8
lr0=0.001
cos\_lr=True
warmup\_epochs=3
weight\_decay=5e-4
patience=5
device=0
box=10.0
dfl=2.0
cls=0.1
kobj=0.5
# lista de aumentaciones
mosaic=1.0
mixup=0.1
hsv\_h=0.015
hsv\_s=0.6
hsv\_v=0.4

```

```
degrees=5.0
translate=0.05
scale=0.3
shear=1.0
fliplr=0.5
```

### 5.1.5. Implementación del algoritmo SORT

Para el seguimiento de objetos en secuencias de video se implementó el algoritmo Deep-SORT (*Simple Online and Realtime Tracking*). Este método permitió asociar detecciones consecutivas en diferentes fotogramas, asignando identificadores únicos a cada objeto con el fin de mantener su seguimiento a lo largo del tiempo.

### 5.1.6. Evaluación del sistema

Para evaluar el sistema fue necesario disponer de secuencias de video que representaran infracciones de tránsito. Dada la dificultad de obtener material real aplicable, se optó por emplear el entorno virtual del videojuego *Grand Theft Auto V* (GTA V), el cual ofrece una simulación urbana altamente realista que reproduce con precisión la dinámica del tráfico, las condiciones de iluminación y el comportamiento de peatones y vehículos.

Diversos trabajos previos han demostrado la utilidad de este entorno para la generación de datos sintéticos y la validación de modelos de percepción visual. Por ejemplo, Hurl et al. (2019) generaron el conjunto de datos PreSIL a partir de GTA V, destacando su fidelidad visual y diversidad de escenarios. De forma similar, Doan et al. (2018) desarrollaron la herramienta G2D, que permite controlar condiciones ambientales y parámetros de cámara, mientras que Ciampi et al. (2023) crearon el benchmark MC-GTA para el seguimiento de vehículos en entornos sintéticos multicámara. Estas investigaciones evidencian que GTA V constituye una plataforma válida, versátil y reproducible para experimentación en visión por computador.

Una vez configurado el entorno virtual, se procedió a la recolección de datos. Participaron dos sujetos con roles definidos: el primero empleó la función de *cámara en primera persona* para simular una cámara de tránsito fija, mientras que el segundo ejecutó las infracciones de tránsito correspondientes a las señales evaluadas. Este procedimiento permitió obtener registros visuales realistas y controlados, adecuados para el posterior procesamiento por el sistema y la evaluación de su desempeño en la detección de infracciones.

El sistema completo sigue la secuencia de ejecución descrita a continuación. En primer lugar, se cargan los tres modelos de detección junto con el video seleccionado para el análisis, y se inicializa el *tracker*. Posteriormente, se inicia el procesamiento de los cuadros del video. Los tres primeros pasos consisten en la detección de vehículos, señales y semáforos, actualizando el *tracker* en cada etapa. Luego, se analiza cada vehículo por separado, actualizando su estado que se conforma por el siguiente diccionario:

- **frames\_stop\_alto**: Número de cuadros en los que el vehículo se detuvo ante una señal de Alto.
- **frames\_stop\_red**: Número de cuadros en los que el vehículo se detuvo ante un semáforo en Rojo.
- **violation\_alto**: Indicador de infracción por no respetar una señal de Alto.
- **violation\_red**: Indicador de infracción por no respetar un semáforo en Rojo.
- **violation\_uturn**: Indicador de infracción por realizar un giro en U prohibido.
- **prev\_pos**: Coordenadas de la posición anterior del vehículo.
- **position\_history**: Historial completo de posiciones registradas.
- **approaching\_alto**: Historial de acciones del vehículo respecto a señales de Alto detectadas.
- **approaching\_red**: Historial de acciones del vehículo respecto a semáforos detectados.
- **exempt\_alto**: Bandera que exime al vehículo de una infracción de Alto específica, cuando no le aplica o actuó correctamente.
- **exempt\_red**: Bandera que exime al vehículo de una infracción de Rojo específica, cuando no le aplica o actuó correctamente.
- **near\_no\_uturn**: Indicador de proximidad a una señal de “Prohibido el Giro en U”.

La actualización del estado se realiza recorriendo las detecciones de señales y semáforos presentes en el cuadro actual. Se verifica si estos elementos afectan directamente al vehículo, es decir, si el carril por el que circula se encuentra bajo su influencia. Esto se determina mediante el cálculo del vector de velocidad a partir de las posiciones previas, lo que permite establecer si la señal o semáforo se encuentra en la dirección del movimiento del vehículo.

Cuando una señal afecta al vehículo, se aplican las validaciones correspondientes a cada tipo de infracción. En el caso de la señal de Alto y el semáforo en Rojo, se espera que el vehículo se detenga al menos durante dos segundos. Para los semáforos, la validación finaliza si en el cuadro siguiente se detecta una luz Verde. Si el vehículo actúa correctamente, se le exonera de la infracción asociada.

En cuanto a la señal de “Prohibido el Giro en U”, el sistema primero verifica que exista una señal de este tipo en las cercanías y que esté orientada hacia el vehículo. Luego evalúa si se dispone de suficiente historial de cuadros para determinar que el vehículo está ejecutando efectivamente un giro en U. Si se cumple esta condición, se calcula el ángulo de giro y su variación durante la trayectoria; si estos parámetros exceden el límite establecido, se registra la infracción correspondiente.

Finalmente, el sistema presenta de forma visual las detecciones realizadas y las alertas de infracciones cometidas por cada vehículo.

# CAPÍTULO 6

---

## Resultados

---

### 6.1. Resultados

#### 6.1.1. Vehículos

Luego del entrenamiento por 15 épocas, donde cada una se tardó un promedio de 3 minutos, se obtuvieron los primeros resultados para la red YOLO encargada de detectar automóviles. Se utilizó la siguiente distribución de clases:

- **Camión de carga:** Representado por la clase 0.
- **Vehículo liviano:** Representado por la clase 1.
- **SUV:** Representado por la clase 2.
- **Bus de Transporte:** Representado por la clase 3.

Como se puede ver en la *Figura 7*, hay un recopilatorio de algunas imágenes procesadas por la red durante el entrenamiento. Existe detección de todas las clases y las cajas de contorno están cerradas para el vehículo en cuestión, lo que muestra un buen desempeño durante el entrenamiento.



Figura 7: Ejemplos de detección y clasificación de vehículos

En la **Figura 8** se pueden ver las métricas obtenidas durante el entrenamiento y validación de esta red. El desempeño esencial del modelo se basa en los resultados de la métrica *box\_loss* la cual representa, como se puede inferir por el nombre, qué tan acertadas son las predicciones de cajas de contorno. Durante el entrenamiento se puede ver que la gráfica de dicha métrica tiene forma casi perfecta. La mejora es clara y disminuye la magnitud de la pérdida considerablemente. Por otro lado, durante la validación la forma de la métrica es más compleja. Presenta máximos locales pero en su forma final se puede ver una clara mejoría.

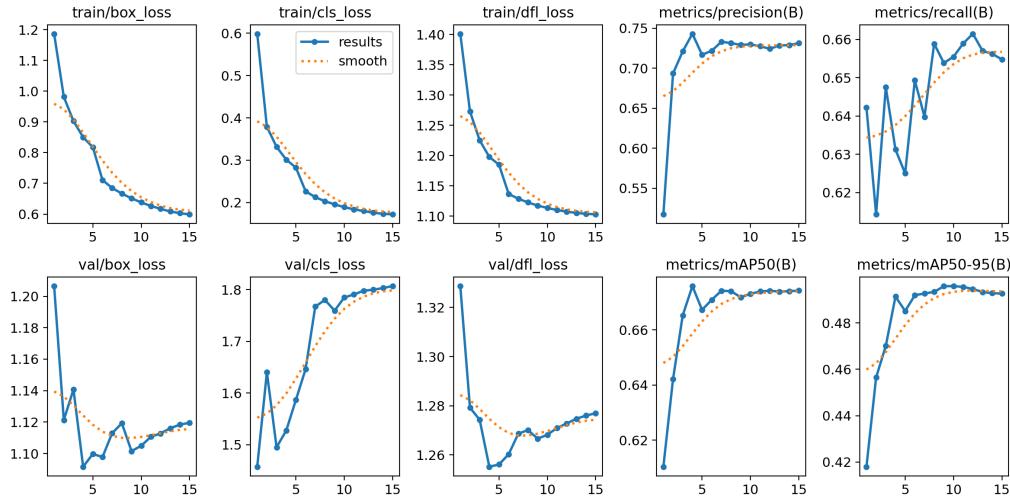


Figura 8: Métricas de YOLO en detección y clasificación de automóviles.

### 6.1.2. Señales de Tránsito

A diferencia de la red de vehículos, la red de señales de tránsito tuvo 30 épocas de 3 segundos promedio cada una. La diferencia principal son la cantidad de datos recopilados, pues de señales de tránsito se obtuvieron alrededor de 350 registros. De igual manera, los resultados de este modelo pueden verse en las *Figuras 9 y 10*. En la primera se pueden observar los ejemplos de ambas clases las cuales siguen la siguiente estructura:

- **Señal de Alto:** Representado por la clase 0.
- **Señal de Prohibido el Giro en U:** Representado por la clase 1.

Sin importar el fondo de la imagen, la red es capaz de detectar la señal, predecir una caja de contorno ajustada a la misma y su clase.

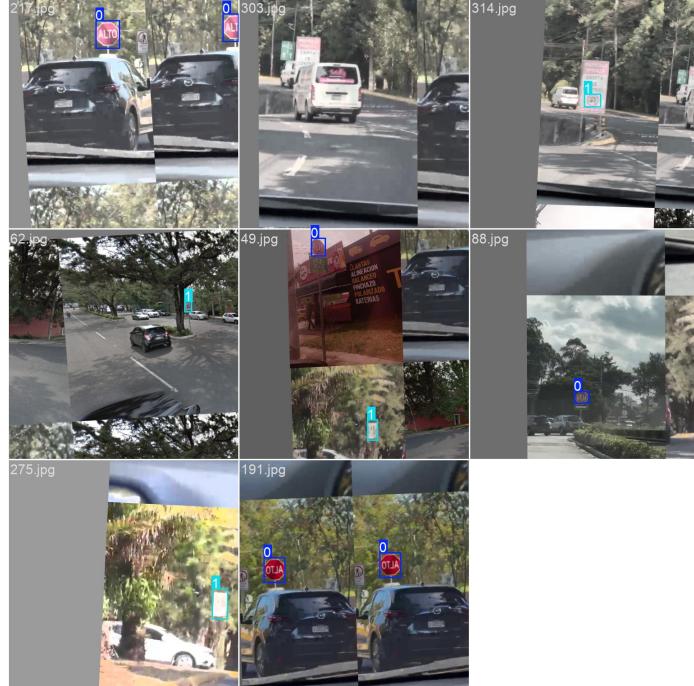


Figura 9: Ejemplos de detección y clasificación de señales de tránsito.

Las métricas obtenidas pueden verse en la segunda imagen. En este caso, las métricas de interés son tanto *box\_loss* como *cls\_loss*, donde la segunda define qué tan acertada es la predicción de clase. Puede verse en la figura que *box\_loss* tuvo complicaciones, aunque la forma final de la gráfica indica que es capaz de generalizar correctamente. Por otro lado, *cls\_loss* muestra buen desempeño tanto en entrenamiento como en validación.

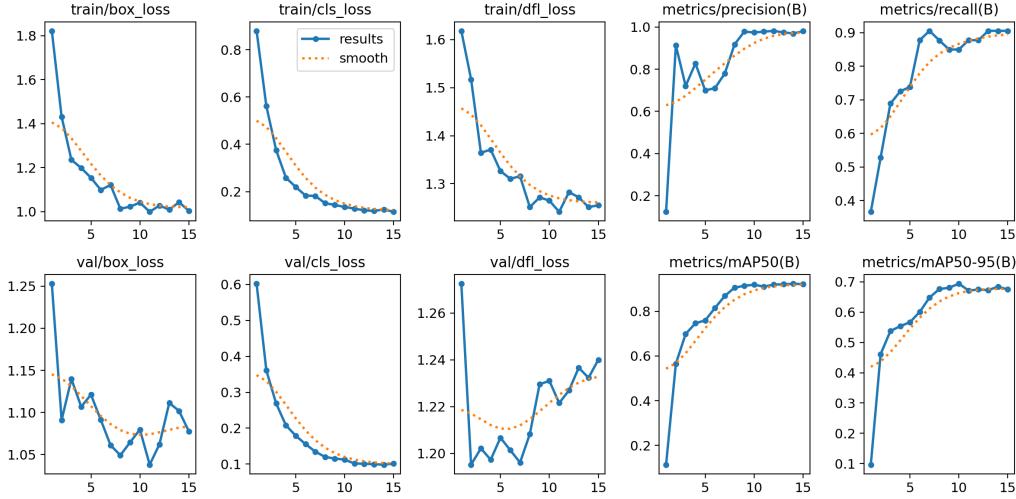


Figura 10: Métricas de YOLO en detección y clasificación de señales de tránsito.

### 6.1.3. Semáforos

Esta red tuvo 25 épocas de 7 segundos promedio cada una. Similar a lo sucedido con las señales de tránsito, existe un desbalanceo de clases entre los semáforos en rojo y verde con semáforos en amarillo. En este caso, esta situación no es problemática debido a que los estados importantes para la detección de infracciones son 'en verde' y 'en rojo'. Los resultados de la red pueden verse en las *Figuras 11 y 12*. La distribución de clases es de la siguiente forma:

- **Semáforo en Amarillo:** Representado por la clase 0.
- **Semáforo en Rojo:** Representado por la clase 1.
- **Semáforo en Verde:** Representado por la clase 2.

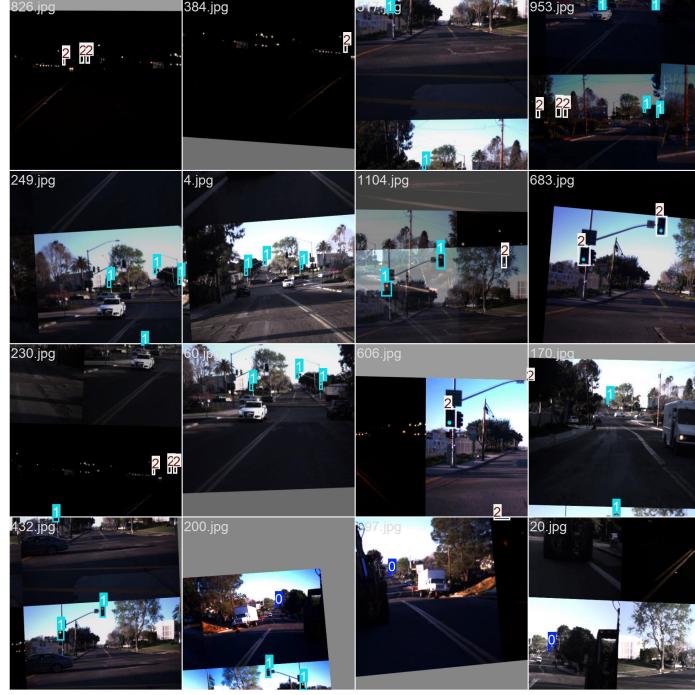


Figura 11: Ejemplos de detección y clasificación de semáforos.

En la segunda figura se puede ver el desempeño durante el entrenamiento y validación. Las métricas importantes para esta red son las mismas que para la red de señales (*box\_loss* y *cls\_loss*) ya que ambas serán procesadas de la misma manera durante la ejecución del sistema. Se puede ver que la predicción de clases es muy buena. La forma de la gráfica es casi perfecta y las magnitudes de pérdida son bajos. Para la predicción de cajas de contorno se pueden encontrar más complicaciones. Durante la validación hay más máximos locales pero la forma general de la gráfica indica que el modelo es capaz de generalizar.

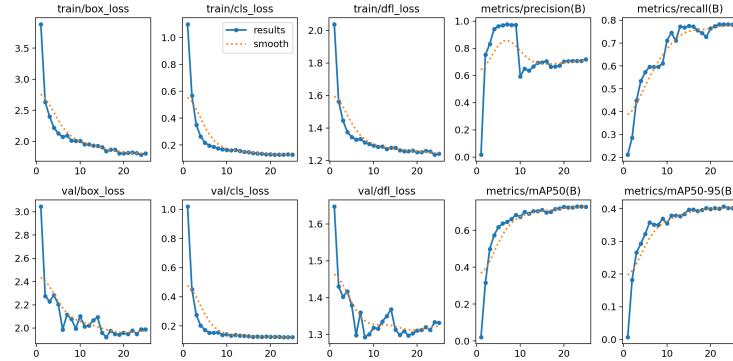


Figura 12: Métricas de YOLO en detección y clasificación de semáforos.

#### 6.1.4. Sistema completo

El sistema completo se validó mediante vídeos tomados del entorno virtual que provee *GTA V*. Los resultados para la detección de infracciones según cada señal o semáforo fueron exitosas. El sistema es capaz de alertar cuando un vehículo está cometiendo una infracción. Como se puede ver en las *Figuras 13, 14 y 15*, se le coloca el tag correspondiente al vehículo que ha cometido la infracción. El catálogo de tags es:

- ALTO VIOLATION
- RED LIGHT VIOLATION
- U-TURN VIOLATION



Figura 13: Ejemplo del sistema completo en funcionamiento para Semáforos en Rojo.



Figura 14: Ejemplo del sistema completo en funcionamiento para señal de Prohibido el Giro en U.



Figura 15: Ejemplo del sistema completo en funcionamiento para señal de Alto.



# CAPÍTULO 7

---

## Discusión de resultados

---

### 7.1. Discusión de resultados

La arquitectura YOLO demostró ser una elección adecuada para los propósitos del sistema. Su capacidad de detección en tiempo real y su bajo costo computacional relativo permitieron implementar un flujo de procesamiento estable y eficiente. Si bien el modelo requiere un tiempo inicial de carga menor a dos minutos, este retraso no afecta el rendimiento operativo, ya que las detecciones se ejecutan de forma fluida y constante. En el contexto de un despliegue mediante API, este tiempo de arranque resulta irrelevante, puesto que el servicio permanecería activo de manera continua y preparado para recibir solicitudes entrantes.

Durante las primeras iteraciones del entrenamiento, se observaron signos de *overfitting* en las tres redes (vehículos, semáforos y señales), manifestados principalmente en la divergencia entre las curvas de entrenamiento y validación. Este fenómeno fue mitigado mediante la incorporación de técnicas de *Data Augmentation* y el ajuste de hiperparámetros, en particular del tamaño de lote, el número de épocas y el uso del programador de tasa de aprendizaje (*coseine learning rate*). Estas medidas mejoraron significativamente la capacidad de generalización de los modelos, evidenciada por la reducción sostenida de las pérdidas y la mayor estabilidad de las curvas de validación en las últimas épocas.

Uno de los desafíos más notables fue el desbalanceo de clases en los conjuntos de datos empleados. El conjunto *UA-DETRAC* presentaba una predominancia de vehículos livianos frente a camiones o autobuses, mientras que el conjunto *LISA* mostraba una cantidad reducida de ejemplos de semáforos en amarillo y semáforos en distintos ambientes o iluminación. En el caso del dataset propio de señales de tránsito, la clase “Prohibido el giro en U” fue menos representada que la clase “Alto”. Este desbalance impacta en la precisión de detección de las clases minoritarias y puede conducir a sesgos en la toma de decisiones del sistema. Aunque se aplicaron técnicas de aumentación y reponderación de pérdida para compensar parcialmente este efecto, los resultados sugieren que la incorporación de datos adicionales y más equilibrados seguiría siendo una prioridad para futuras versiones.

En cuanto al desempeño individual de los modelos, la red de vehículos mostró un comportamiento aceptable tanto en la detección de contornos como en la clasificación por tipo, con pérdidas bajas y convergencia estable. La red de semáforos también alcanzó resultados satisfactorios, especialmente en la identificación de luces rojas y verdes, que son las clases críticas para la detección de infracciones. No obstante, la clase amarilla evidenció una menor consistencia, atribuible tanto al desbalanceo de datos como a la alta variabilidad de condiciones de iluminación y ángulo de captura en las imágenes.

El modelo de señales de tránsito presentó resultados aceptables considerando la limitada cantidad de datos disponibles ( 350 imágenes). Aunque las métricas de pérdida reflejan cierta variabilidad, la red demostró una correcta capacidad de generalización y una detección robusta frente a fondos complejos o condiciones de iluminación adversas. La fidelidad de las cajas de contorno sugiere que el modelo logró aprender las características distintivas de cada tipo de señal, incluso con ejemplos limitados.

A nivel sistémico, la integración de los tres modelos con el algoritmo DeepSORT permitió un seguimiento coherente y estable de los objetos a lo largo de los fotogramas. La asociación de detecciones mediante identificadores únicos fue esencial para validar las infracciones de manera persistente, evitando errores de reasignación entre vehículos. En las pruebas realizadas en el entorno virtual de *GTA V*, el sistema mostró un desempeño sólido, identificando correctamente infracciones de tipo “Alto”, “Semáforo en rojo” y “Giro en U prohibido”. La combinación de detección, clasificación y seguimiento resultó efectiva para generar alertas visuales precisas y oportunas.

Sin embargo, se identificaron limitaciones inherentes al uso de datos sintéticos. Aunque el entorno de *GTA V* ofrece una simulación visual realista, las texturas, el modelado de movimiento y las condiciones ambientales no replican completamente la complejidad de entornos urbanos reales. Esto podría afectar la capacidad de generalización del sistema ante datos reales de tráfico, especialmente en aspectos como reflejos, occlusiones o degradación de señales por desgaste o suciedad. Para mitigar este riesgo, futuras versiones del sistema deberían incorporar una fase de validación con material de cámaras reales, lo que permitiría cuantificar la transferencia de aprendizaje entre dominios sintéticos y reales.

Por último, la eficiencia computacional del sistema es un aspecto destacable. La arquitectura propuesta mantiene una ejecución rápida incluso con múltiples detecciones simultáneas, gracias al uso optimizado de YOLOv8 y DeepSORT. Esto confirma la viabilidad del sistema no solo como prototipo experimental, sino también como base para una posible aplicación en sistemas de monitoreo vial o en plataformas inteligentes de control de tráfico.

En síntesis, los resultados alcanzados demuestran que el enfoque propuesto es funcional y escalable, aunque su desempeño podría incrementarse con conjuntos de datos más balanceados y una validación en entornos reales. La combinación de modelos ligeros y técnicas de seguimiento eficientes permite establecer una base sólida para sistemas automáticos de análisis de tráfico e identificación de infracciones con visión por computadora.

# CAPÍTULO 8

---

## Conclusiones

---

### 8.1. Conclusiones

1. El sistema detectó con alta precisión vehículos livianos, SUVs, camiones de carga y buses en imágenes y metrajes pregrabados.
2. El modelo identificó correctamente semáforos y señales de tránsito en diversas condiciones en imágenes y metrajes pregrabados.
3. El sistema logró analizar de forma autónoma la interacción entre vehículos y elementos viales, identificando las infracciones de tránsito establecidas en el alcance del proyecto.



# CAPÍTULO 9

---

## Recomendaciones

---

### 9.1. Recomendaciones generales

La recomendación principal se relaciona con una limitación encontrada respecto al objetivo inicial del proyecto. El sistema fue concebido para operar de manera completamente automática, sin requerir intervención previa del usuario. Sin embargo, los resultados muestran que este enfoque incrementa de forma considerable la carga computacional al exigir la detección simultánea de vehículos, semáforos y señales. Tras analizar el desempeño, se concluye que una alternativa más eficiente consiste en restringir las detecciones únicamente a vehículos y parametrizar manualmente la posición de semáforos y señalizaciones en función de la perspectiva fija de la cámara.

Si bien esta solución reduce la autonomía del sistema, pues requiere una etapa de configuración inicial, también disminuye de manera significativa el costo de procesamiento y la latencia, al evitar que el modelo ejecute detecciones complejas en cada cuadro. En escenarios donde las cámaras se mantienen estáticas, esta compensación resulta razonable y permite obtener un mejor equilibrio entre rendimiento, eficiencia y estabilidad operativa.

#### 9.1.1. Recomendaciones complementarias

Además de la recomendación principal, se identificaron varias áreas de mejora que pueden fortalecer el rendimiento y la capacidad de generalización del sistema:

- **Ampliación y balanceo de los conjuntos de datos.** Se recomienda incrementar la cantidad de ejemplos de clases minoritarias, particularmente señales poco representadas y semáforos en distintas condiciones de iluminación. Un dataset más equilibrado reduciría sesgos y mejoraría la estabilidad del modelo durante la inferencia.
- **Validación con datos reales.** Aunque el entorno sintético utilizado proporciona condiciones controladas, se aconseja incorporar una fase adicional de validación con

material capturado por cámaras reales. Esto permitiría evaluar la transferencia de aprendizaje y ajustar el sistema a escenarios urbanos con variaciones climáticas, luminosidad compleja o desgaste físico de las señales.

- **Evaluación del seguimiento en escenarios complejos.** Se recomienda analizar el desempeño del algoritmo de seguimiento en situaciones con occlusiones prolongadas o mayor densidad vehicular, ya que estos factores pueden afectar la persistencia de identificadores y la consistencia del análisis temporal.

# CAPÍTULO 10

---

## Bibliografía

---

- [1] Instituto Nacional de Estadística (INE), *Indicadores de Accidentes de Tránsito 2023. Período 2019–2023*, <https://www.ine.gob.gt/sistema/uploads/2024/11/21/2024112180407xRp35JuZin7nN2x88Me8MVCQvyZCnu5K.pdf>, Consultado el 10 de noviembre de 2025, 2024.
- [2] Municipalidad de Guatemala (MuniGuate), *Semáforos inteligentes*, Municipalidad de Guatemala, 2024. dirección: <https://www.muniguate.com/blog/2024/02/22/semaforos-inteligentes/>.
- [3] D. Muiruri, *History of Computing -Computer Vision*, [https://studies.helsinki.fi/api/courses/cur/file/pub/system/files/attachments/2023-05/Dennis\\_Muiruri\\_Computer\\_Vision.pdf?hmac=13ebc0970f2c9373d80e3812bda9944b843401bd46f855a4e0832023](https://studies.helsinki.fi/api/courses/cur/file/pub/system/files/attachments/2023-05/Dennis_Muiruri_Computer_Vision.pdf?hmac=13ebc0970f2c9373d80e3812bda9944b843401bd46f855a4e0832023).
- [4] D. A. Lienhard, *David H. Hubel and Torsten N. Wiesel's Research on Optical Development in Kittens*, Embryo Project Encyclopedia, Published 2017-10-11. ISSN: 1940-5030, 2017. dirección: <https://hdl.handle.net/10776/12995>.
- [5] J. Zhou, N. C. Benson, K. N. Kay y J. Winawer, *Compressive Temporal Summation in Human Visual Cortex*, Journal of Neuroscience, 38(3):691–709, Epub 2017 Nov 30. PMID: 29192127; PMCID: PMC5777115, 2018. DOI: 10.1523/JNEUROSCI.1724-17.2017.
- [6] R. A. Kirsch, H. Belar, G. W. Brown y C. L. Saxton, *Experiments in Processing Pictorial Information with a Digital Computer*, Proceedings of the Eastern Joint Computer Conference, pp. 221–229, 1957.
- [7] D. Marr y S. Ullman, *Directional Selectivity and Its Use in Early Visual Processing*, Proceedings of the Royal Society of London. Series B, Biological Sciences, Vol. 211, No. 1183, pp. 151–180, 1981.
- [8] R. A. Kirsch, *Computers and the Human Mind*, Computers and Automation, Vol. 13, No. 6, pp. 12–19, 1964.
- [9] G. Nagy, *Twenty Years of Document Image Analysis*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 1, pp. 38–62, 1992.

- [10] L. G. Roberts, *Machine Perception of Three-Dimensional Solids*, PhD Thesis, Massachusetts Institute of Technology, 1963.
- [11] K. Fukushima, *Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position*, Biological Cybernetics, Vol. 36, pp. 193–202, 1980.
- [12] Y. LeCun, L. Bottou, Y. Bengio y P. Haffner, *Gradient-Based Learning Applied to Document Recognition*, Proceedings of the IEEE, Vol. 86, No. 11, pp. 2278–2324, 1998.
- [13] S. C. Bergmann D., *Backpropagation*, <https://www.ibm.com/think/topics/backpropagation>, 2025.
- [14] Y. LeCun, L. Bottou, Y. Bengio y P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86(11):2278–2324, 1998.
- [15] A. Krizhevsky, I. Sutskever y G. E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*, Advances in Neural Information Processing Systems (NeurIPS), Vol. 25, 2012.
- [16] K. Simonyan y A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, arXiv preprint arXiv:1409.1556, 2014.
- [17] C. Szegedy, W. Liu, Y. Jia et al., *Going Deeper with Convolutions*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9, 2015.
- [18] C. Szegedy, W. Liu, Y. Jia et al., *Going deeper with convolutions*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–9, 2015.
- [19] K. Simonyan y A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, International Conference on Learning Representations (ICLR), 2015.
- [20] K. He, X. Zhang, S. Ren y J. Sun, *Deep Residual Learning for Image Recognition*, arXiv preprint arXiv:1512.03385, 2015.
- [21] T. D. Science, *The Annotated ResNet-50*, 2023. dirección: <https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758>.
- [22] P. Viola y M. Jones, *Robust Real-Time Object Detection*, International Journal of Computer Vision, Based on the 2001 IEEE CVPR paper; pp. 137–154, 2004.
- [23] Ultralytics, *What is R-CNN? A Quick Overview*, 2024. dirección: <https://www.ultralytics.com/blog/what-is-r-cnn-a-quick-overview>.
- [24] J. Redmon, S. Divvala, R. Girshick y A. Farhadi, *You Only Look Once: Unified, Real-Time Object Detection*, 2016. arXiv: 1506.02640 [cs.CV]. dirección: <https://arxiv.org/abs/1506.02640>.
- [25] M. El Aidouni, *Understanding YOLO and YOLOv2*, 2018. dirección: <https://manalelaidouni.github.io/Understanding%20YOLO%20and%20YOLOv2.html>.
- [26] A. Bewley, Z. Ge, L. Ott, F. Ramos y B. Upcroft, *Simple Online and Realtime Tracking*, IEEE International Conference on Image Processing (ICIP), pp. 3464–3468, 2016.
- [27] N. Wojke, A. Bewley y D. Paulus, *Simple Online and Realtime Tracking with a Deep Association Metric*, IEEE International Conference on Image Processing (ICIP), pp. 3645–3649, 2017.

- [28] Y. Du, Z. Zhao, Y. Song et al., *StrongSORT: Make DeepSORT Great Again*, 2023. arXiv: 2202.13514 [cs.CV]. dirección: <https://arxiv.org/abs/2202.13514>.
- [29] Ultralytics, *Ultralytics Trackers Documentation: ByteTrack, BoT-SORT, OC-SORT and Related Multi-Object Tracking Methods*, Ultralytics Documentation, Available at <https://docs.ultralytics.com/modes/track/>, 2024.
- [30] Policía Nacional Civil, *Reglamento De Tránsito*, 2020. dirección: <https://transito.gob.gt/wp-content/uploads/2021/01/CARTILLA-DE-INFRACCIONES-Y-SANCIONES-DE-TR%C3%81NSITO.pdf>.
- [31] D. Dede, M. A. Sarsil, A. Shaker, O. Altintas y O. Ergen, *Next-gen traffic surveillance: AI-assisted mobile traffic violation detection system*, 2023. arXiv: 2311.16179 [cs.CV]. dirección: <https://arxiv.org/abs/2311.16179>.
- [32] A. Dixit, R. Bhandari y P. Maji, *RoadEye: A Framework For Detecting Traffic Rule Violations*, Proceedings of the 26th International Conference on Distributed Computing and Networking (ICDCN '25), ISBN: 9798400710629, New York, NY, USA, 2025. DOI: 10.1145/3700838.3703683. dirección: <https://doi.org/10.1145/3700838.3703683>.
- [33] G. Research, *Project Green Light: Using AI to reduce stop-and-go traffic and emissions in cities*, <https://sites.research.google/gr/greenlight/>, 2023.
- [34] O. o. M. I. City of Detroit, *City's Office of Mobility Innovation Unveils Dashboard to Improve Traffic Flow and Pedestrian Safety*, News release, City of Detroit, 2025. dirección: <https://detroitmi.gov/news/citys-office-mobility-innovation-unveils-dashboard-improve-traffic-flow-and-pedestrian-safety-0>.
- [35] Ultralytics, *Data Preprocessing Techniques for Annotated Computer Vision Data — Splitting the Dataset*, [https://docs.ultralytics.com/guides/preprocessing\\_annotated\\_data/#splitting-the-dataset](https://docs.ultralytics.com/guides/preprocessing_annotated_data/#splitting-the-dataset), 2025.
- [36] Ultralytics, *Model Evaluation Metrics in Object Detection and Classification*, Ultralytics Documentation, 2025. dirección: [https://docs.ultralytics.com/guides/model\\_evaluation\\_metrics/](https://docs.ultralytics.com/guides/model_evaluation_metrics/).
- [37] C.-Y. Wang, A. Bochkovskiy y H.-Y. M. Liao, *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*, 2022. arXiv: 2207.02696 [cs.CV]. dirección: <https://arxiv.org/abs/2207.02696>.
- [38] E. Commission, *Ethics Guidelines for Trustworthy AI*, European Commission High-Level Expert Group on Artificial Intelligence, 2021. dirección: <https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai>.
- [39] L. Wen, D. Du, Z. Cai et al., *UA-DETRAC: A New Benchmark and Protocol for Multi-Object Detection and Tracking*, UA-DETRAC Benchmark, 2020. dirección: <https://detrac-db.rit.albany.edu>.
- [40] S. Messaoud, S. Maldonado-Bascon, L. Boquete, D. López-Ferreras y N. Pérez de la Blanca, *LISA Traffic Light and Sign Dataset*, Laboratory for Intelligent and Safe Automobiles (LISA), UC San Diego, 2010. dirección: <http://cvrr.ucsd.edu/LISA/lisa-traffic-sign-dataset.html>.
- [41] HumanSignal, *labelImg: A graphical image annotation tool*, <https://github.com/HumanSignal/labelImg>, 2024.

- [42] B. Hurl, K. Czarnecki, M. Salvarani, A. Wong y S. L. Waslander, *Precise Synthetic Image and LiDAR (PreSIL) Dataset for Autonomous Vehicle Perception*, arXiv preprint arXiv:1905.00160, 2019. dirección: <https://arxiv.org/abs/1905.00160>.
- [43] K. N. Doan, C. Madsen, T. Nguyen, Q.-S. Tran, W. MacLean y S. L. Waslander, *G2D: from GTA to Data—A GPU Based Framework for Procedurally Generating Synthetic Data Using Grand Theft Auto V*, arXiv preprint arXiv:1806.07810, 2018. dirección: <https://arxiv.org/abs/1806.07810>.
- [44] L. Ciampi, C. Gennaro y G. Amato, *MC-GTA: A New Benchmark for Multi-Camera Vehicle Tracking Based on GTA V*, arXiv preprint arXiv:2303.12928, 2023. dirección: <https://arxiv.org/abs/2303.12928>.
- [45] IBM, *¿Qué es la información de identificación personal (PII)?* <https://www.ibm.com/es-es/think/topics/pii>, 2025.
- [46] R. Girshick, J. Donahue, T. Darrell y J. Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 580–587, 2014.
- [47] A. Stock, *Imagen de señal de ALTO*, 2025. dirección: <https://stock.adobe.com/gt/search?k=se%C3%B1al+de+alto>.
- [48] D. de Tránsito de Guatemala, *Señal de ALTO*, 2025. dirección: <https://transito.gob.gt/alto-aprenda-las-senales-verticales/>.
- [49] TN8, *¿Por qué la señal ALTO es octogonal?* 2025. dirección: <https://www.tn8.tv/tendencias/425598-has-preguntado-por-que-señal-alto-es-octogonal/>.
- [50] Provialmex, *Señal de ALTO*, 2025. dirección: <https://provialmex.com.mx/señal-de-alto/>.
- [51] Reddit, *Alto Stop Sign IRL*, 2025. dirección: [https://www.reddit.com/r/geoguessr/comments/1h93dtj/alto-stop-sign\\_irl/](https://www.reddit.com/r/geoguessr/comments/1h93dtj/alto-stop-sign_irl/).
- [52] C. Traffic, *Señales Restrictivas*, 2025. dirección: [https://www.cactustraffic.com.mx/upcp\\_product/senales-restrictivas/](https://www.cactustraffic.com.mx/upcp_product/senales-restrictivas/).
- [53] Soy502, *Cámara capta hombre dañando señal de alto*, 2025. dirección: <https://www.soy502.com/articulo/camara-vigilancia-capta-hombre-danando-señal-alto-101025>.
- [54] S. Supply, *Señal de ALTO*, 2025. dirección: <https://sigusupply.com/producto/señal-alto/>.
- [55] M. Obson, *Avanza obra del crucero*, 2025. dirección: <https://mediosobson.com/2025/06/10/avanza-obra-del-crucero-en-mision-del-real-tome-precauciones-al-circular/>.
- [56] P. Pública, *Oxec y consultas indígenas*, 2025. dirección: <https://www.plazapublica.com.gt/content/oxec-se-convierte-en-la-batalla-por-las-consultas-los-pueblos-indigenas>.
- [57] P. México, *Camino Real La Paz*, 2025. dirección: <https://www.posta.com.mx/tag/camino-real-la-paz>.
- [58] L. G. P. Peñasco, *Instalación de altos con nomenclatura*, 2025. dirección: <https://laguiadepuertopenasco.com/avanza-gobierno-municipal-con-la-instalacion-del-85-de-altos-con-nomenclatura/>.

- [59] Facebook, *Grupo CNK*, 2025. dirección: <https://www.facebook.com/grupocnk/>.
- [60] Construex, *Señal ALTO Tijuana*, 2025. dirección: [https://www.construex.com.mx/exhibidores/soluciones\\_viales\\_senalamientos/producto/senal\\_alto\\_tijuana\\_baja\\_california](https://www.construex.com.mx/exhibidores/soluciones_viales_senalamientos/producto/senal_alto_tijuana_baja_california).
- [61] YouTube, *Video sobre señalización vial*, 2025. dirección: <https://www.youtube.com/watch?v=rVClvzHz12Y>.
- [62] S. L. RC, *Colocación de nueva señalización*, 2025. dirección: <https://sanluisrc.gob.mx/comunicacion/notas/se-analiza-colocacion-de-nueva-senalizacion-vial>.
- [63] Publimagen, *Señal ALTO*, 2025. dirección: <https://publimagen.mx/producto/alto/>.
- [64] Facebook, *Señalización para fraccionamientos*, 2025. dirección: <https://www.facebook.com/Construzonehillo/posts/922662085171385/>.
- [65] Provialmex, *Senalamientos viales para escuelas*, 2025. dirección: <https://provialmex.com.mx/5-senalamientos-viales-para-escuelas/>.
- [66] Infotam, *Reposición de señalamientos dañados*, 2025. dirección: <https://infotam.info/reposicion-de-senalamientos-danados/>.
- [67] L. N. de Tulum, *Reglamento de tránsito*, 2025. dirección: <https://www.lasnoticiasdetulum.com/2014/11/10-situaciones-que-preve-el-reglamento.html>.
- [68] Facebook, *Senvial*, 2025. dirección: <https://www.facebook.com/senvial/>.
- [69] Paperblog, *Esto y cerda al paso*, 2025. dirección: <https://es.paperblog.com/mx/esto-y-cerda-al-paso-205876/>.
- [70] Provialmex, *Catálogo de señalización vial*, 2025. dirección: <https://provialmex.com.mx/>.
- [71] Alamy, *Imagen de señal ALTO en México*, 2025. dirección: <https://www.alamy.com/stock-photo-stop-sign-written-in-spanish-in-mexico-city-mexico-15451202.html>.
- [72] M. U. México, *Seguridad peatonal*, 2025. dirección: <https://mobiliario-urbano-mexico.com/blog/2022/02/la-seguridad-peatonal/>.
- [73] S. Reynosa, *Catálogo vial*, 2025. dirección: <https://sysreynosa.com.mx/>.
- [74] R. de Banqueta, *Mantenimiento a señalamientos*, 2025. dirección: <https://reporterosdebanqueta.com/transito-da-mantenimiento-a-senalamientos-en-chetumal/>.
- [75] E. Norte, *Renovarán señalética*, 2025. dirección: <https://www.elnorte.com/renovaran-senaletica-en-calles-del-centro/ar2773423>.
- [76] D. del Yaqui, *Modernizan nomenclatura*, 2025. dirección: <https://diariodelyaqui.mx/ciudadobregon/modernizan-nomenclatura-en-ciudad-obregon/42294>.
- [77] L. T. Noreste, *Supervisan señalización*, 2025. dirección: <https://latalachanoreste.com/2019/11/06/supervisa-adrian-senalizacion-vial-y-nomenclatura/>.
- [78] V. México, *Señales restrictivas*, 2025. dirección: <https://www.vw.com.mx/es/experiencia/tips/senales-transito-restrictivas.html>.
- [79] Medium, *Stop doing this if you want to learn a language*, 2025. dirección: <https://medium.com/illumination/stop-doing-this-if-you-want-to-learn-a-language-4e1f8b2b86e5>.

- [80] Alamy, *Old cracked ALTO sign*, 2025. dirección: <https://www.alamy.com/stock-photo-an-old-cracked-grungy-road-sign-saying-alto-which-means-stop-in-spanish-94131583.html>.
- [81] T. / . X, *Publicación sobre tránsito y señalización*, 2025. dirección: <https://x.com/TransitoSoleda1/status/1510964310176116737>.
- [82] iStock, *Fotos de señal Prohibido girar en U*, 2025. dirección: <https://www.istockphoto.com/es/fotos/prohibido-girar-en-u>.
- [83] Facebook, *Publicación sobre señal de Prohibido girar en U*, 2025. dirección: <https://www.facebook.com/pzactual/posts/1333683026706850/>.
- [84] Facebook, *Publicación sobre señal reglamentaria*, 2025. dirección: <https://www.facebook.com/elenchufle/posts/1073353808135630/>.
- [85] M. de Xela, *Publicación sobre señales de tránsito*, 2025. dirección: <https://www.facebook.com/MunicipalidadXela/posts/5776698485693081/>.
- [86] L. P. Honduras, *Convieren retorno en entrada de emergencia*, 2025. dirección: <https://www.laprensa.hn/sanpedro/convierten-retorno-entrada-porton-emergencia-hospital-mario-rivas-CG10163991>.
- [87] Instagram, *Publicación fotográfica de señal vial*, 2025. dirección: [https://www.instagram.com/p/Cu2dR2mPKG0/?img\\_index=1](https://www.instagram.com/p/Cu2dR2mPKG0/?img_index=1).
- [88] Google, *Google Street View Imagery*, Imágenes consultadas para referencias de señalización y entorno vial, 2025. dirección: <https://www.google.com/maps>.

