



**unl**

Universidad  
Nacional  
de Loja

**Universidad Nacional de Loja**

**Facultad de la Energía, las Industrias y los  
Recursos Naturales no Renovables**

**Carrera de Ingeniería en Telecomunicaciones**

**Desarrollo de un prototipo basado en Visión por Computadora para  
detectar la invasión del carril de buses en la ciudad de Loja.**

**Trabajo de Integración Curricular,  
previo a la obtención del Título de  
Ingeniera en Telecomunicaciones**

**AUTORA:**

Ibeth Katherine Román Carrión

**DIRECTOR:**

Ing. Luis Eduardo Rodríguez Montoya, Mg. Sc.

**Loja – Ecuador**

**2024**

## Certificación



unl

Universidad  
Nacional  
de Loja

Sistema de Información Académico  
Administrativo y Financiero - SIAAF

### CERTIFICADO DE CULMINACIÓN Y APROBACIÓN DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Yo, **Rodriguez Montoya Luis Eduardo**, director del Trabajo de Integración Curricular denominado “**DESARROLLO DE UN PROTOTIPO BASADO EN VISIÓN POR COMPUTADORA PARA DETECTAR LA INVASIÓN DEL CARRIL DE BUSES EN LA CIUDAD DE LOJA**”, perteneciente al estudiante **IBETH KATHERINE ROMAN CARRION**, con cédula de identidad N° **1105736597**.

**Certifico:**

Que luego de haber dirigido el **Trabajo de Integración Curricular**, habiendo realizado una revisión exhaustiva para prevenir y eliminar cualquier forma de plagio, garantizando la debida honestidad académica, se encuentra concluido, aprobado y está en condiciones para ser presentado ante las instancias correspondientes.

Es lo que puedo certificar en honor a la verdad, a fin de que, de así considerarlo pertinente, el/la señor/a docente de la asignatura de **Integración Curricular**, proceda al registro del mismo en el Sistema de Gestión Académico como parte de los requisitos de acreditación de la Unidad de Integración Curricular del mencionado estudiante.

Loja, 8 de Marzo de 2024



Firmado electrónicamente por:  
**LUIS EDUARDO  
RODRIGUEZ MONTOYA**

F) -----

**DIRECTOR DE TRABAJO DE INTEGRACIÓN  
CURRICULAR**



Certificado TIC/TT.: UNL-2024-000967

1/1

Educamos para **Transformar**

## **Autoría**

Yo, **Ibeth Katherine Román Carrión**, declaro ser autora del presente **Trabajo de Integración Curricular** y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos, de posibles reclamos o acción legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación de mi Trabajo de Integración Curricular en el Repositorio Digital Institucional - Biblioteca Virtual.

**Firma:**

**Cedula de Identidad:** 1105736597

**Fecha:** 02/12/24

**Correo electrónico:** [ibeth.roman@unl.edu.ec](mailto:ibeth.roman@unl.edu.ec)

**Teléfono:** 0988583854

**Carta de autorización por parte de la autora, para la consulta, reproducción parcial o total y/o publicación electrónica del texto completo, del Trabajo de Integración Curricular**

Yo **Ibeth Katherine Román Carrión**, declaro ser autor Trabajo de Integración Curricular denominado: **Desarrollo de un prototipo basado en Visión por Computadora para detectar la invasión del carril de buses en la ciudad de Loja.**, como requisito para optar el título de Ingeniero en Telecomunicaciones, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que, con fines académicos, muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Integración Curricular que realice un tercero.

Para constancia de esta autorización, suscribo, en la ciudad de Loja, a los veinte nueve días del mes de noviembre del dos mil veinticuatro.

**Firma:**

**Autor:** Ibeth Katherine Román Carrión

**Cédula de identidad:** 1105736597

**Dirección:** Catamayo, Barrio San Antonio

**Correo electrónico:** ibeth.roman@unl.edu.ec

**Teléfono:** 0988583854

**DATOS COMPLEMENTARIOS:**

**Director Trabajo de Integración Curricular:** Ing. Luis Eduardo Rodríguez Montoya, Mg. Sc.

## **Dedicatoria**

Para mi amado Johan Ezequiel,

Si en algún momento llegas a leer este contenido, quiero transmitirte que el conocimiento es nuestra mayor riqueza. Por ello, te animo a prepararte tanto física como intelectualmente, perseguir tus sueños con todo tu corazón, enfrentar los desafíos con valentía y no rendirte ante las adversidades. Aunque en el camino encontrarás un sinnúmero de dificultades, con esfuerzo y dedicación, tus metas pueden ser alcanzadas.

*Recuerda que siempre estaremos juntos, desde tu primer latido hasta el último de mis suspiros.*

*Con amor infinito,*

**Ibeth Katherine Román Carrión.**

## **Agradecimientos**

Con profunda emoción, deseo expresar mis más sinceros agradecimientos:

Al creador por otorgarme el privilegio de estar viviendo esta experiencia aquí y ahora.

A mi pequeño Ezequiel, por ser mi compañero de estudios y acompañarme en tantas desveladas.

A mis padres, hermanos y sobrinas, cuyo apoyo incondicional ha sido el motor que me impulsa a seguir adelante en cada paso de mi vida.

A mis colegas telekos, por ser más que compañeros, por ser verdaderos hermanos y apoyarme dentro y fuera de las aulas.

Al personal del GAD Municipal de la ciudad de Loja, quienes han permitido que este proyecto se pueda desarrollar favorablemente.

A los estimados docentes de la Poderosa Carrera de Telecomunicaciones, cuya dedicación y sabiduría han enriquecido mi formación profesional, sus enseñanzas perdurarán a lo largo de mi vida laboral y personal.

Y a todas las personas que han sido parte de este proceso...

***¡Gracias totales!***

**Ibeth Katherine Román Carrión.**

## Índice de contenido

<b>Portada</b> .....	<b>i</b>
<b>Certificación</b> .....	<b>ii</b>
<b>Autoría</b> .....	<b>iii</b>
<b>Carta de autorización</b> .....	<b>iv</b>
<b>Dedicatoria</b> .....	<b>v</b>
<b>Agradecimientos</b> .....	<b>vi</b>
<b>Índice de contenido</b> .....	<b>vii</b>
Índice de tablas: .....	x
Índice de figuras: .....	xi
Índice de anexos: .....	xiii
<b>1. Título</b> .....	<b>1</b>
<b>2. Resumen</b> .....	<b>2</b>
Abstract .....	3
<b>3. Introducción</b> .....	<b>4</b>
<b>4. Marco teórico</b> .....	<b>6</b>
4.1. Antecedentes.....	6
4.2. Base legal.....	7
4.3. Transporte Público.....	8
4.3.1. <i>Definición</i> .....	8
4.3.2. <i>Sistema Integrado de Transporte Urbano (SITU)</i> .....	8
4.3.3. <i>Componentes principales</i> .....	8
4.4. Visión por computadora .....	9
4.4.1. <i>Definición</i> .....	9
4.4.2. <i>Etapas del proceso</i> .....	9
4.4.3. <i>Modelos de aprendizaje</i> .....	10
4.5. Procesamiento de imágenes digitales .....	11
4.6. Técnicas para detección de objetos.....	12
4.6.1. <i>Detección de bordes.</i> .....	12
4.6.2. <i>Segmentación de imágenes.</i> .....	12
4.6.3. <i>Reconocimiento de patrones.</i> .....	13
4.6.4. <i>Extracción de características.</i> .....	13

4.7.	Métricas utilizadas para evaluación.....	14
4.8.	Reconocimiento óptico de caracteres .....	15
4.9.	Elementos de software.....	15
4.9.1.	<i>Lenguaje de programación</i> .....	15
4.9.2.	<i>Entorno de desarrollo</i> .....	16
4.9.3.	<i>Etiquetado de imágenes</i> .....	17
4.9.4.	<i>Almacenamiento de resultados</i> .....	17
4.10.	Elementos de hardware.....	18
4.10.1.	<i>Cámara</i> .....	18
4.10.2.	<i>Dispositivo de grabación</i> .....	19
<b>5.</b>	<b>Metodología.....</b>	<b>21</b>
5.1.	Área de Estudio .....	21
5.2.	Materiales y métodos.....	21
5.2.1.	<i>Modelo de entrenamiento</i> .....	21
5.2.2.	<i>Software</i> .....	21
5.2.3.	<i>Hardware</i> .....	22
5.3.	Diseño del Sistema .....	23
5.2.4.	<i>Implementación del sistema</i> .....	23
5.2.5.	<i>Recopilación de imágenes</i> .....	24
5.2.6.	<i>Preparación de datos</i> .....	27
5.2.7.	<i>Entrenamiento del modelo</i> .....	33
5.2.8.	<i>Evaluación</i> .....	40
<b>6.</b>	<b>Resultados .....</b>	<b>47</b>
6.1.	Resultados para primer conjunto de datos .....	47
6.2.	Resultados para primer conjunto con aumento de datos .....	50
6.3.	Resultados para el segundo conjunto de datos .....	53
6.4.	Resultados para el segundo conjunto con aumento de datos.....	55
6.5.	Resultados para el tercer conjunto de datos.....	58
6.6.	Resultados para el tercer conjunto con aumento de datos .....	61
6.7.	Resultados de la detección de invasión .....	66
<b>7.</b>	<b>Discusión .....</b>	<b>68</b>
<b>8.</b>	<b>Conclusiones .....</b>	<b>70</b>

<b>9. Recomendaciones .....</b>	<b>71</b>
<b>10. Bibliografía .....</b>	<b>72</b>
<b>11. Anexos .....</b>	<b>77</b>

## Índice de tablas:

<b>Tabla 1.</b> Modelos de aprendizaje.....	11
<b>Tabla 2.</b> Técnicas de procesamiento de imágenes.....	11
<b>Tabla 3.</b> Técnicas de detección de bordes .....	12
<b>Tabla 4.</b> Técnicas de segmentación de imágenes .....	13
<b>Tabla 5.</b> Técnicas de reconocimiento de patrones.....	13
<b>Tabla 6.</b> Extracción de características .....	14
<b>Tabla 7.</b> Métricas de evaluación.....	14
<b>Tabla 8.</b> Herramientas y bibliotecas de OCR .....	15
<b>Tabla 9.</b> Lenguajes de programación utilizados en visión por computadora .....	15
<b>Tabla 10.</b> Entornos de desarrollo para python.....	16
<b>Tabla 11.</b> Aplicaciones para etiquetado de imágenes.....	17
<b>Tabla 12.</b> Gestores de base de datos.....	18
<b>Tabla 13.</b> Cámaras.....	18
<b>Tabla 14.</b> Almacenamiento para cámara .....	19
<b>Tabla 15.</b> Capas de la arquitectura de YOLOv8 .....	33
<b>Tabla 16.</b> Parámetros para entrenamiento del primer conjunto de datos .....	35
<b>Tabla 17.</b> Valores para entrenamiento del segundo conjunto de datos .....	35
<b>Tabla 18.</b> Valores para entrenamiento del tercer conjunto de datos.....	35
<b>Tabla 19.</b> Resultados de métricas de primer conjunto de datos .....	41
<b>Tabla 20.</b> Resultados de métricas del primer conjunto con aumento de datos.....	42
<b>Tabla 21.</b> Resultados de métricas del segundo conjunto datos .....	43
<b>Tabla 22.</b> Resultados de métricas del segundo conjunto con aumento de datos .....	44
<b>Tabla 23.</b> Resultados de métricas del tercer conjunto de datos .....	45
<b>Tabla 24.</b> Resultados de métricas del tercer conjunto de datos.....	46

## Índice de figuras:

<b>Figura 1.</b> Congestión vehicular en la Av. Manuel Agustín Aguirre. ....	6
<b>Figura 2.</b> Ruta corredora exclusivo para buses .....	9
<b>Figura 3.</b> Etapas de un proceso de visión por computador.....	10
<b>Figura 4.</b> Ubicación del prototipo.....	23
<b>Figura 5.</b> Parte interna del prototipo.....	24
<b>Figura 6.</b> Parte Externa del prototipo .....	24
<b>Figura 7.</b> Conjunto de datos original .....	25
<b>Figura 8.</b> Conjunto de datos con aumento de datos.....	25
<b>Figura 9.</b> Conjunto de datos original .....	26
<b>Figura 10.</b> Conjunto de datos con aumento de datos.....	26
<b>Figura 11.</b> Conjunto de datos original .....	27
<b>Figura 12.</b> Conjunto de datos con aumento de datos.....	27
<b>Figura 13.</b> Imagen Original - Redimensionamiento de imágenes .....	28
<b>Figura 14.</b> Imagen resultante - Redimensionamiento de imágenes.....	28
<b>Figura 15.</b> Imagen original - Técnica de ampliación/reducción.....	29
<b>Figura 16.</b> Imagen resultante - Técnica de ampliación/reducción.....	29
<b>Figura 17.</b> Imagen original - Técnica de rotación .....	30
<b>Figura 18.</b> Imagen resultante - Técnica de ampliación/reducción.....	30
<b>Figura 19.</b> Imagen original - Técnica modificación de saturación .....	31
<b>Figura 20.</b> Imagen resultante - Técnica de saturación.....	31
<b>Figura 21.</b> Imagen original – Técnica modificación de brillo.....	32
<b>Figura 22.</b> Imagen resultante – Técnica modificación de brillo.....	32
<b>Figura 23.</b> Imagen original – Técnica desenfoque .....	33
<b>Figura 24.</b> Imagen resultante – Técnica desenfoque .....	33
<b>Figura 25.</b> Zona de detección – Invasión no permitida .....	36
<b>Figura 26.</b> Zona de detección – Invasión permitida .....	37
<b>Figura 27.</b> Diagrama de flujo del software.....	38
<b>Figura 28.</b> Procesamiento de imágenes para la placa .....	39
<b>Figura 29.</b> Evaluación del primer conjunto de datos.....	40
<b>Figura 30.</b> Evaluación del primer conjunto de datos con aumento de datos .....	42
<b>Figura 31.</b> Evaluación del segundo conjunto .....	43
<b>Figura 32.</b> Evaluación del segundo conjunto con aumento de datos.....	44
<b>Figura 33.</b> Evaluación del tercer conjunto de datos .....	45

<b>Figura 34.</b> Evaluación del tercer conjunto de datos .....	46
<b>Figura 35.</b> Pérdidas de caja delimitadora y clases primer conjunto .....	47
<b>Figura 36.</b> Métricas de evaluación generales primer conjunto.....	48
<b>Figura 37.</b> Resultados de primer conjunto de datos .....	49
<b>Figura 38.</b> Pérdidas de caja delimitadora y clases primer conjunto con aumento de datos ....	50
<b>Figura 39.</b> Métricas de evaluación generales primer conjunto con aumento de datos .....	51
<b>Figura 40.</b> Resultados de primer conjunto con aumento de datos.....	52
<b>Figura 41.</b> Pérdidas de caja delimitadora y clases segundo conjunto de datos .....	53
<b>Figura 42.</b> Métricas generales del modelo.....	54
<b>Figura 43.</b> Resultados de segundo conjunto de datos.....	55
<b>Figura 44.</b> Pérdidas de caja delimitadora y clases del segundo conjunto con aumentación de datos .....	56
<b>Figura 45.</b> Métricas generales con el segundo conjunto con aumentación de datos .....	57
<b>Figura 46.</b> Resultados del segundo conjunto con aumentación de datos .....	58
<b>Figura 47.</b> Pérdidas de caja delimitadora y clases tercer conjunto de datos.....	59
<b>Figura 48.</b> Métricas generales con el tercer conjunto de datos.....	60
<b>Figura 49.</b> Resultados del tercer conjunto de datos.....	61
<b>Figura 50.</b> Pérdidas de caja delimitadora y clases tercer conjunto con aumento de datos .....	62
<b>Figura 51.</b> Métricas generales con el tercer conjunto con aumento de datos.....	63
<b>Figura 52.</b> Resultados del tercer conjunto con aumento de datos .....	64
<b>Figura 53.</b> Resultados para la mAP .....	65
<b>Figura 54.</b> Resultados para detección de invasión .....	66

## **Índice de anexos:**

<b>Anexo 1.</b> Bitácora Transmisión de cámara .....	77
<b>Anexo 2.</b> Recorte de frames y selección de puntos de interés .....	78
<b>Anexo 3.</b> Entrenamiento del modelo.....	79
<b>Anexo 4.</b> Conexión a la base de datos.....	81
<b>Anexo 5.</b> Código completo .....	82
<b>Anexo 6.</b> Certificación emitida por la Unidad de Movilidad.....	86
<b>Anexo 7.</b> Certificación de traducción del resumen .....	87

## **1. Título**

**Desarrollo de un prototipo basado en Visión por Computadora para detectar la invasión del carril de buses en la ciudad de Loja.**

## 2. Resumen

La visión por computadora combina técnicas de inteligencia artificial, procesamiento de imágenes y aprendizaje profundo para desarrollar sistemas que puedan analizar el contenido visual de manera automatizada y, de esta forma, lograr dar solución a problemas reales de la sociedad. En este contexto, en la ciudad de Loja, se evidencia que existe una gran cantidad de vehículos que invaden los carriles exclusivos del transporte público, una infracción catalogada como contravención de cuarta y sexta clase según el COIP. Por ello, la presente investigación tiene como objetivo desarrollar un prototipo basado en visión por computadora para detectar la invasión de carril de buses. Se utilizó una metodología de desarrollo que abarcó las siguientes etapas: implementación del sistema, recopilación de imágenes, preparación de datos, entrenamiento del modelo y evaluación. El modelo detector de objetos obtuvo resultados prometedores con una mAP50 de todas las clases del 92% y una mAP50-95 del 84%, lo que significa que es capaz de detectar una amplia variedad de objetos con alta precisión en una variedad de condiciones. En cuanto a la detección de invasión, se alcanzó un 94% de precisión con el sistema trabajando en tiempo real, concluyendo que es altamente efectivo para su implementación. Además, cabe recalcar que este proyecto proporciona un apoyo invaluable a las autoridades de tránsito en la identificación y sanción de conductores que invaden el carril de bus. No obstante, no puede trabajar de manera autónoma, requiere la colaboración de una persona que verifique y valide los datos.

**Palabras clave:** Detector de objetos, infracciones COIP, invasiones de carril, procesamiento de imágenes, visión por computadora

## **Abstract**

Computer vision combines techniques of artificial intelligence, image processing, and deep learning to develop systems that can automatically analyze visual content and thus address real societal problems. In this context, in the city of Loja, it is evident that there is a significant number of vehicles invading lanes designated for public transportation, an infraction classified as fourth and sixth-class contravention according to the COIP. Therefore, the present research aims to develop a computer vision-based prototype to detect bus lane invasions. A development methodology was employed, encompassing the following stages: system implementation, image collection, data preparation, model training, and evaluation. The object detection model yielded promising results with an mAP50 of 92% across all classes and an mAP50-95 of 84%, indicating its capability to detect a wide range of objects with high precision under diverse conditions. Regarding invasion detection, a 94% precision was achieved with the system operating in real-time, concluding its high effectiveness for implementation. Moreover, it is noteworthy that this project provides invaluable support to transit authorities in identifying and penalizing drivers who invade the bus lane. However, it cannot operate autonomously; it requires the collaboration of an individual to verify and validate the data.

**Keywords:** Object detector, COIP infractions, lane invasions, image processing, computer vision.

### 3. Introducción

En la actualidad, los sistemas de visión por computadora han emergido como un campo de investigación que brinda soluciones innovadoras destinadas a solventar problemas reales de la sociedad. En este contexto, el presente trabajo de investigación se enfoca en la creación de un prototipo capaz de identificar de manera precisa y en tiempo real los vehículos que infringen las normas de tránsito al invadir los carriles exclusivos destinados al transporte público. El constante incumplimiento de estas normas ha dado lugar a situaciones de riesgo como colisiones y accidentes de tránsito. Además, esta práctica también afecta la fluidez del tráfico, provocando congestión y retrasos en las vías principales de la ciudad de Loja.

El desarrollo del proyecto conlleva la instalación del prototipo y el desarrollo del modelo de detección basado en técnicas de visión por computadora, así como su posterior validación en condiciones reales de tráfico en la ciudad de Loja, específicamente en la calle Manuel Agustín, en donde se ha comprobado que existe gran cantidad de tráfico, además que en esta vía se encuentra el corredor exclusivo para el transporte público. Los resultados obtenidos se encontrarán alojados en una base de datos en la nube, en donde los organismos y autoridades tengan acceso y puedan tomar las medidas pertinentes de acuerdo a los estatutos establecidos en la Unidad de Control Operativa de Tránsito (UCOT) y la Ley Orgánica de Tránsito y Seguridad Vial (LOTSV).

La implementación de este prototipo aporta con notables beneficios a diversos actores de la comunidad lojana. En primer lugar, los usuarios del transporte público y peatones experimentarán un viaje más seguro y eficiente, los conductores de buses, podrán llevar a cabo sus labores con seguridad, sabiendo que existe un sistema que respalda el cumplimiento de las regulaciones viales. Finalmente, las autoridades de tránsito contarán con herramientas altamente efectivas para regular la seguridad vial en la ciudad, permitiéndoles una gestión más eficiente de la movilidad. En consecuencia, esta investigación resulta importante debido a que aborda un problema crítico que afecta significativamente la eficiencia y la seguridad del transporte público en la ciudad de Loja.

Para cubrir estas expectativas se ha tomado en cuenta investigaciones previas del campo de la visión por computadora, sistemas de vigilancia y movilidad aplicados en entornos urbanos. Los cuales sirven como base para analizar distintos modelos de entrenamiento y técnicas de detección de objetos, adaptándolas de manera específica para identificar cuando un vehículo particular invade las normas de tránsito al invadir el carril.

El objetivo general de investigación se enfoca en desarrollar un prototipo basado de visión por computadora para detectar la invasión de carril de buses en la ciudad de Loja. Así mismo los objetivos específicos que engloban este trabajo se detallan a continuación:

- Investigar técnicas de visión por computadora adecuadas para detectar vehículos de transporte público y líneas de carril.
- Crear un conjunto de datos destinado al entrenamiento del modelo de detección, por medio de la recopilación de imágenes de un tramo de la Av. Universitaria.
- Evaluar el modelo entrenado para medir su precisión y eficiencia por medio de un prototipo.

Además, es fundamental establecer preguntas de investigación destinadas a abordar aspectos teóricos y prácticos, las cuales se detallan a continuación:

- ✓ ¿Cuáles son los modelos y técnicas de visión por computadora más adecuados para la detección de objetos?
- ✓ ¿Cómo afectaría la eficacia del modelo de detección, factores como la variabilidad de la iluminación y las condiciones climáticas?
- ✓ ¿Cómo podrían los datos recopilados por el prototipo contribuir a la toma de decisiones y políticas de transporte público en Loja?

## 4. Marco teórico

### 4.1. Antecedentes

La ciudad de Loja, ubicada en la región sur de Ecuador, se caracteriza por su constante crecimiento urbano. El incremento de la población y la actividad comercial en las regiones centrales ha generado una alta demanda en términos de movilidad. Siendo el transporte público un componente vital para satisfacer las necesidades de los ciudadanos que necesitan movilizarse, la implementación de carriles exclusivos para los buses se implementó con el objetivo de mejorar la eficiencia del sistema de transporte y reducir la congestión vial. No obstante, la eficacia de estos carriles se ve amenazada por la invasión constante de vehículos particulares, disminuyendo la fluidez del tráfico y la puntualidad del transporte público.

La **Figura 1** evidencia la problemática que enfrentan a diario los conductores de transporte público, en la ciudad de Loja. La presencia constante de vehículos particulares, especialmente de taxis, en el carril destinado a los buses urbanos. Esta infracción no solo causa efectos negativos en la rutina operativa del SITU, sino que también destaca la importancia de implementar medidas efectivas para hacer cumplir las leyes y reglamentos de tránsito.

#### Figura 1.

Congestión vehicular en la Av. Manuel Agustín Aguirre.



**Fuente:** Elaboración propia

Actualmente, la supervisión y control del uso correcto del carril se realiza por medio de métodos convencionales de vigilancia, como la presencia de agentes de tránsito. No obstante, el carril abarca una extensión de aproximadamente 12 kilómetros, lo que dificulta la implementación de un elemento policial que logre controlar una invasión.

En base a lo antes mencionado, la implementación de este sistema resulta eficaz y adecuada. Por ello, se ha tomado en cuenta la instalación de una cámara en una de las avenidas que cuentan corredor exclusivo para circulación del bus con el fin detectar este tipo de

infracciones. Inicialmente, se eligió la intersección entre la Av. Universitaria y Mercadillo para esta instalación. Sin embargo, en esta zona ya se encuentran instaladas varias cámaras de seguridad con diferentes propósitos, por lo que se llevó a cabo una modificación estratégica a la Av. Manuel Agustín Aguirre y 10 de Agosto (Anexo 6. Certificación emitida por la Unidad de Movilidad). Este ajuste se realizó con la firme intención de brindar seguridad a la zona, es decir, la instalación de este dispositivo no solo contribuye al éxito del proyecto, sino que también refuerza la seguridad en este sector, donde anteriormente no existía presencia de cámaras de vigilancia cercanas.

#### **4.2. Base legal**

El presente proyecto se fundamenta bajo las leyes y reglamentos del Código Orgánico Integral Penal (COIP) en concordancia con Ley Orgánica De Transporte Terrestre Tránsito Y Seguridad Vial (LOTTTSV), vigentes en Ecuador. En particular, se hace referencia a los artículos 389 y 391 los cuales se detallan textualmente a continuación:

**Art. 389.- Contravenciones de tránsito de cuarta clase.-** Serán sancionados con multa equivalente al treinta por ciento de un salario básico unificado del trabajador en general, y reducción de seis puntos en su licencia de conducir.

Numeral 1 La o el conductor que desobedezca las órdenes de los agentes de tránsito, o que no respete las señales manuales de dichos agentes, en general toda señalización colocada en las vías públicas, tales como: semáforos, pare, ceda el paso, cruce o preferencia de vías (Ministerio de Defensa Nacional, 2021, Pág. 144).

**Art. 391. Contravenciones de tránsito de sexta clase.-** Será sancionado con multa equivalente al diez por ciento de un salario básico unificado del trabajador general y reducción de tres puntos en su licencia de conducir: **Numeral 3.** La o el conductor que invada con su vehículo las vías exclusivas asignadas a los buses de transporte rápido (Ministerio de Defensa Nacional, 2021, Pág. 142).

Los artículos mencionados se han tomado con referencia debido a que, en la actualidad, la ciudad de Loja carece de una ordenanza específica para regular infracciones relacionadas con el uso correcto de las vías públicas, específicamente el carril de bus. En consecuencia, las sanciones para este tipo de infracciones quedan bajo la jurisdicción del COIP.

Estos artículos respaldan de forma legal que la implementación del sistema de visión artificial para la detección de invasión del carril del bus en la ciudad de Loja es viable, proporcionando un marco jurídico sólido para garantizar la seguridad vial y el cumplimiento de las normativas de tránsito vigentes.

### **4.3. Transporte Público**

#### **4.3.1. Definición**

El sistema de transporte público, es una red completa de vehículos de transporte que puede desplazar a un gran número de personas a la vez y satisfacer sus necesidades de movilidad (Celi, 2018). De acuerdo con (Briceño, 2018) el objetivo principal de este sistema es proporcionar un transporte eficaz, rápido, cómodo y seguro de las personas entre los numerosos lugares donde trabajan y desarrollan sus actividades.

En la era moderna en donde todas las personas obligatoriamente necesitan movilizarse para realizar sus actividades diarias, los servicios de transporte público por medio de buses es el principal medio de movilización en las grandes y medianas ciudades (Abad, 2022)

#### **4.3.2. Sistema Integrado de Transporte Urbano (SITU)**

Dentro de la ciudad de Loja, existe un sistema de transporte público debidamente estructurado, se trata del Sistema Integrado de Transportación Urbana (SITU). Según (Díaz, 2017) este proyecto se enmarca dentro del desarrollo sustentable de la ciudad y tiene como objetivo principal mejorar el servicio de transporte público, proporcionando seguridad a los peatones.

Este sistema de transporte tiene objetivos claros reconocidos por (Hurtado Torres, 2018) los cuales se enlistan a continuación:

- ✓ Optimizar y mejorar la calidad y transporte.
- ✓ Incentivar la auto sostenibilidad del sistema.
- ✓ Mejorar condiciones de tráfico.
- ✓ Eliminar la guerra del centavo (ganar más pasajeros para mayor beneficio económico).

#### **4.3.3. Componentes principales**

El SITU fue diseñado como un sistema de transporte público masivo con una ruta principal lineal y cerrada, que se nutre de líneas alimentadoras que convergen en estaciones de integración y transferencia de pasajeros. De acuerdo con (Chamba Coronel, 2013) sus componentes principales son:

- ✓ **Ruta Troncal Principal o Corredor Exclusivo:** Diseñado para autobuses que operan en dirección norte-sur y viceversa, este corredor exclusivo tiene una extensión de 12.45 kilómetros y se encuentra ubicado en los carriles derechos de las Avenidas Manuel Agustín Aguirre, Universitaria, Cuxibamba y 8 de diciembre, reservado exclusivamente para el transporte público.

**Figura 2.**

Ruta corredora exclusivo para buses



**Fuente:** (Chamba Coronel, 2013)

- ✓ **Estaciones de Transferencia (2):** En los extremos del corredor se encuentran la "Estación Sur Podocarpus" y la "Estación Norte Zoológico", en las cuales se han planificado puntos de integración para las líneas alimentadoras que provienen de los extremos sur y norte respectivamente.
- ✓ **Subestaciones de Transferencia (3):** Las "Subestación Pablo Palacio", "Subestación Salvador Bustamante Celi" y "Subestación Plaza de la Independencia" están situadas en el tramo central del corredor.
- ✓ **Paradas Troncales:** Son 28 en total, 14 por sentido de circulación y se encuentran ubicadas a un promedio de cada 300 metros en el corredor exclusivo.

#### **4.4. Visión por computadora**

##### **4.4.1. Definición**

De acuerdo con (Sinha & Bajaj, 2021) “La visión por computadora hace que los ordenadores se asemejen a los humanos y, por tanto, utiliza el concepto de visión humana. Las capacidades de la visión humana se trasladan a los ordenadores, realizando un gran número de operaciones”.

Es una rama de la inteligencia artificial que recopila datos digitales de imágenes o audio y los procesa para identificar sus características. Cada paso del proceso implica la obtención, el análisis, la interpretación, la identificación y la extracción de datos de imagen. La capacidad de las computadoras para percibir cualquier contenido visible y actuar en consecuencia es el resultado de este extenso procesamiento (Leonard, 2021)

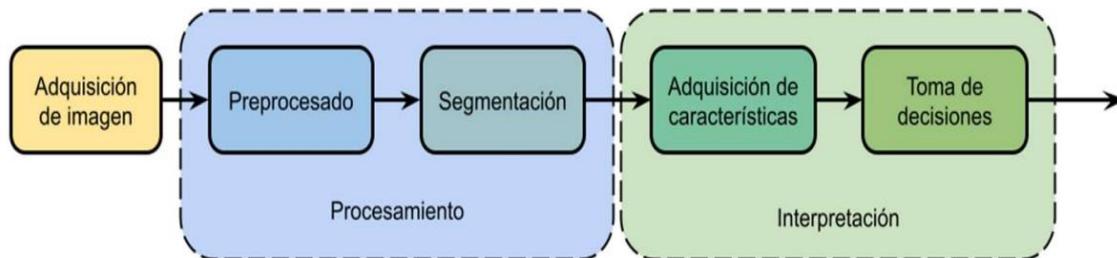
##### **4.4.2. Etapas del proceso**

Un proceso de visión por computadora generalmente consta de varias etapas o fases, que permiten analizar y extraer información útil de las imágenes. Según (Navarro Soler, 2022)

las etapas pueden variar dependiendo de la aplicación específica y la complejidad del problema a resolver, sin embargo, se suelen tomar en cuenta estas tres etapas principales: adquisición de la imagen, procesamiento de la imagen e interpretación.

**Figura 3.**

Etapas de un proceso de visión por computador



**Fuente:** (Navarro Soler, 2022)

- 1. Adquisición de la imagen:** En esta etapa, se capturan las imágenes a través de cámaras, sensores u otros dispositivos de adquisición. La calidad y la resolución de las imágenes son aspectos cruciales para obtener resultados precisos en etapas posteriores (Jiménez Rodríguez, 2019).
- 2. Procesamiento de la imagen:** Una vez adquiridas las imágenes, se aplican técnicas de procesamiento y segmentación para mejorar la calidad de la imagen y resaltar características relevantes. Estas técnicas pueden incluir filtrado, corrección de distorsiones, realce de bordes y ajuste del contraste, entre otros (Navarro Soler, 2022)
- 3. Interpretación:** La etapa final implica el análisis e interpretación de la información contenida en la imagen procesada. En esta fase, se utilizan algoritmos y técnicas de reconocimiento de patrones para identificar objetos, realizar seguimientos, medir dimensiones, clasificar elementos y extraer datos relevantes de la imagen para que el sistema pueda tomar una decisión (Pérez & Valiente, 2018).

#### **4.4.3. Modelos de aprendizaje**

Los modelos de aprendizaje automático que se entrenan para realizar tareas de visión por computadora, como la detección de objetos, clasificación de imágenes o segmentación de instancias, según menciona (Chai et al., 2021). Estos modelos utilizan técnicas de visión por computadora como componentes fundamentales, pero están diseñados para tareas particulares y se entrenan en conjuntos de datos específicos. Entre los más destacados se encuentran:

**Tabla 1.**

Modelos de aprendizaje

Modelo de aprendizaje	Precisión	Capacidad de detección	Pérdidas	Tiempo de ejecución	Enfoque
YOLO	Alta	Alta	Mínimas	Menor	Una sola pasada
SSD	Moderada	Baja	Considerables	Mayor	Una sola pasada
R-CNN	Alta	Alta	Considerables	Menor	Propuestas regionales y clasificación

Fuente(Reis et al., 2023)

#### 4.5. Procesamiento de imágenes digitales

Como se documenta en (Alegre et al., 2016a):

Las principales técnicas de procesamiento digital de imágenes mejoran y realzan sus características. Las operaciones que son llevadas a cabo por este tipo de técnicas implican la alteración del histograma de la imagen o el procesamiento del valor de sus píxeles.

**Tabla 2.**

Técnicas de procesamiento de imágenes

Técnica	Descripción	Objetivo	Ventajas	Desventajas
Umbralización del histograma	Convierte una imagen a escala de grises en una imagen binaria. Distribuye uniformemente los valores de intensidad en la imagen.	Segmentar la imagen en dos regiones: fondo y objeto.	Simple y rápido	Puede ser sensible al ruido.
Ecuilibración del histograma	Resaltan los bordes y detalles en la imagen.	Mejorar el contraste de la imagen.	efectivo para imágenes con bajo contraste.	Puede aumentar el ruido en la imagen.
Filtros paso alto y bajo	Consiste en modificar el tamaño de una imagen	Mejorar la nitidez de la imagen.	Resalta detalles	Genera ruido
Redimensión	Proporciona luminosidad a cada pixel de la imagen.	Adaptar la imagen a un tamaño específico	Mejora visualización y rendimiento.	Pérdida de calidad o distorsión de la imagen.
Brillo	Invierte los valores de intensidad de la imagen.	Ajustar la apariencia de la imagen.	Variable	Pierde detalles en áreas oscuras o brillantes.
Inversión		Crear un efecto negativo o resaltar detalles en áreas oscuras.	Simple	Pierde detalles

Rotación y giro	Rota la imagen en un ángulo específico.	Cambiar o corregir la orientación de la imagen.	Simple	Computacionalmente costos
Suavizado	Promedia los valores de intensidad de los píxeles vecinos.	Eliminar el ruido y suavizar la imagen.	Elimina ruido	Difumina borde y detalles innecesarios.

**Fuente:** (Alegre et al., 2016a; González et al., 2006)

#### 4.6. Técnicas para detección de objetos

De acuerdo con (Ansari, 2020) en aplicaciones de visión por computadora, las imágenes se adquieren de diversas Fuente como cámaras, archivos o flujos de otras aplicaciones. En estos casos es necesario realizar ajustes en estas imágenes, como ajustes de tamaño, rotaciones, cambios de color, eliminación del fondo o fusiones de imágenes y algunos otros métodos y algoritmos con la finalidad de procesar los datos para que sean aptos para implementarse en un modelo (Pág. 56).

##### 4.6.1. Detección de bordes.

Según (Alegre et al., 2016b) “El realce de bordes de una imagen se suele realizar a través de filtros espaciales” (Pág. 45). Las técnicas clásicas de detección de borde se basan principalmente en diferenciar a la imagen, esto es, encontrar la primera o segunda derivada respecto a los ejes x y y, o gradiente (Sucar, 2015)

**Tabla 3.**

Técnicas de detección de bordes

Técnica	Función amiento	Ventajas	Desventajas
<b>Operadores de Roberts</b>	Máscaras para detectar bordes diagonales, verticales y horizontales.	Buena respuesta ante bordes diagonales	Mayor sensibilidad al ruido
<b>Operadores de Prewitt, Sobel y Frei-Chen</b>	Calculan una aproximación de las primeras derivadas en los ejes x e y.	Detección de bordes nítidos.	Cada uno se enfoca en diferentes tipos de líneas.
<b>Laplaciano</b>	Basados en la segunda derivada.	Destaca bordes y detalles finos. Elimina el ruido.	Requiere alta capacidad computacional
<b>Algoritmo de Canny</b>	Utilizan el gradiente de la imagen para realzar los bordes	Logra bordes precisos y bien definidos.	Requiere ajuste de múltiples parámetros.

**Fuente:** (Alegre et al., 2016b; Puga et al., 2021; Sucar, 2015)

##### 4.6.2. Segmentación de imágenes.

Según (Alegre et al., 2016) “Segmentar una imagen digital significa dividirla en zonas disyuntivas e individualizas. Es decir, consiste en diferenciar los diversos objetivos y donde se encuentran del fondo, que puede ser más o menos complejo de la imagen” (Pág. 57).

En otras palabras, segmentar una imagen consiste en dividirla en regiones significativas con el fin de localizar objetivos o determinar límites.

**Tabla 4.**

Técnicas de segmentación de imágenes

<b>Técnica</b>	<b>Función amiento</b>	<b>Ventajas</b>	<b>Desventajas</b>
Semántica	Asigna una etiqueta de clase a cada píxel de una imagen.	Información detallada sobre la clase de cada píxel en una imagen, lo que puede ser útil para tareas como la detección de objetos.	No distingue entre instancias de objetos individuales, lo que puede ser un problema en escenarios donde es importante detectar y segmentar objetos específicos.
Instancias	Detecta y segmenta cada instancia de objeto en una imagen.	Permite la separación de instancias de objetos.	Computacionalmente costoso. Dificultades para manejar objetos superpuestos.
Panóptica	Unifica las tareas de segmentación semántica y de instancias, generando una segmentación coherente de escenas rica y completa.	Proporciona una comprensión completa del contenido de la imagen.	Computacionalmente costoso. Dificil de implementar.

**Fuente:** (Kirillov et al., 2018)

#### **4.6.3. Reconocimiento de patrones.**

Para (Kuncheva & Whitaker, 2015) “el reconocimiento de patrones consiste en asignar objetos a las clases. Cada objeto se describe mediante un conjunto de  $x$  características y puede ser considerado como un punto en algunos  $x$  espacio de características dimensionales”.

**Tabla 5.**

Técnicas de reconocimiento de patrones

<b>Técnica</b>	<b>Función amiento</b>	<b>Ventajas</b>	<b>Desventajas</b>
Supervisada	Utiliza un conjunto de datos de entrenamiento etiquetado para entrenar un modelo que puede reconocer patrones en datos no etiquetados.	Alto rendimiento y precisión cuando se dispone de datos etiquetados.	Dependencia de datos etiquetados de alta calidad y costosos de obtener.
No Supervisada	No requiere etiquetas previas y busca patrones intrínsecos en los datos, como la agrupación o reducción de dimensionalidad.	Útil cuando las etiquetas son escasas o costosas de obtener.	Resultados menos interpretables y requiere más capacidad de procesamiento.

**Fuente:** (Kuncheva & Whitaker, 2015)

#### **4.6.4. Extracción de características.**

Como señala (Sherer, 2022) la selección de características es un procedimiento que posibilita la identificación de los atributos más significativos dentro de un conjunto de datos. Esta técnica permite descartar los que no son pertinentes o duplicados, de manera que solo se

consideren los más relevantes durante el análisis. Esto conlleva a un incremento en la eficacia de los modelos de análisis y una mejora en la precisión de los resultados.

**Tabla 6.**

Extracción de características

Técnica	Funciónamiento	Ventajas	Desventajas
Histograma de Colores	Representa la distribución de colores en una imagen mediante un histograma.	Simple y eficaz para la caracterización de la distribución de colores.	No considera la estructura espacial de la imagen.
Histograma de Gradientes Orientados (HOG)	Calcula la distribución de gradientes en la imagen, útil para detectar bordes y texturas.	Efectivo en la detección de formas y bordes.	Sensible a la orientación y variaciones de escala.
Transformada de Ondículas (Wavelet Transform)	Descompone la imagen en diferentes escalas y orientaciones para analizar tanto las características de baja como de alta frecuencia	Capaz de capturar detalles finos y texturas.	Requiere un conocimiento previo de la elección de las ondículas y parámetros de descomposición.
Transformada de Hough	Se utiliza para detectar líneas, círculos y otras formas en la imagen mediante la transformada de Hough.	Eficaz en la detección de formas geométricas específicas.	Sensible a la parametrización y costoso computacionalmente.

**Fuente:** (Corrales Bastidas, 2022; Martínez, 2022; Nieto, 2020)

#### 4.7. Métricas utilizadas para evaluación

Existen varias métricas que se utilizan evaluar un sistema de visión por computador, las cuales están destinadas en medir su rendimiento y precisión en función de los resultados obtenidos:

**Tabla 7.**

Métricas de evaluación

Métrica	Descripción	Fórmula													
Matriz de Confusión	Compara las clases predichas con las clases reales en un conjunto de datos.	<table border="1"> <tr> <td colspan="2" rowspan="2"></td> <th colspan="2">True Class</th> </tr> <tr> <th>Negative</th> <th>Positive</th> </tr> <tr> <th rowspan="2">Predicted Class</th> <th>Negative</th> <td>True Negative (TN)</td> <td>False Negative (FN)</td> </tr> <tr> <th>Positive</th> <td>False Positive (FP)</td> <td>True Positive (TP)</td> </tr> </table>			True Class		Negative	Positive	Predicted Class	Negative	True Negative (TN)	False Negative (FN)	Positive	False Positive (FP)	True Positive (TP)
		True Class													
		Negative	Positive												
Predicted Class	Negative	True Negative (TN)	False Negative (FN)												
	Positive	False Positive (FP)	True Positive (TP)												
Precisión ( <i>Precision</i> )	Porcentaje de verdaderos positivos respecto a todos los positivos predichos por el modelo.	$P = \frac{(VP)}{(VP) + (FP)}$													
Recuperación ( <i>Recall</i> )	Proporción de verdaderos positivos respecto a todos los casos positivos reales en el conjunto de datos.	$R = \frac{(VP)}{(VP) + (FN)}$													
F1 Score	Combina precisión y recuperación en un solo valor, proporcionando una medida equilibrada del rendimiento del modelo.	$F1 = 2 \frac{P \times R}{P + R}$													

**Fuente:** (Boesch, 2023; Kashyapa, 2022)

## 4.8. Reconocimiento óptico de caracteres

En el artículo titulado *Review On Optical Character Recognition*, (Ahmed et al., 2019) define al Reconocimiento óptico de caracteres como técnica de identificación automática de caracteres a partir de una imagen, además permite el reconocer caracteres alfanuméricos impresos o escritos a mano, texto numérico, letras y símbolos en un formato procesable por computadora, incluidos ASCII, Unicode, entre otros.

**Tabla 8.**  
Herramientas y bibliotecas de OCR

Características	EasyOCR	Tesseract
Lenguajes soportados	Admite más de 80 idiomas	Múltiples idiomas, cantidad no especificada
Manejo de fuentes	Reconoce gran cantidad de fuentes incluida la manuscrita.	Enfrenta dificultades con algunas fuentes
Velocidad	Alta	Alta
Rendimiento	Alto rendimiento en diferentes conjuntos de datos	Alto rendimiento, excepto en conjuntos de datos con ruido
Calidad de reconocimiento	Alto nivel de reconocimiento	Nivel medio de reconocimiento

Fuente: (Chawla et al., 2022; Ren et al., 2023)

## 4.9. Elementos de software

### 4.9.1. Lenguaje de programación

Según (Joyanes Aguilar, 2008): “Los lenguajes de programación se utilizan para escribir programas. Los programas de las computadoras modernas constan de secuencias de instrucciones que se codifican como secuencias de dígitos numéricos que podrán entender dichas computadoras” (Pág. 66). Para ello se utiliza traductores conocidos como compiladores o intérpretes, los cuales convierten un programa en instrucciones legibles en lenguajes máquina (0 y 1, bits) para que pueda ser entendido por el computador. Para un modelo basado en visión por computadora se pueden utilizar los siguientes lenguajes:

**Tabla 9.**  
Lenguajes de programación utilizados en visión por computadora

Lenguaje de programación	Ventajas	Desventajas	Aplicaciones
Python	<ul style="list-style-type: none"> <li>✓ Amplia comunidad y soporte</li> <li>✓ Numerosas bibliotecas de visión por computadora</li> <li>✓ Ofrece una sintaxis clara y legible</li> <li>✓ Fácil de aprender y usar.</li> </ul>	Puede ser más lento que lenguajes compilados.	Detección de objetos, reconocimiento de patrones, segmentación de imágenes.

Java	<ul style="list-style-type: none"> <li>✓ Rendimiento excepcional debido a su naturaleza compilada.</li> <li>✓ Bibliotecas como JavaCV.</li> </ul>	Menor rendimiento con respecto a Python.	Aplicaciones de visión en dispositivos Android, procesamiento de imágenes en entornos empresariales.
MATLAB	<ul style="list-style-type: none"> <li>✓ Herramientas de procesamiento de imágenes y visión por computadora.</li> <li>✓ Permite prototipado y visualización.</li> </ul>	Software privativo Licencias costosas	Investigación en visión por computadora, procesamiento de imágenes médicas.

**Fuente:** (IBM, 2022; MathWorks, 2023; Toro Bonilla, 2022)

#### 4.9.2. Entorno de desarrollo

Un entorno de desarrollo integrado (IDE) representa una herramienta de software que proporciona una eficaz asistencia a los programadores en el proceso de creación de código de software. Este sistema eleva la productividad de los desarrolladores al unificar funciones como la edición, creación, pruebas y empaquetado de software en una aplicación de fácil manejo. De manera análoga a cómo los escritores recurren a editores de texto y los contables a hojas de cálculo, los desarrolladores de software hacen uso de los IDE con el propósito de simplificar sus tareas (AWS, 2023)

**Tabla 10.**  
Entornos de desarrollo para python

Característica	PyCharm	Jupyter Notebook	Google Colab
<b>Tipo</b>	IDE de escritorio	Aplicación web interactiva	Entorno basado en la nube
<b>Acceso a recursos de hardware</b>	Instalación local	Instalación local	Acceso gratuito a una máquina virtual con GPU
<b>Entorno de desarrollo</b>	Instalación local	Instalación local	Funcionamiento en línea
<b>Función es de desarrollo avanzadas</b>	Ofrece una amplia gama de herramientas de desarrollo, como depuración, pruebas unitarias, análisis de código, refactorización, etc.	Proporciona herramientas de desarrollo, aunque puede requerir complementos adicionales.	Ofrece herramientas básicas de desarrollo, pero está más enfocado en la ejecución de cuadernos Jupyter
<b>Bibliotecas para visión por computadora</b>	Permite instalar y gestionar bibliotecas utilizadas en visión por computadora	Permite instalar y gestionar bibliotecas utilizadas en visión por computadora	Permite instalar bibliotecas, pero puede requerir configuración adicional para el uso de recursos de GPU.

<b>Integración con servicios de nube</b>	Capaz de utilizar servicios en la nube y acceder a recursos de GPU en la nube.	Se integra con servicios en la nube, aunque requiere configuración adicional.	Se integra con servicios de Google Cloud.
--	--	---	---

**Fuente:** (Clarke, 2023; Nick, 2020; SourceForge, n.d.)

### 4.9.3. Etiquetado de imágenes

El proceso de etiquetado de imágenes describe las características específicas de los objetos dentro de esa imagen, generalmente que contienen información sobre el tamaño de la imagen y los cuadros delimitadores de los objetos con sus respectivas coordenadas, las cuales contienen posiciones de los bordes izquierdo, superior, derecho e inferior de un cuadro delimitador que encierra el objeto (Brownlee, 2019, Pág. 409)

**Tabla 11.**

Aplicaciones para etiquetado de imágenes

Característica	LabelImg	Roboflow	Hasty
<b>Tipo de software</b>	Aplicación de escritorio	Plataforma en línea	Plataforma en línea
<b>Costo</b>	Gratuito	Plan gratuito, con opciones de pago	Plan gratuito, con opciones de pago
<b>Soporte de formatos de anotación</b>	XML, YOLO, CSV, Pascal VOC	COCO, PASCAL VOC, YOLO, Labelbox, JSON, CSV	COCO, Pascal VOC, entre otros
<b>Etiquetado de objetos</b>	Permite dibujar cajas delimitadoras alrededor de objetos y asignar etiquetas	Herramientas de etiquetado automático para cuadro delimitadores y polígonos inteligentes.	Admite etiquetado de objetos con cajas delimitadoras, polígonos, y etiquetas de texto. Además, tiene asistentes para generar etiquetas sugeridas.
<b>Almacenamiento de datos</b>	Los proyectos deben guardarse localmente o en sistemas de almacenamiento propios.	Almacenamiento en la nube y escalable	Ofrece almacenamiento en la nube y la posibilidad de utilizar almacenamiento propio.
<b>Uso</b>	Fácil de usar, pero requiere instalación en el equipo local	Interfaz de usuario sencilla e intuitiva	Interfaz de usuario amigable y fácil de usar en un entorno web.

**Fuente:** (Lauren, 2022; Roboflow, n.d.)

### 4.9.4. Almacenamiento de resultados

De acuerdo con (Vélez, 2023) un gestor de datos es un programa informático utilizado para crear, actualizar y gestionar bases de datos. Existen numerosos gestores de bases de datos, tanto comerciales como de código abierto.

**Tabla 12.**  
Gestores de base de datos

Característica	Firestore	MongoDB	PostgreSQL
<b>Tipo de base de datos</b>	Post-relacional	Post-relacional	Relacional
<b>Escalabilidad</b>	Escalabilidad horizontal sencilla	Escalabilidad horizontal sencilla	Escalabilidad horizontal con configuración más compleja
<b>Modelo de datos</b>	Almacena datos en formato JSON	Almacena datos en formato BSON	Almacena datos en tablas
<b>Escritura/lectura de imágenes</b>	Puede almacenar imágenes y otros tipos de datos en Firebase Cloud Storage	Puede almacenar imágenes como datos binarios en colecciones	Puede almacenar imágenes como datos binarios en columnas
<b>Seguridad y autenticación</b>	Ofrece opciones de autenticación y reglas de seguridad	Proporciona opciones de autenticación y roles de usuario	Admite políticas de seguridad personalizadas y roles de usuario
<b>Compatibilidad con lenguajes</b>	Compatible con varios lenguajes, incluyendo Python	Compatible con varios lenguajes, incluyendo Python	Ampliamente compatible con varios lenguajes, incluyendo Python

**Fuente:** (Arkhipov, 2023; DB Engines, 2023)

## 4.10. Elementos de hardware

### 4.10.1. Cámara

Como señala (González et al., 2006):

La cámara es el dispositivo que, utilizando un objetivo formado por un juego de lentes y el diafragma, construye una imagen sobre el plano del sensor compuesto de elementos fotosensibles, la digitaliza y la transmite hacia la tarjeta de adquisición del procesador. Están compuestas por un sensor y la electrónica asociada.

Las cámaras proporcionan una señal de vídeo en un formato estándar para su digitalización (en el caso analógico) o directamente la información en formato digital que constituye la imagen captada por la misma (en el caso de cámaras digitales).

**Tabla 13.**  
Cámaras

Característica	DS-2CD1047G0-L(UF)	DS-2CE16H0T-ITF(C)	DS-2CE76D0T-ITMF(C)
<b>Modelo</b>	4 MP	5 MP	2 MP
<b>Tipo de montaje</b>	Bala Fija ColorVu	Mini Bullet Fija	Torreta Fija
<b>Tipo de Cámara</b>	IP (Red)	Analógica	Analógica
<b>Protección</b>	IP67 (IEC 60529-2013)	IP67	IP67
<b>Max. Resolución</b>	2560 × 1440	2560 × 1440	1920 × 1080
<b>Iluminación mínima</b>	Color: 0.01 Lux @ (F2.0, AGC ON), 0 Lux with IR	Color: 0.001 Lux @ (F1.0, AGC ON), 0 Lux with white light	0.01 Lux @ (F2.0, AGC ON), 0 Lux with IR

<b>Alcance</b>	Superior a 30 m	30 m	20 m
<b>Video Bit Rate</b>	32 Kbps a 8 Mbps	4 Mbps a 10 Mbps	2 Mbps a 6 Mbps
<b>Compresión de Video</b>	H.264+ / H.265+	H.264+ / H.265+	H.264+ / H.265+
<b>Fuente de alimentación</b>	12 VDC ± 25% PoE: 802.3af, Class 3	12 VDC ±25%	12 VDC ±25%
<b>Costo</b>	Medio	Alto	Bajo
<b>Aplicaciones</b>	Tecnología colorvu para imágenes a color en condiciones de poca luz.	Seguridad estándar.	Seguridad de calidad estándar.
<b>Presentación</b>			

Fuente: (Hikvision, n.d.)

#### 4.10.2. Dispositivo de grabación

Los dispositivos de grabación deben ser servidores especializados, equipados con Fuente de alimentación y tarjetas de red redundantes para garantizar una disponibilidad elevada (Stolojescu-Crisan et al., 2022).

**Tabla 14.**  
Almacenamiento para cámara

Característica	NVR (Network Video Recorder)	DVR (Digital Video Recorder)
<b>Tipo de Cámaras Compatibles</b>	Cámaras IP (cámaras de red)	Cámaras analógicas
<b>Calidad de Video</b>	Mayor resolución y calidad de video	Calidad de video generalmente más baja.
<b>Conexión de Cámaras</b>	Utiliza una red Ethernet para conectar cámaras IP a través de un enrutador o switch.	Conexión analógica a través de cables coaxiales o BNC.
<b>Almacenamiento</b>	Mayor capacidad de almacenamiento y flexibilidad. Puede usar discos duros internos o externos, almacenamiento en red (NAS), o almacenamiento en la nube.	Generalmente utiliza discos duros internos para almacenamiento local. Almacenamiento en red limitado.
<b>Calidad de Grabación</b>	Grabación de video de alta calidad con capacidad para múltiples resoluciones y tasas de fotogramas.	Calidad de grabación limitada por la resolución de las cámaras analógicas.
<b>Función alidades Avanzadas</b>	Soporte para análisis de video, detección de movimiento avanzada, búsqueda inteligente, acceso remoto, notificaciones y más.	Función es limitadas en comparación con los NVR, generalmente Función es básicas de grabación y reproducción.
<b>Compatibilidad con Aplicaciones Móviles</b>	Aplicaciones móviles avanzadas para acceder y	Aplicaciones móviles más simples y menos Función es.

---

<b>Costo</b>	gestionar cámaras desde dispositivos móviles. Generalmente más caro debido a la mayor capacidad de procesamiento y características avanzadas.	Generalmente más económico, especialmente para sistemas más pequeños con cámaras analógicas.
--------------	--	--

---

**Fuente:** (Stolojescu-Crisan et al., 2022)

## 5. Metodología

### 5.1. Área de Estudio

El escenario elegido para el desarrollo de este proyecto de investigación corresponde a la ciudad de Loja, que forma parte del cantón y provincia homónimos. Esta elección se basa en la identificación de las vías principales que experimentan una alta afluencia de tráfico vehicular, lo cual es un entorno propicio para la implementación del modelo de detección propuesto.

De acuerdo con (Municipio de Loja, 2022) es una de las ciudades más antiguas de Ecuador, fundada por el mariscal de Campo Alonso de Mercadillo en 1548, quien le dio el nombre de su ciudad natal en España. La ciudad de Loja, ubicada en la región de la Sierra al sur de Ecuador, es la capital de la provincia de Loja, que se encuentra en la misma región. Geográficamente, se sitúa a una latitud de aproximadamente 3.9932 grados sur y a una longitud de alrededor de 79.2042 grados oeste.

### 5.2. Materiales y métodos

#### 5.2.1. Modelo de entrenamiento

Todos los modelos descritos en la **Tabla 1**, están basados en Redes Neuronales Convolucionales (CNN). De acuerdo con (Ghosh et al., 2019) las CNN, también conocidas como ConvNets, son un tipo de Red Neuronal Artificial (ANN) con una arquitectura de alimentación profunda que destaca por su capacidad para generalizar e identificar eficazmente características muy abstractas en datos espaciales, especialmente imágenes. En este modelo, las capas iniciales extraen características de bajo nivel, mientras que las capas más profundas manejan características de alto nivel. Este modelo consta de capas de procesamiento que aprenden características a varios niveles de abstracción.

El modelo seleccionado para este sistema es YOLO (You Only Look Once), debido a que presenta una alta precisión para reconocer una gran variedad de objetos, además permite detección en tiempo real, por lo que es el adecuado para el implementarlo en el sistema de detección de invasión de carril.

La versión seleccionada es la última actualización, versión 8, la cual se basa en los últimos avances en aprendizaje profundo y visión por ordenador, y ofrece un rendimiento sin precedentes en términos de velocidad y precisión. Según (Jocher & Chaurasia, 2024) YOLOv8 admite una gama completa de tareas de IA de visión, como la detección, la segmentación, la estimación de la pose, el seguimiento y la clasificación.

#### 5.2.2. Software

**Lenguaje de programación.** En base a la **Tabla 9**, para el desarrollo de este proyecto se considera lenguaje de programación Python debido a sus ventajas y aplicaciones altamente

pertinentes para las aplicaciones de visión por computadora. La elección de Python se respalda por su amplia gama de bibliotecas especializadas como OpenCV, que facilitarán significativamente la implementación de algoritmos de detección de objetos. Además, la disponibilidad de recursos en línea y tutoriales hará que el proceso de desarrollo sea más eficiente, lo que en última instancia contribuirá al éxito del proyecto (Toro Bonilla, 2022).

**Entorno de desarrollo.** Según la **Tabla 10**, PyCharm destaca como la mejor opción en porque es un IDE para el desarrollo de software que ofrece una amplia gama de herramientas de desarrollo enfocadas en el desarrollo de sistemas de visión por computadora como: acceso a recursos locales, gestión amplia bibliotecas y flexibilidad para integrarse con servicios en la nube. Por lo tanto, ha sido seleccionado como el entorno de desarrollo ya que es el que mejor se acopla a las necesidades del presente proyecto. Además, como recurso adicional se utiliza Google Colab debido a que ofrece altas capacidades de procesamiento (Clarke, 2023; Nick, 2020).

**Etiquetado de imágenes.** Se ha tomado en cuenta la plataforma Roboflow, puesto que de acuerdo a la **Tabla 11**, se puede destacar varias características importantes, que incluyen un etiquetado de objetos versátil, compatibilidad con varios formatos de anotación, entre ellos se encuentra el formato para el entrenamiento del modelo YOLO, almacenamiento en la nube y una interfaz de usuario intuitiva en un entorno web. Adicionalmente, dispone de asistentes que generan etiquetas sugeridas en función de modelos preentreados con el conjunto de datos COCO, agilizando de esta manera el proceso de anotación. Por otro lado, es importante destacar que esta plataforma (Roboflow, n.d.).

**Almacenamiento para resultados.** De acuerdo con las características de la **Tabla 12**, se ha decidido que los resultados obtenidos serán almacenados en Firebase debido a que ofrece una solución integral que abarca tanto el almacenamiento de imágenes como el de la base de datos (DB Engines, 2023).

### **5.2.3. Hardware**

**Cámara.** Tomando en cuenta las características de la **Tabla 13**. Se seleccionó la cámara DS-2CD1047G0-L(UF) debido a que su tecnología permite una transmisión de alta calidad, por lo que es ideal para la implementación de este proyecto. Otra característica importante que se ha tomado en cuenta es la clasificación IP67 para resistencia a la intemperie debido a que, al ser instalada en un sitio externo está propensa al sol, lluvia y otros factores medioambientales (Hikvision, n.d.).

**Dispositivo de grabación.** La **Tabla 14**, muestra dos dispositivos utilizados para la recopilación de datos de una cámara de la marca Hikvision. La elección de un NVR (Network

Video Recorder) en lugar de un DVR (Digital Video Recorder) se justifica por varias razones fundamentales. Primero, el NVR es compatible con cámaras IP, por tanto, es adecuado para la cámara seleccionada en la sección anterior. Además, permite la generación de videos e imágenes de alta resolución, que son apropiados para la creación del dataset que entrenará el modelo de detección. Aunque los NVR tienen un costo elevado, su capacidad para aprovechar al máximo las capacidades de las cámaras IP los hace ideales para aplicaciones que requieren alta calidad y funcionalidad (Stolojescu-Crisan et al., 2022).

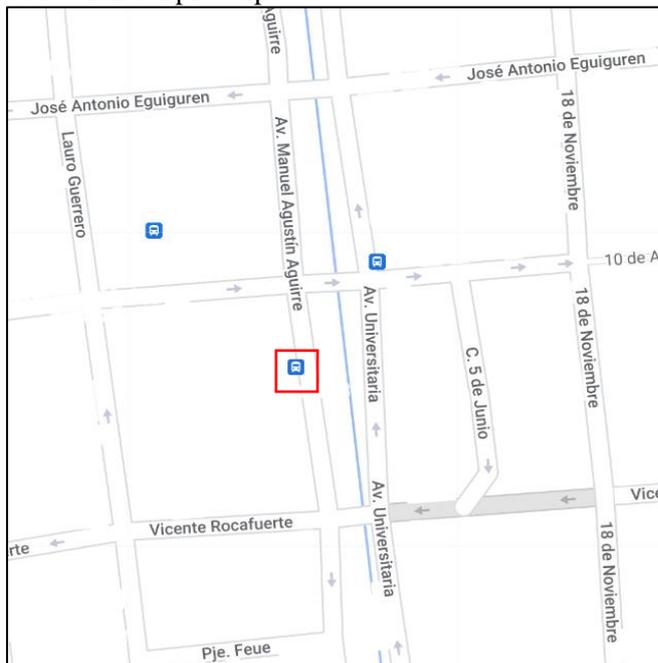
### 5.3. Diseño del Sistema

#### 5.2.4. Implementación del sistema

La implementación del prototipo se lleva a cabo tras la obtención de la autorización respectiva por parte del Gobierno Autónomo Descentralizado Municipal del cantón Loja, por medio del Director de la Unidad de Control Operativa de Tránsito (UCOT) y el departamento de la Dirección de Movilidad. Tras una revisión de la propuesta presentada, los encargados asignaron un espacio público considerado apropiado para la instalación de los equipos. Esta elección se basa en la capacidad del lugar no sólo para proporcionar los datos e información requeridos para el desarrollo del proyecto, sino también para brindar seguridad a la zona.

#### Figura 4.

Ubicación del prototipo

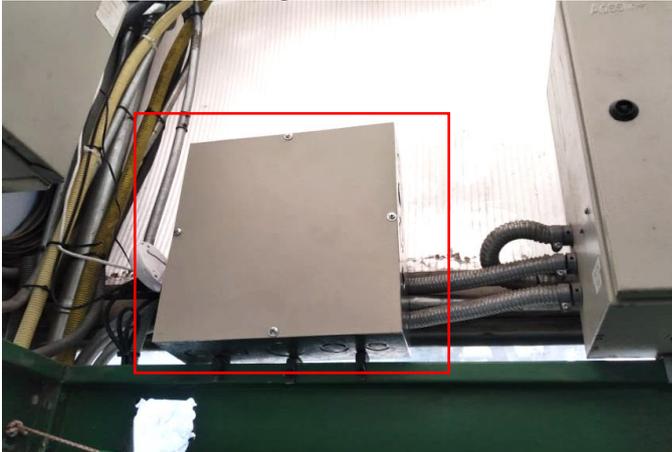


Fuente: Google maps

Es importante mencionar que, este prototipo consta de dos partes, una caja de seguridad destinada a proteger los equipos necesarios (ONT, UPS y roseta óptica) y se encuentra ubicada en la parte interna de la parada.

**Figura 5.**

Parte interna del prototipo



**Fuente:** Elaboración propia

En el exterior del prototipo se encuentra instalada una cámara Hikvision 4 MP ColorVu Fixed Bullet Network de la marca Hikvision, la cual se encarga de recopilar imágenes y videos que sirven para el posterior entrenamiento del modelo de detección.

**Figura 6.**

Parte Externa del prototipo



**Fuente:** Elaboración propia

### **5.2.5. Recopilación de imágenes**

Dentro de esta etapa se implementó el protocolo RTSP (Protocolo de Control de Transmisión en Tiempo Real) para la transmisión de datos. De esta manera se aprovecha la capacidad de transmisión en tiempo real para realizar las pruebas necesarias y facilita el acceso a las transmisiones previamente guardadas desde ubicaciones remotas.

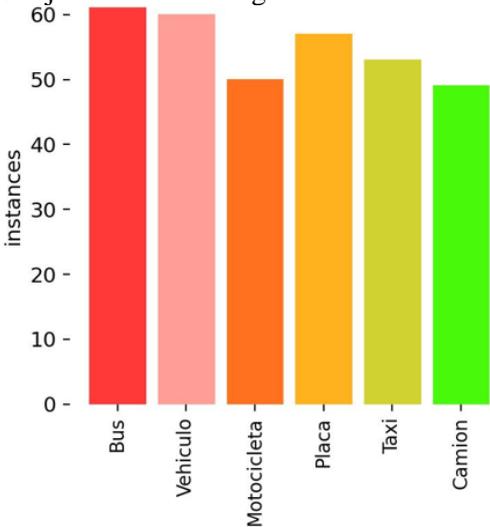
Los datos recopilados fueron almacenados en un NVR (Network Video Recorder), el cual opera de manera remota. Mediante la aplicación Hik-Connect, los vídeos capturados se encuentran disponibles tanto para el desarrollo del proyecto como para el GAD Municipal.

Esta etapa se llevó a cabo de forma continua desde el mes de septiembre hasta el mes de febrero, debido a que el conjunto de datos se actualizó continuamente con la finalidad de abarcar una amplia gama de variaciones. El objetivo es construir una base de datos diversificada que incluya una amplia gama de escenarios y condiciones que reflejen la complejidad del entorno real en donde se va a operar el modelo.

El resultado de esta etapa es la creación de tres conjuntos de datos que permiten distinguir varios objetos, incluyendo vehículos, buses, taxis, camiones, motocicletas y placas vehiculares. Estos conjuntos se dividen en tres categorías según su tamaño: el primero consta de 250 imágenes, el segundo de 500 y el tercero de 1000.

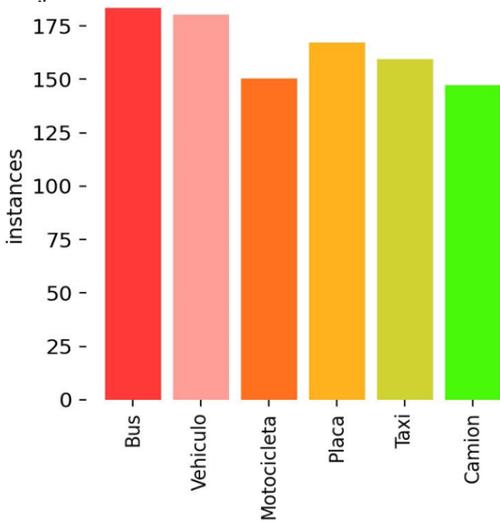
**Representación de objetos del primer conjunto de datos (250 imágenes)**

**Figura 7.**  
Conjunto de datos original



Fuente: Elaboración propia

**Figura 8.**  
Conjunto de datos con aumento de datos

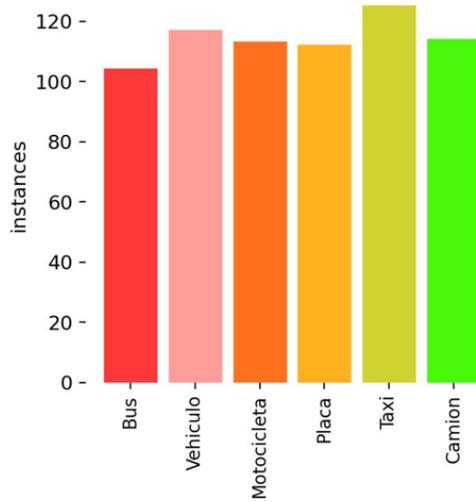


Fuente: Elaboración propia

## Representación de objetos del segundo conjunto de datos (500 imágenes)

**Figura 9.**

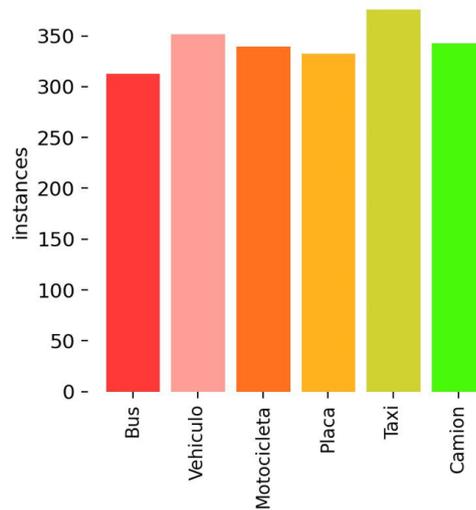
Conjunto de datos original



Fuente: Elaboración propia

**Figura 10.**

Conjunto de datos con aumento de datos

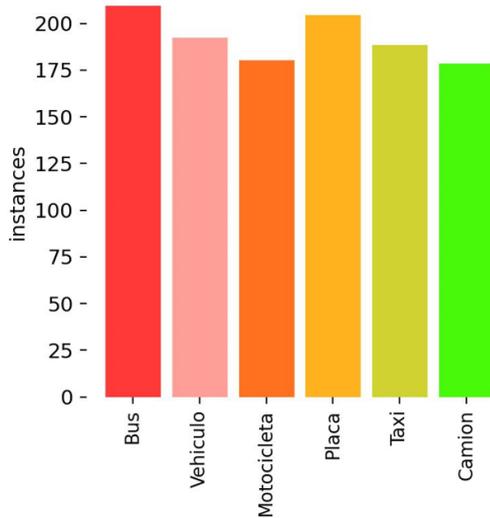


Fuente: Elaboración propia

## Representación de objetos del segundo conjunto de datos (1000 imágenes)

**Figura 11.**

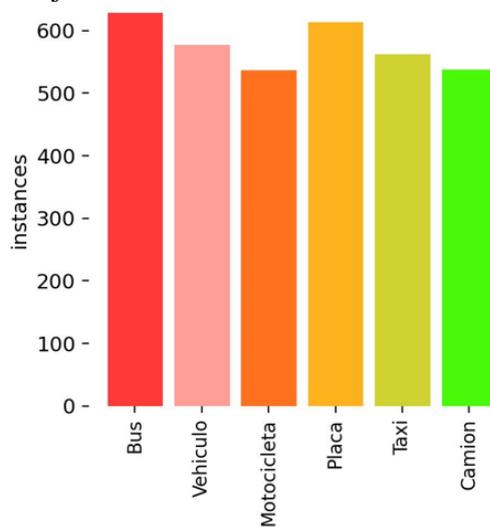
Conjunto de datos original



**Fuente:** Elaboración propia

**Figura 12.**

Conjunto de datos con aumento de datos



**Fuente:** Elaboración propia

### 5.2.6. Preparación de datos

Durante el proceso de preparación de datos, es fundamental llevar a cabo la anotación de las clases, que implica delimitar cada objeto de interés mediante una caja delimitadora. En este contexto, se empleó la plataforma RoboFlow, la cual facilita esta tarea puesto que incorpora una opción de etiquetado automático basada en un modelo pre entrenado del conjunto de datos COCO. Además, ofrece la posibilidad de exportar directamente el conjunto de datos en un

formato adecuado para YOLOv8, generando dos archivos: la imagen correspondiente y un archivo de texto que detalla la clase y ubicación dentro de la imagen.

Además, antes del entrenamiento del modelo, se llevó a cabo un redimensionamiento de imágenes en cada conjunto de datos. Todas las imágenes fueron ajustadas a 640x640 píxeles. Esta estandarización se realizó para cumplir con un requisito fundamental para la implementación del modelo YOLO.

**Figura 13.**

Imagen Original - Redimensionamiento de imágenes



**Fuente:** Elaboración propia

**Figura 14.**

Imagen resultante - Redimensionamiento de imágenes



**Fuente:** Elaboración propia

Adicional a esto, a los conjuntos de datos se les aplico de forma individual técnicas de procesamiento de imágenes mencionadas en **Tabla 2**. De esta manera, se llevó a cabo el proceso de aumento de datos, lo que resultó en un aumento considerable del tamaño del conjunto de datos de entrenamiento, triplicando su tamaño original.

**Zoom.** Implica la ampliación o reducción de la escala de una imagen en un rango que va desde el 0% hasta el 30%. Este ajuste en la escala permite modificar la perspectiva y el nivel

de detalle de la imagen, lo que puede ser útil para mejorar la robustez del modelo frente a variaciones en el tamaño y la posición de los objetos dentro de la imagen.

**Figura 15.**

Imagen original - Técnica de ampliación/reducción



**Fuente:** Elaboración propia

**Figura 16.**

Imagen resultante - Técnica de ampliación/reducción



**Fuente:** Elaboración propia

**Rotación.** Implica girar la imagen en un ángulo que varía entre -15% y 15% con respecto a su posición original. Esta rotación puede ser en sentido horario o antihorario y permite modificar la orientación de los objetos dentro de la imagen. Esto permite una variación controlada en la orientación de la imagen, lo que puede ser útil para mejorar la capacidad del modelo para reconocer objetos desde diferentes perspectivas.

**Figura 17.**

Imagen original - Técnica de rotación



Fuente: Elaboración propia

**Figura 18.**

Imagen resultante - Técnica de ampliación/reducción



(a)



(b)

Nota: (a) Rotación a  $-15^\circ$ , (b) Rotación a  $15^\circ$

Fuente: Elaboración propia

**Saturación:** Implica ajustar el nivel de saturación de una imagen en un rango que va desde -30% hasta 30%. Esto significa que los colores de la imagen pueden ser desaturados en hasta un 30% (reducción en la intensidad de los colores) o saturados en hasta un 30% (aumento en la intensidad de los colores).

**Figura 19.**

Imagen original - Técnica modificación de saturación



Fuente: Elaboración propia

**Figura 20.**

Imagen resultante - Técnica de saturación



(a)



(b)

**Nota:** (a) Saturación a  $-30^\circ$  y (b) Saturación a  $30^\circ$

Fuente: Elaboración propia

**Brillo.** Implica ajustar el nivel de brillo de una imagen en un rango que va desde  $-25\%$  hasta  $25\%$ . Esto significa que la luminosidad de la imagen puede ser disminuida en hasta un  $25\%$  (reducción en la intensidad de la luz) o aumentada en hasta un  $25\%$  (aumento en la intensidad de la luz). Este ajuste permite modificar el nivel de luminosidad de la imagen, lo que puede ser útil para corregir imágenes subexpuestas o sobreexpuestas, así como para mejorar la visibilidad de ciertos detalles en diferentes condiciones de iluminación.

**Figura 21.**

Imagen original – Técnica modificación de brillo



**Fuente:** Elaboración propia

**Figura 22.**

Imagen resultante – Técnica modificación de brillo



(a)



(b)

**Nota:** (a) Brillo a  $-25^\circ$ , (b) Brillo a  $25^\circ$

**Fuente:** Elaboración propia

**Desenfocar.** Implica aplicar un efecto de desenfoque a la imagen, donde el valor del 1.5% indica el grado de desenfoque gaussiano aplicado. Este efecto suaviza los detalles y los bordes de la imagen, lo que puede ser útil para reducir el ruido o para crear efectos visuales más suaves y menos definidos.

**Figura 23.**

Imagen original – Técnica desenfoque



**Fuente:** Elaboración propia

**Figura 24.**

Imagen resultante – Técnica desenfoque



**Fuente:** Elaboración propia

### 5.2.7. *Entrenamiento del modelo*

Para el entrenamiento del modelo de detección de objetos, se utiliza la versión 8 de YOLO, debido a su enfoque especializado en diversas tareas de visión por computadora. En este caso, se seleccionó este modelo ya que presenta una alta eficacia la detección de objetos.

A continuación, se presenta su arquitectura:

**Tabla 15.**

Capas de la arquitectura de YOLOv8

Nro.	Parámetros	Modulo
0	464	Capa de convolución con 16 filtros de 3x3, stride2
1	4672	Capa de convolución con 32 filtros de 3x3, stride 2.
2	7360	Capa de bloque C2f con 32 canales de entrada y salida
3	18560	Capa de convolución con 64 filtros de 3x3, stride 2
4	49664	Capa de bloque C2f con 64 canales de entrada y salida
5	73984	Capa de convolución con 128 filtros de 3x3, stride 2.
6	197632	Capa de bloque C2f con 128 canales de entrada y salida

7	295424	Capa de convolución con 256 filtros de 3x3, stride 2.
8	460288	Capa de bloque C2f con 256 canales de entrada y salida.
9	164608	Capa de factor de escala espacial (SPPF) con 256 canales de entrada y salida.
10	0	Capa de upsampling con método 'nearest'.
11	0	Capa de concatenación.
12	148224	Capa de bloque C2f con 384 canales de entrada y 128 de salida.
13	0	Capa de upsampling con método 'nearest'.
14	0	Capa de concatenación
15	37248	Capa de bloque C2f con 192 canales de entrada y 64 de salida.
16	36992	Capa de convolución con 64 filtros de 3x3, stride 2.
17	0	Capa de concatenación
18	123648	Capa de bloque C2f con 192 canales de entrada y 128 de salida.
19	147712	Capa de convolución con 128 filtros de 3x3, stride 2.
20	0	Capa de concatenación.
21	493056	Capa de bloque C2f con 384 canales de entrada y 256 de salida
22	752482	Capa de detección con 6 clases y canales de entrada [64, 128, 256].

**Fuente:** Elaborada a partir de los entrenamientos del modelo YOLOv8

La arquitectura de YOLOv8 se compone de una serie de capas que conforman su estructura. Comienza con capas de convolución que aplican filtros de diferentes tamaños y desplazamientos para extraer características de la imagen de entrada. Estas capas se alternan con bloques C2f, que son módulos diseñados para mejorar el flujo de gradiente en la red y reducir la complejidad. Además, se utilizan capas de factor de escala espacial (SPPF) para ajustar la escala de las características espaciales. Las capas de aumento de resolución espacial y concatenación se utilizan para aumentar la resolución de las características y fusionar información de diferentes niveles de la red, respectivamente. Finalmente, se emplea una capa de detección que produce las predicciones finales de objetos, con 6 clases y canales de entrada especificados para la detección de diferentes escalas de objetos. En conjunto, estas capas conforman la arquitectura de YOLOv8, que se caracteriza por su eficiencia y precisión en la detección de objetos.

Una vez conocida la arquitectura del modelo a utilizar, se realizó el entrenamiento, para ello se utiliza la plataforma *Google Colaboratory* debido a su alta disponibilidad de recursos computacionales.

### Primer conjunto de datos

**Tabla 16.**

Parámetros para entrenamiento del primer conjunto de datos

<b>Parámetros</b>	<b>Descripción</b>
Modelo	YOLOv8n
Tamaño imágenes	640 píxeles
Clases	Bus, vehículo, motocicleta, placa, taxi y camión
Canales	3
Tamaño del lote	32
Iteraciones	50
Cantidad de imágenes	250 (100%)
Imágenes de entrenamiento	117 (70%)
Imágenes de validación	54 (20%)
Imágenes de verificación	19 (10%)
Aumento de datos	x3
Cantidad de imágenes resultante	604 (117 x 3) +54+19

**Fuente:** Elaboración propia

### Segundo conjunto de datos

**Tabla 17.**

Valores para entrenamiento del segundo conjunto de datos

<b>Parámetros</b>	<b>Descripción</b>
Modelo	YOLOv8n
Tamaño imágenes	640 píxeles
Clases	Bus, vehículo, motocicleta, placa, taxi y camión
Canales	3
Tamaño del lote	64
Iteraciones	100
Cantidad de imágenes	500 (100%)
Imágenes de entrenamiento	355 (70%)
Imágenes de validación	94 (20%)
Imágenes de verificación	51 (10%)
Aumento de datos	x3
Cantidad de imágenes resultante	1210 (355 x 3) +94+51

**Fuente:** Elaboración propia

### Tercer conjunto de datos

**Tabla 18.**

Valores para entrenamiento del tercer conjunto de datos

<b>Parámetros</b>	<b>Descripción</b>
Modelo	YOLOv8n
Tamaño imágenes	640 píxeles
Clases	Bus, vehículo, motocicleta, placa, taxi y camión
Canales	3
Tamaño del lote	128

Iteraciones	200
Cantidad de imágenes	1000 (100%)
Imágenes de entrenamiento	700 (70%)
Imágenes de validación	206 (20%)
Imágenes de verificación	94 (10%)
Aumento de datos	x3
Cantidad de imágenes resultante	2400 (700 x 3) + 206 + 94

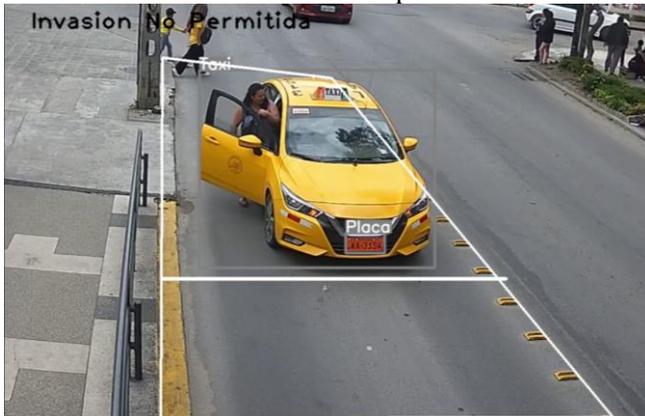
**Fuente:** Elaboración propia

El proceso de verificación de invasión de carril comienza después de obtener los resultados del modelo de detección, tal como se visualiza en la

**Figura 27.** Una vez que se detectan los objetos, estos resultados se pasan a través de un código realizado en lenguaje Python para verificar si alguno está invadiendo un carril. Para ello se utiliza varios componentes y técnicas, como la delimitación de regiones de interés (ROI), determinación de una zona de invasión y la aplicación de una línea de referencia para capturar la imagen del vehículo invasor.

**Figura 25.**

Zona de detección – Invasión no permitida



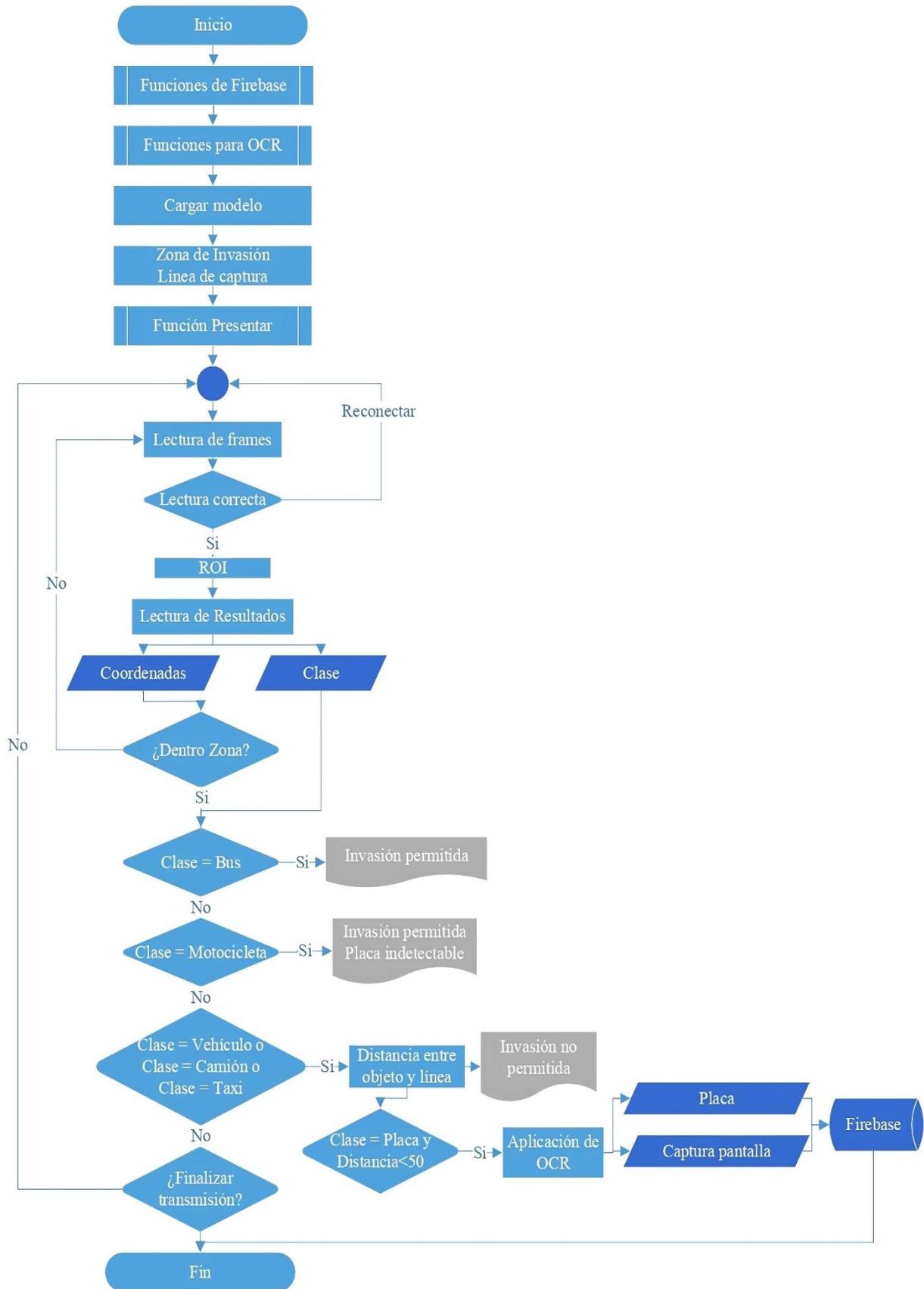
**Fuente:** Elaboración propia

**Figura 26.**  
Zona de detección – Invasión permitida



**Fuente:** Elaboración propia

**Figura 27.**  
Diagrama de flujo del software



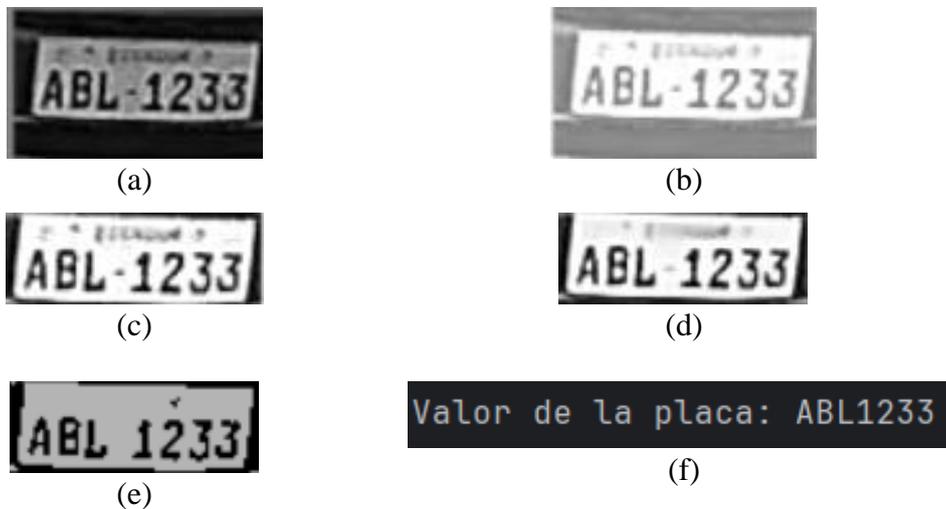
Fuente: Elaboración propia

De esta manera, cuando se detecta un objeto que pertenece a la clase "bus", se clasifica como una invasión permitida, mientras que, si el objeto detectado no pertenece a esta clase, se considera como una invasión no permitida.

Además, se considera importante incorporar la clase "placa" en el proceso de detección de objetos, ya que esta permite determinar cuándo un vehículo está lo suficientemente cerca para capturar una imagen de su placa. Esta imagen será sometida a una serie de técnicas de procesamiento de imagen, detalladas en la **Figura 28**, con el objetivo de optimizar la calidad de la imagen y prepararla adecuadamente para la aplicación del reconocimiento óptico de caracteres (OCR). De esta forma, se puede obtener datos precisos de las placas de vehículos invasores, lo que contribuye significativamente a la efectividad y precisión.

**Figura 28.**

Procesamiento de imágenes para la placa



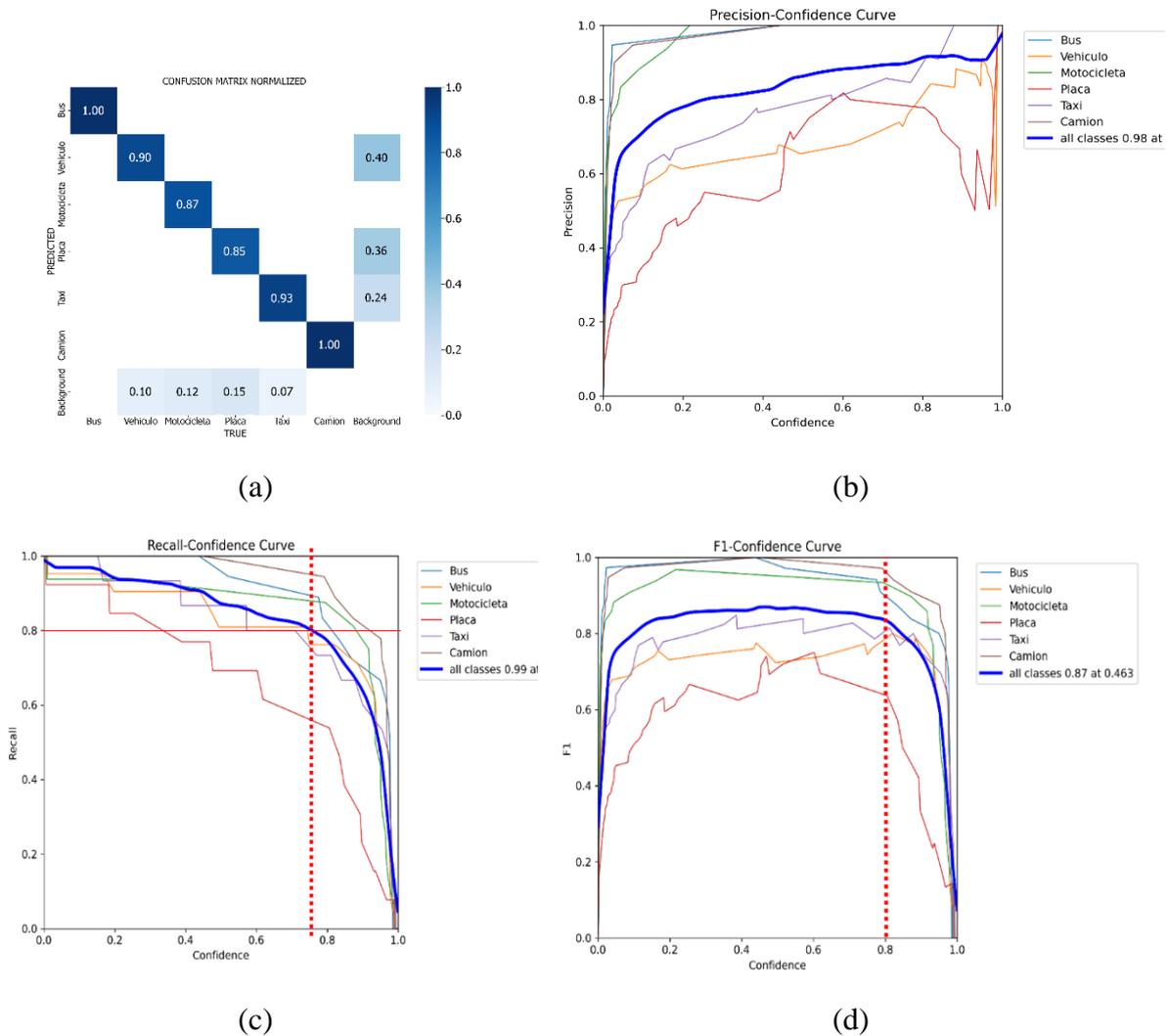
**Nota:** (a) Imagen original, (b) Aplicación de brillo, (c) Normalización, (d) Eliminación de ruido, (e) Umbralización, (f) Resultado del OCR

**Fuente:** Elaboración propia

### 5.2.8. Evaluación

Las métricas de evaluación utilizadas para determinar la precisión y eficiencia del modelo de detección mencionadas en la **Tabla 7**; las cuales incluyen, la matriz de confusión, curva de precisión, recuperación (Recall) y F1 Score, se tomaron en cuenta para medir la precisión de los modelos implementados en los tres conjuntos de datos, de los cuales se obtuvo los siguientes resultados.

**Figura 29.**  
Evaluación del primer conjunto de datos



**Nota:** (a) Matriz de confusión, (b) Curva de precisión, (c) Curva de Recall y (d) Curva de Puntuación F1

Según la matriz de confusión, se observa un alto nivel de clasificación correcta. Esta conclusión se respalda al notar que la diagonal principal contiene la mayoría de los valores, los cuales alcanzan porcentajes desde el 87% hasta el 100%. Estos porcentajes indican las

instancias en las que las etiquetas predichas por el modelo coinciden exactamente con las etiquetas reales.

Sin embargo, esto no implica que este modelo sea idóneo para su implementación, debido a que el conjunto de datos no está balanceado completamente, por lo que es necesario implementar otras métricas, además de un umbral o punto de equilibrio en donde el modelo alcanza un alto nivel de precisión. En la gráfica (b) se puede evidenciar la precisión, donde algunas líneas que representan a las clases muestran irregularidades, indicando así que este modelo presenta ciertas deficiencias, especialmente en las clases de placa, vehículo y taxi. En este sentido el umbral o punto de equilibrio se logra con un 85%, hasta este punto se logra realizar predicciones precisas, luego la curva empieza a descender un tramo lo que significa que la precisión del modelo tiende a disminuir.

La curva de Recall representada en la gráfica (c) muestra que, al principio, con una confianza baja, el modelo logra predecir correctamente la mayoría de los casos positivos (alto recall). Sin embargo, a medida que aumenta la confianza, la Recall comienza a disminuir, indicando que el modelo está perdiendo algunos casos positivos. En este caso, el umbral puede establecerse en un 79% de confianza, ya que las clases mantienen un alto nivel de Recall en este punto.

La curva F1 representa la calidad y equilibrio de las predicciones pueden variar dependiendo del nivel de confianza, la gráfica (d) muestra que en un principio los valores de F1 tienden a aumentar, alcanzando un equilibrio decente aproximadamente con un 80% de confianza. Este punto indica el nivel de confianza óptimo para las predicciones del modelo.

**Tabla 19.**  
Resultados de métricas de primer conjunto de datos

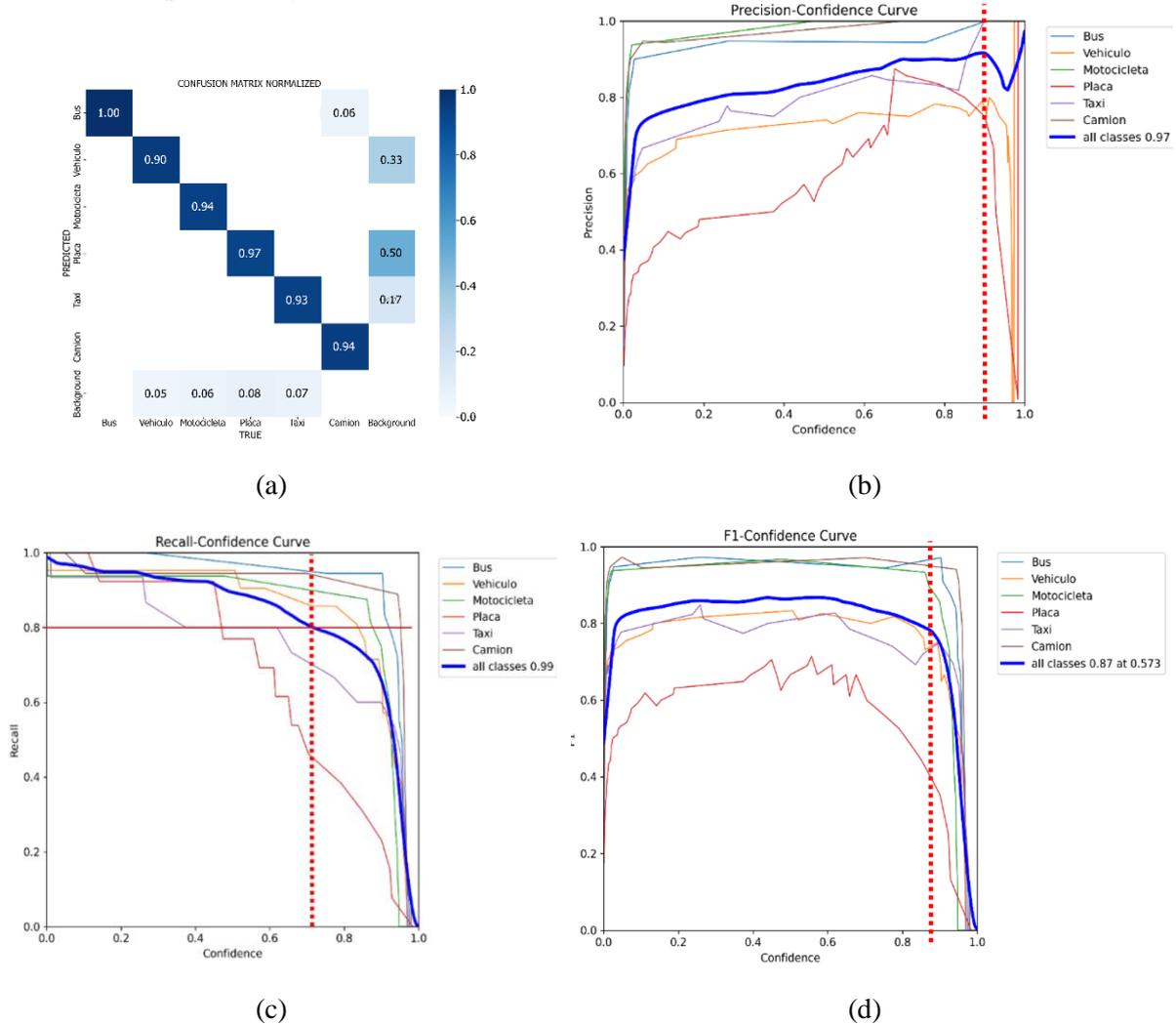
<b>Métrica</b>	<b>Punto de equilibrio</b>
Precisión	85% de confianza
Recall	79% de confianza
Puntuación F1	80% confianza

**Fuente:** Elaboración propia

## Primer conjunto de datos con aumento de datos

**Figura 30.**

Evaluación del primer conjunto de datos con aumento de datos



**Nota:** (a) Matriz de confusión, (b) Curva de precisión, (c) Curva de Recall y (d) Curva de Puntuación

F1

**Tabla 20.**

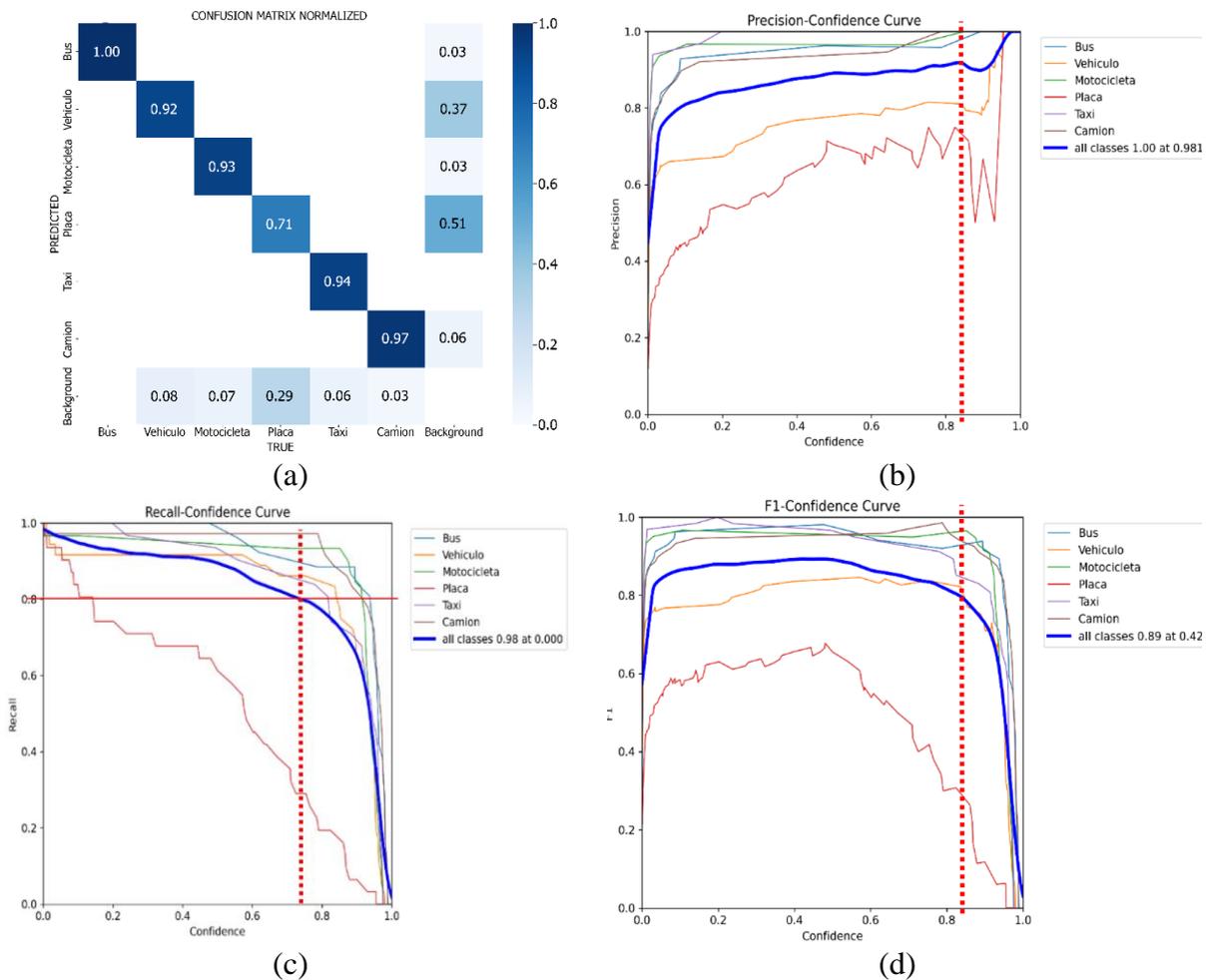
Resultados de métricas del primer conjunto con aumento de datos

Métrica	Punto de equilibrio
Precisión	90% de confianza
Recall	72% de confianza
Puntuación F1	83% confianza

**Fuente:** Elaboración propia

## Segundo conjunto de datos

**Figura 31.**  
Evaluación del segundo conjunto



**Nota:** (a) Matriz de confusión, (b) Curva de precisión, (c) Curva de Recall y (d) Curva de Puntuación

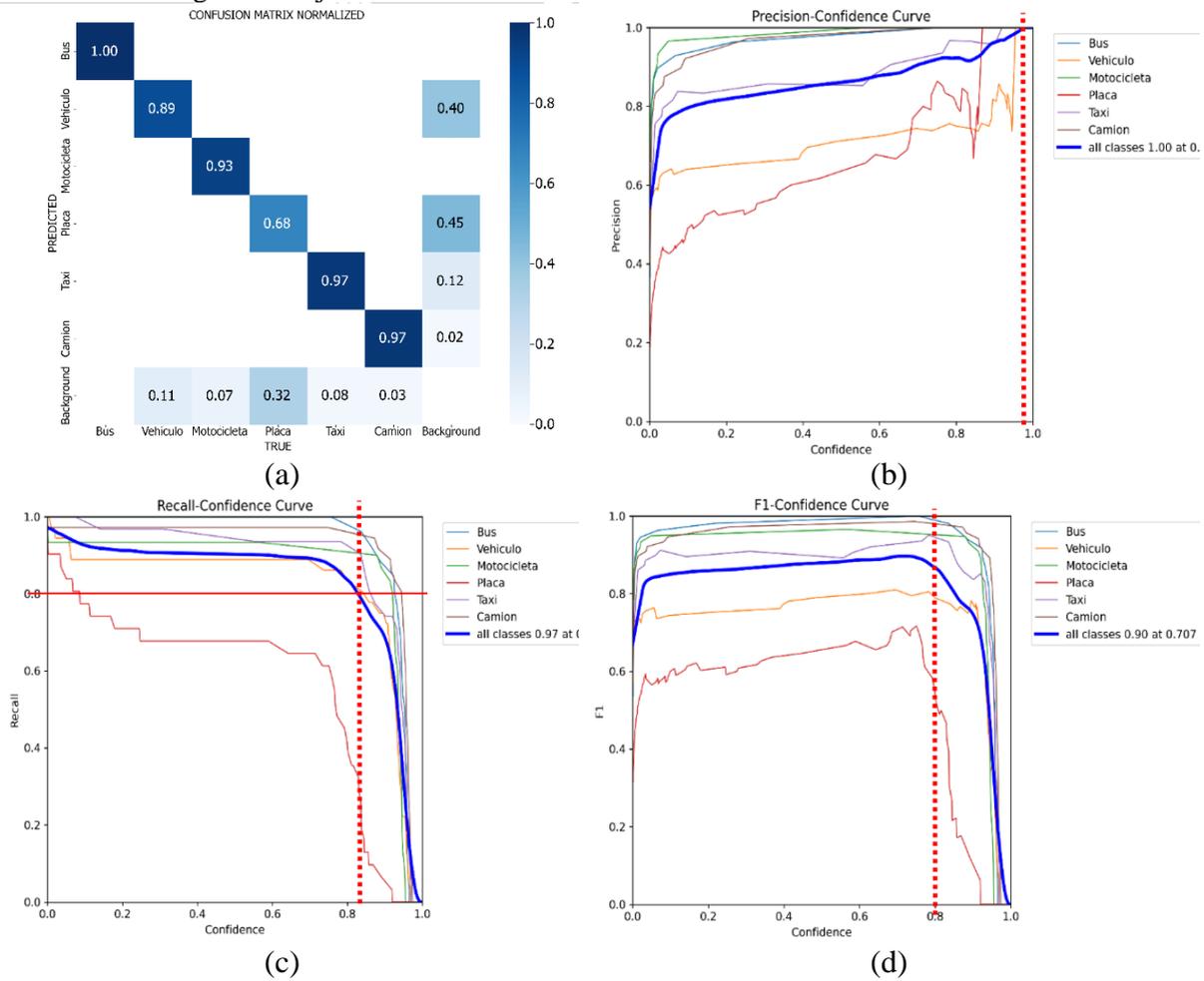
F1

**Tabla 21.**  
Resultados de métricas del segundo conjunto de datos

Métrica	Punto de equilibrio
Precisión	82% de confianza
Recall	78% de confianza
Puntuación F1	82% confianza

**Fuente:** Elaboración propia

**Figura 32.**  
Evaluación del segundo conjunto con aumento de datos



**Nota:** (a) Matriz de confusión, (b) Curva de precisión, (c) Curva de Recall y (d) Curva de Puntuación

F1

**Tabla 22.**

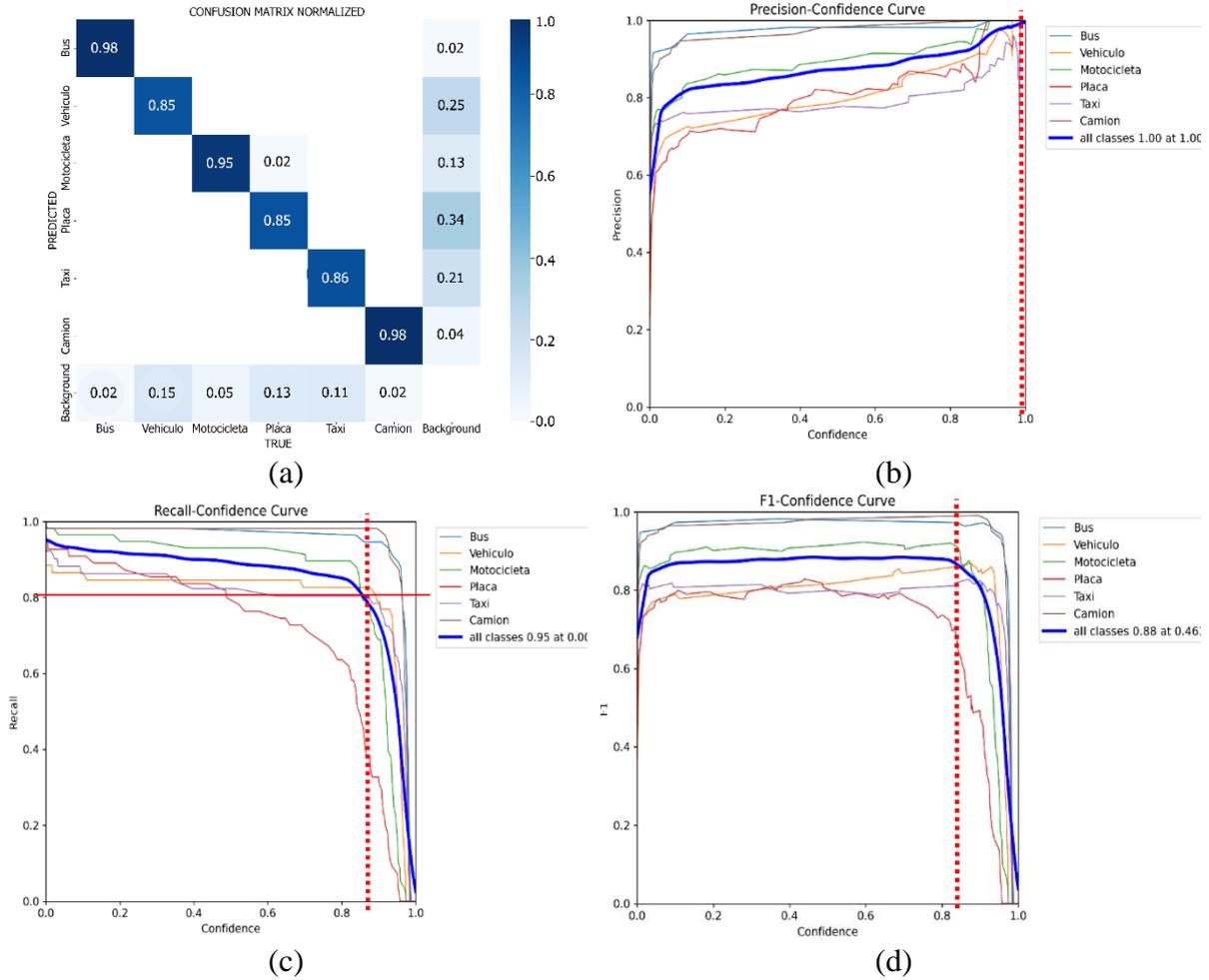
Resultados de métricas del segundo conjunto con aumento de datos

Métrica	Punto de equilibrio
Precisión	99% de confianza
Recall	Alcanzo predicción perfecta
Puntuación F1	81% confianza

**Fuente:** Elaboración propia

## Tercer conjunto de datos

**Figura 33.**  
Evaluación del tercer conjunto de datos



**Nota:** (a) Matriz de confusión, (b) Curva de precisión, (c) Curva de Recall y (d) Curva de Puntuación

F1

**Tabla 23.**

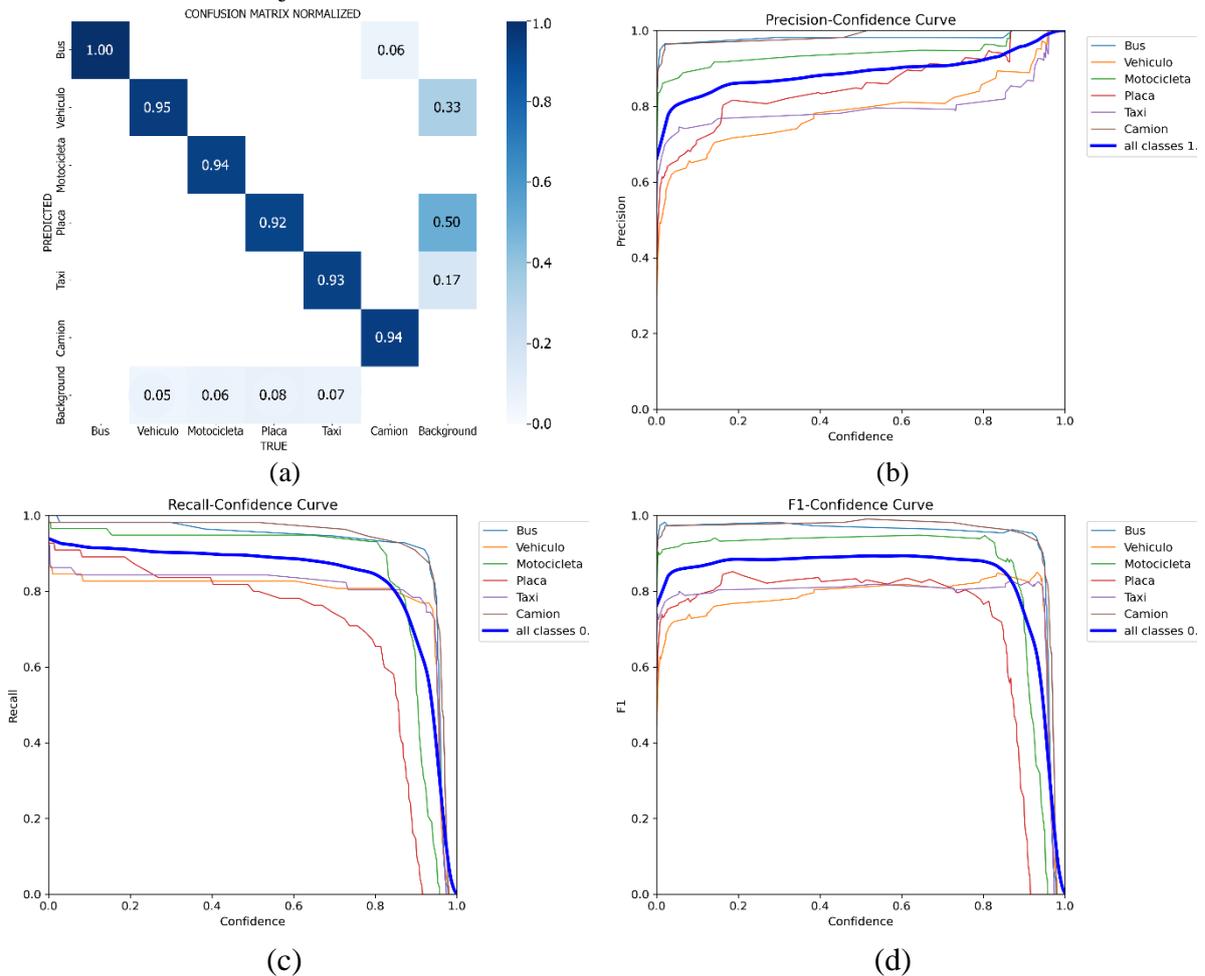
Resultados de métricas del tercer conjunto de datos

Métrica	Punto de equilibrio
Precisión	100% de confianza
Recall	Alcanzo predicción perfecta
Puntuación F1	85% de confianza

**Fuente:** Elaboración propia

## Tercer conjunto con aumento de datos

**Figura 34.**  
Evaluación del tercer conjunto de datos



**Nota:** (a) Matriz de confusión, (b) Curva de precisión, (c) Curva de Recall y (d) Curva de Puntuación

F1

**Tabla 24.**

Resultados de métricas del tercer conjunto de datos

Métrica	Punto de equilibrio
Precisión	98% de confianza
Recall	Alcanzo predicción perfecta
Puntuación F1	93% confianza

**Fuente:** Elaboración propia

## 6. Resultados

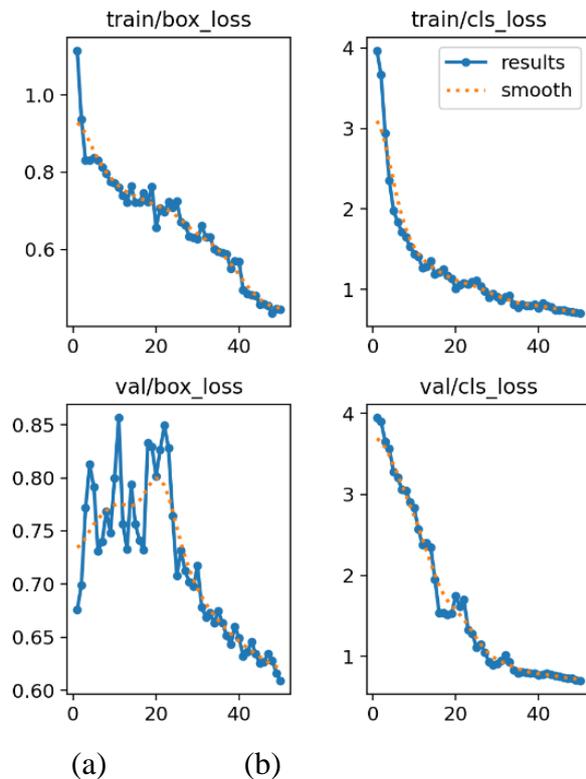
Los resultados obtenidos al entrenar el modelo YOLOv8 con conjuntos de datos de diferentes tamaños revelaron importantes variaciones en su rendimiento, lo que destaca la importancia de que el tamaño del conjunto de datos afecta la capacidad predictiva y precisión del modelo. Por ello, es importante seleccionar cuidadosamente el tamaño adecuado del conjunto de datos para entrenar modelos de detección de objetos, ya que un tamaño inadecuado puede conducir a resultados deficientes en términos de detección y precisión. A continuación, se presenta un análisis de los resultados obtenidos en cada conjunto de datos de diferentes tamaños:

### 6.1. Resultados para primer conjunto de datos

El entrenamiento se realizó con los siguientes parámetros: se utilizó un tamaño de lote de 32 durante 50 iteraciones. Se dispuso de un total de 250 imágenes, donde el 70% (117 imágenes) se utilizaron para el conjunto de entrenamiento, el 20% (54 imágenes) para la validación, y el 10% restante (19 imágenes) para la verificación del modelo.

**Figura 35.**

Perdidas de caja delimitadora y clases primer conjunto



**Nota:** (a) Perdidas de caja delimitadora y (b) Perdidas de clases

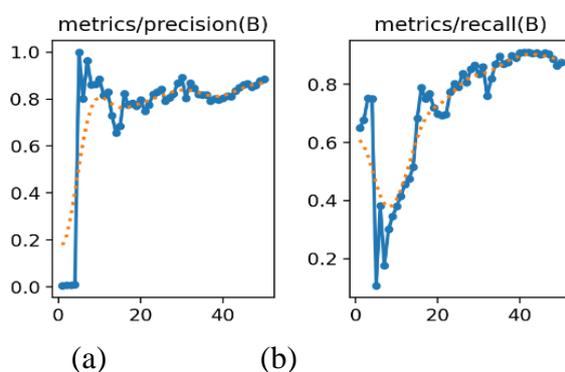
En la gráfica (a), se puede observar el comportamiento de las pérdidas de la caja delimitadora tanto durante el proceso de entrenamiento como en el de validación. Las pérdidas durante el entrenamiento se calculan mediante la comparación de las coordenadas de las cajas

delimitadoras predichas por el modelo con las coordenadas reales en el conjunto de datos. Para el caso del entrenamiento, se puede notar que, a pesar de algunas fluctuaciones, el valor de las pérdidas desciende de manera consistente a lo largo de las 50 iteraciones de aprendizaje. Esto indica que el modelo está mejorando su capacidad para localizar con mayor precisión los objetos en las imágenes.

Por otro lado, las pérdidas de validación muestran una mayor variabilidad. En la gráfica, se pueden identificar varios picos de pérdidas que son más pronunciados que los observados durante el entrenamiento además la curva está por encima de la de entrenamiento. Esto podría sugerir que el modelo está experimentando sobreajuste<sup>1</sup>.

En el gráfico (b), se representan las pérdidas de clasificación de clases, las cuales se calculan mediante la comparación entre las probabilidades de clase predichas por el modelo y las clases reales de los objetos en el conjunto de datos. Las pérdidas de entrenamiento muestran una tendencia descendente constante a lo largo del proceso, aunque se observan algunas fluctuaciones. Al finalizar las 50 iteraciones de aprendizaje, el modelo registra una disminución en la cantidad de pérdidas. En contraste, las pérdidas de validación exhiben una mayor variabilidad, y se evidencia que la curva se encuentra por encima de la de entrenamiento. Por consiguiente, se puede inferir que este modelo tiene sobreajuste.

**Figura 36.**  
Métricas de evaluación generales primer conjunto



**Nota:** (a) Precisión, (b) Recall

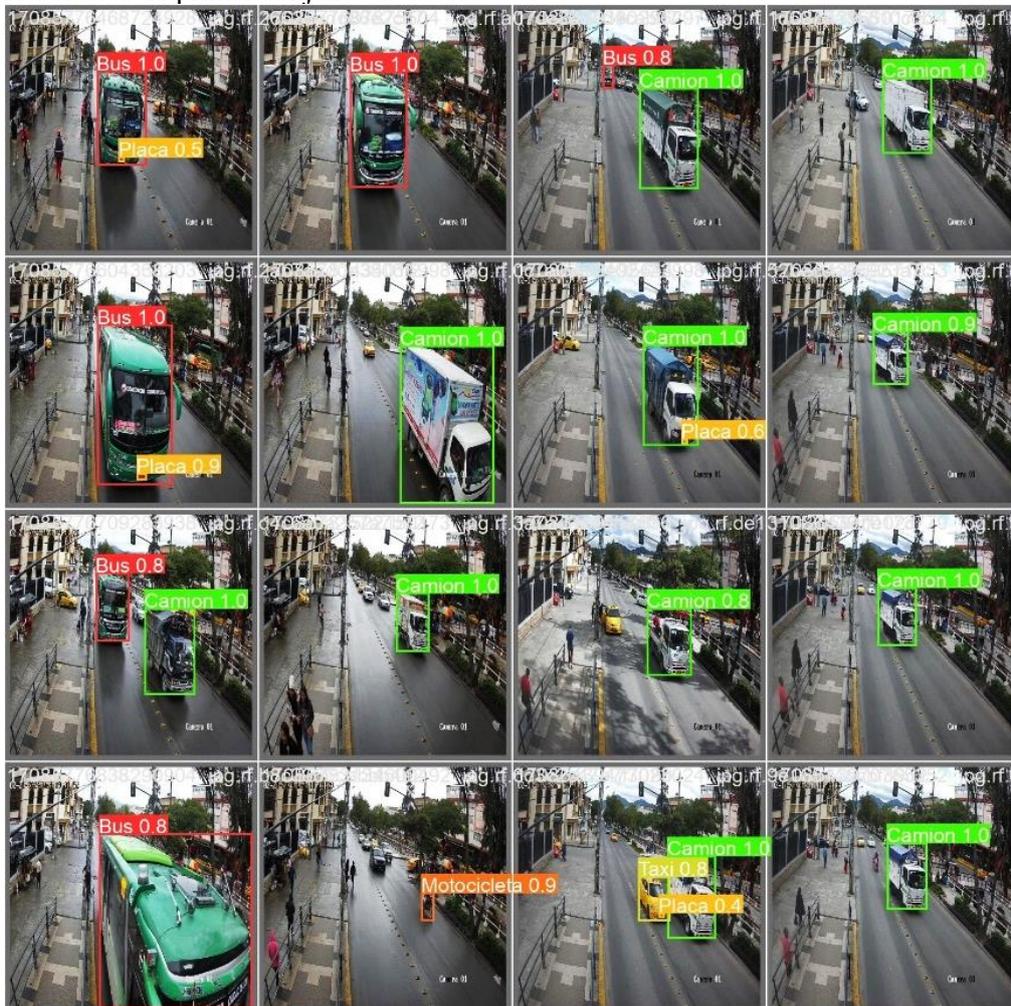
---

<sup>1</sup> **Sobreajuste:** las pérdidas en el conjunto de entrenamiento son muy bajas pero las pérdidas en el conjunto de validación son más altas.

El primer gráfico (a) corresponde a la precisión con respecto a la cantidad de iteraciones de entrenamiento. Se observa que la curva es ascendente lo que significa que el modelo está haciendo una mayor cantidad de predicciones correctas, sin embargo, también existen ciertas fluctuaciones durante las primeras 25 interacciones lo que significa que el rendimiento del modelo no es el óptimo, esto puede estar relacionado con el tamaño del conjunto de datos al trabajar con un conjunto de datos pequeño existe una mayor variabilidad del rendimiento.

El gráfico (b) representa el recall (*Recall*) es decir, es capacidad del modelo para capturar la mayoría de los casos positivos. En este caso, la curva inicia con un valor elevado, pero en las siguientes iteraciones decae, recuperándose considerablemente ascendiendo de forma consistente. Estas irregularidades en la curva podrían representar un sobreajuste a los datos de entrenamiento.

**Figura 37.**  
Resultados de primer conjunto de datos



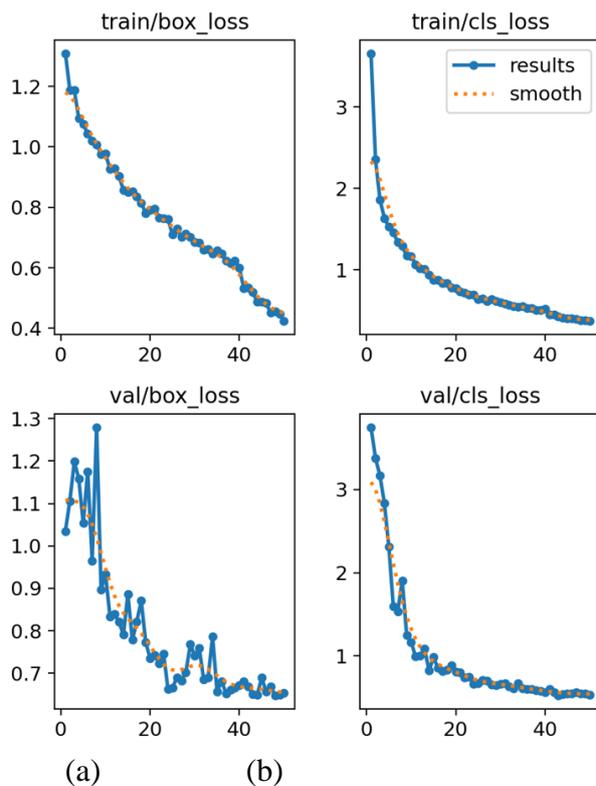
Fuente: Elaboración propia

En la **Figura 37** se puede observar que este modelo, si bien logra algunos aciertos con porcentajes altos de confiabilidad, aún presenta detección de objetos con un porcentaje demasiado bajo. Esto significa que, aunque el modelo puede realizar predicciones precisas en algunos casos, también puede cometer errores significativos o tener dificultades para identificar correctamente ciertos objetos en las imágenes.

## 6.2. Resultados para primer conjunto con aumento de datos

El entrenamiento de este conjunto de datos se realizó con los mismos parámetros del modelo anterior. Además, se aplicaron técnicas de aumento de datos a conjunto de entrenamiento las cuales incluyeron zoom en un rango del 0 al 30%, rotación entre -15% y 15%, saturación entre -30% y 30%, brillo entre -25% y 25%, y un nivel de desenfoque del 1.5%. Con estas técnicas de aumento, se generaron tres veces más imágenes, resultando en un total de 604 imágenes para el entrenamiento, con este conjunto de datos se obtuvieron los siguientes resultados:

**Figura 38.**  
Perdidas de caja delimitadora y clases primer conjunto con aumento de datos



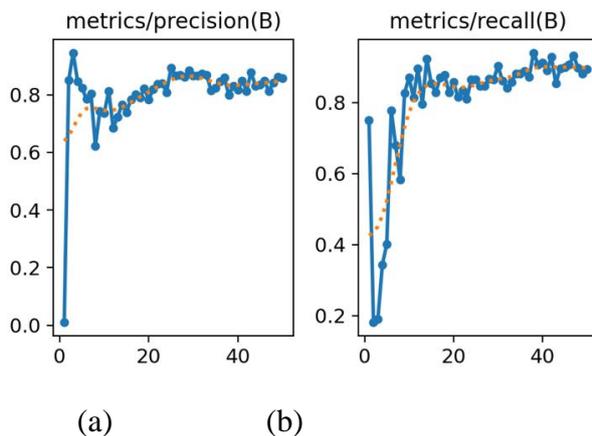
**Nota:** (a) Perdidas de caja delimitadora, (b) Perdidas de clases

Las pérdidas de la caja delimitadora para el segundo modelo muestran resultados más consistentes en comparación con el modelo anterior, y esto se atribuye en parte a las técnicas de aumento de datos que contribuyeron a reducir las fluctuaciones tanto en el entrenamiento como en la validación. Según el gráfico (a), las pérdidas durante el entrenamiento exhiben un descenso constante, mientras que en la validación, aunque se observan menos fluctuaciones, aún existen puntos donde las pérdidas aumentan repentinamente. Aunque el sobreajuste se ha reducido en este caso, como lo indica la curva de validación siendo inferior a la del entrenamiento, es importante destacar que persisten algunos picos en los cuales las pérdidas aumentan.

En cuanto al gráfico (b), que representa las pérdidas de clasificación de clases, se observa una tendencia descendente constante en las pérdidas de entrenamiento a lo largo del proceso, lo cual es indicativo de que el modelo está clasificando correctamente las clases de los objetos. En cuanto a las pérdidas de validación, también se observa un cambio significativo, eliminando la mayoría de los picos que representaban altas pérdidas. Además, al igual que en el gráfico anterior, se evidencia una reducción considerable del sobreajuste.

**Figura 39.**

Métricas de evaluación generales primer conjunto con aumento de datos

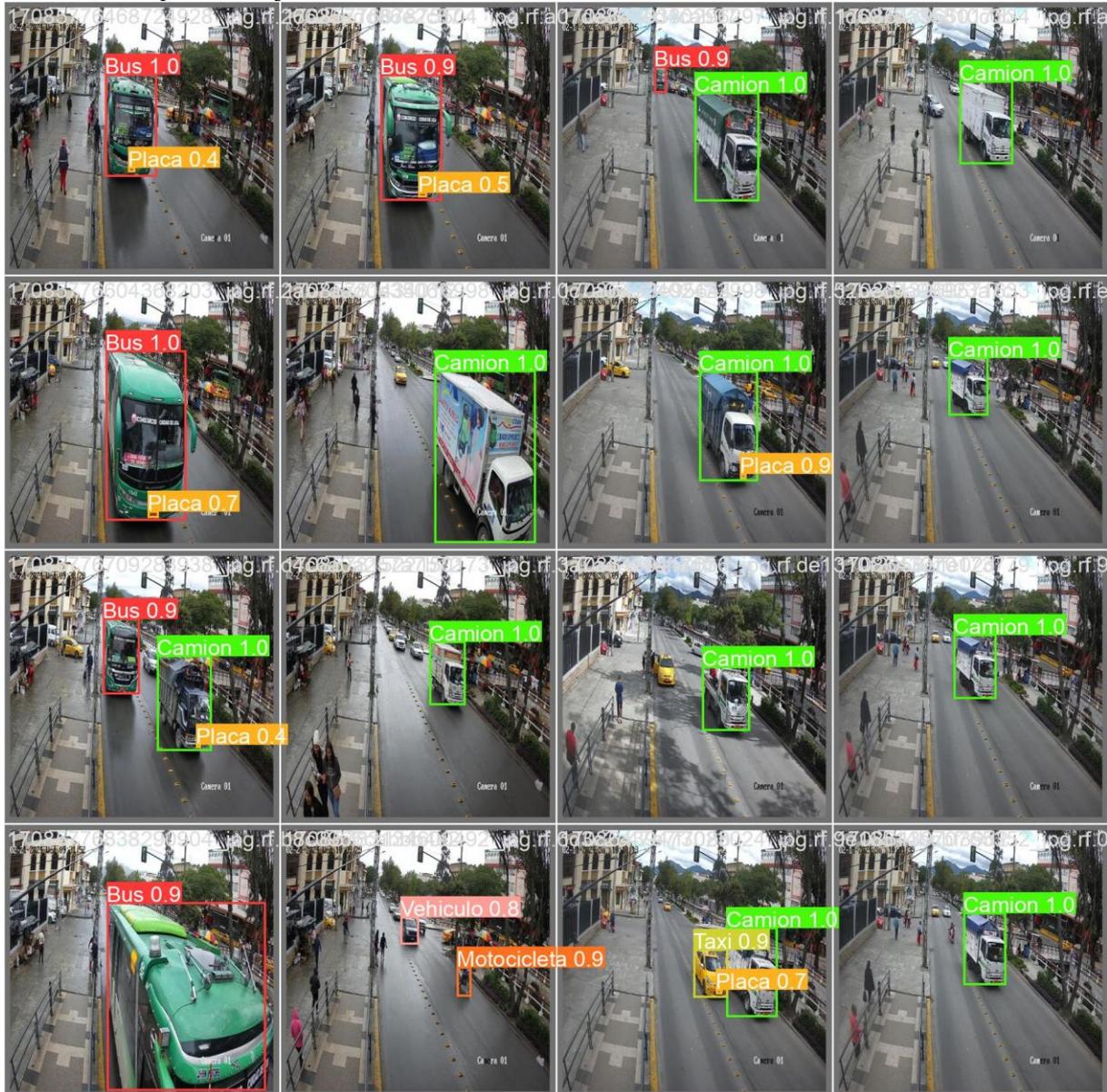


**Nota:** (a) Precisión, (b) Recall

La presión este modelo implementan un conjunto con aumento de datos (a) ha mejorado respecto a la anterior debido a que se observa que la precisión empieza a estabilizar su progreso aproximadamente en la iteración numero 15 manteniendo en un porcentaje elevado de aciertos durante las siguientes iteraciones de entrenamiento. Por otro lado, la curva de recall (b) es más consistente con respecto a la del modelo anterior, ya que se pueden observar una caída después

de las primeras iteraciones del entrenamiento, pero posterior a ella asciende con normalidad hasta lograr una estabilización considerable.

**Figura 40.**  
Resultados de primer conjunto con aumento de datos



**Fuente:** Elaboración propia

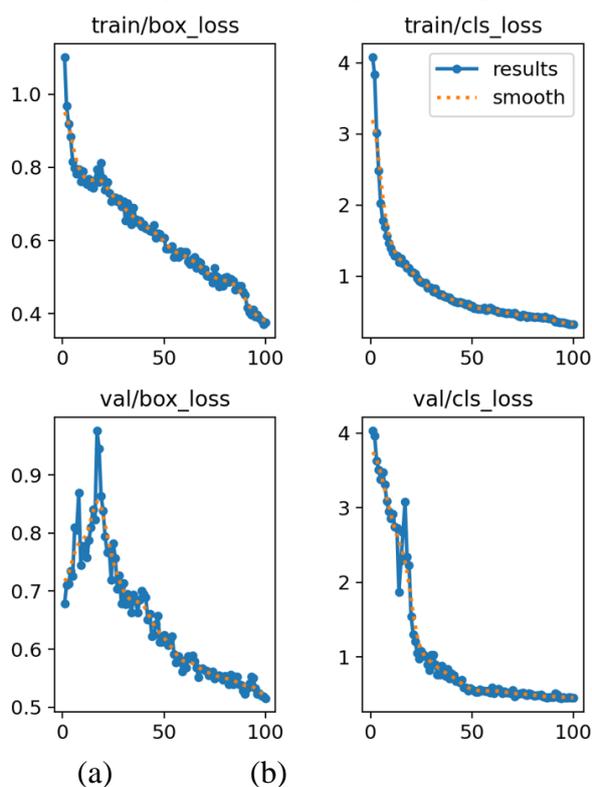
Los resultados representados en la **Figura 40** muestran que el modelo con aumento de datos muestra cierta mejora, en parte debido a que las detecciones se realizaron con un nivel de confiabilidad relativamente alto. Esto sugiere que la inclusión de datos adicionales mediante el aumento de datos contribuyó a mejorar la capacidad del modelo para realizar predicciones con mayor confianza y precisión.

### 6.3. Resultados para el segundo conjunto de datos

El entrenamiento del segundo conjunto de datos se llevó a cabo con los siguientes parámetros: se utilizó un tamaño de lote de 64 durante 100 iteraciones, con un total de 500 imágenes. De estas imágenes, 355 se destinaron al conjunto de entrenamiento, lo que representa el 70% del total. Para la validación se utilizaron 94 imágenes, equivalente al 20% del total, mientras que 51 imágenes, que corresponden al 10% restante, se reservaron para la verificación del modelo.

**Figura 41.**

Perdidas de caja delimitadora y clases segundo conjunto de datos

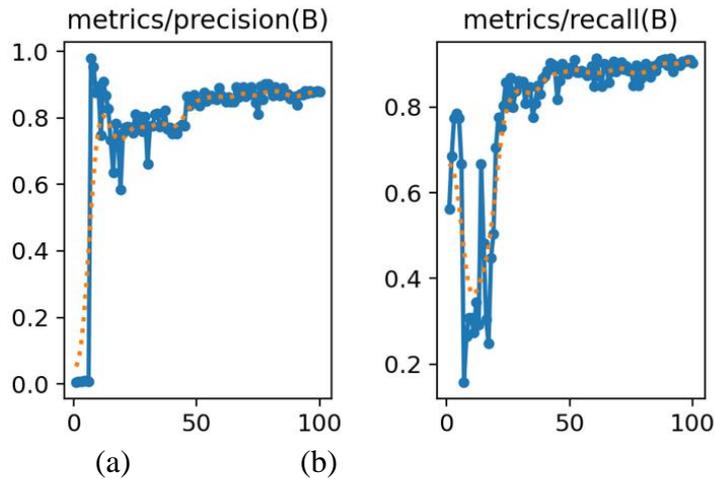


**Fuente:** Elaboración propia

Los resultados del entrenamiento del segundo conjunto de datos indican que la curva de pérdida de cajas delimitadoras (a) durante la etapa de entrenamiento exhibe un descenso constante, con mínimas fluctuaciones, lo que sugiere una correspondencia precisa entre las predicciones realizadas por el modelo y las coordenadas reales del conjunto de datos. Por otro lado, la curva de validación (b) muestra picos que reflejan ciertas pérdidas; sin embargo, logra descender de manera consistente hasta la iteración número 100. Estas gráficas sugieren que el entrenamiento se llevó a cabo de manera eficiente, mejorando la capacidad de detección del

modelo a lo largo del proceso. Inicialmente, parece que el modelo podría estar sufriendo de sobreajuste, ya que la línea de pérdida de validación supera a la del entrenamiento; sin embargo, logra estabilizarse y mejorar el rendimiento del modelo con cada iteración.

**Figura 42.**  
Métricas generales del modelo



**Fuente:** Elaboración propia

La precisión de este entrenamiento, se refleja en el grafico (a) en donde la predicción inicia en cero y posteriormente tiene un pico repentino hasta aproximadamente 1, esto significa que en esta iteración todas las predicciones fueron correctas, lo cual es poco común tomando en cuenta que se está iniciando el entrenamiento y podría significar algún fallo en los datos o en los hiperparámetros de entrenamiento. Este fallo se puede evidenciar en la **Figura 43**, en donde la predicción del bus tiene un 80% de confiabilidad, sin embargo, no hay ningún elemento en esta zona por lo tanto se afirma que el modelo ha fallado.

Por otro lado, la curva de recall (b) inicia con un valor favorable, sin embargo, tiene un descenso, lo que sugiere que en este tramo se están perdiendo algunas instancias positivas en las predicciones, posteriormente vuelve a ascender de manera gradual, lo que significa que el modelo ha mejorado su capacidad para reconocer correctamente los objetos.

**Figura 43.**  
Resultados de segundo conjunto de datos



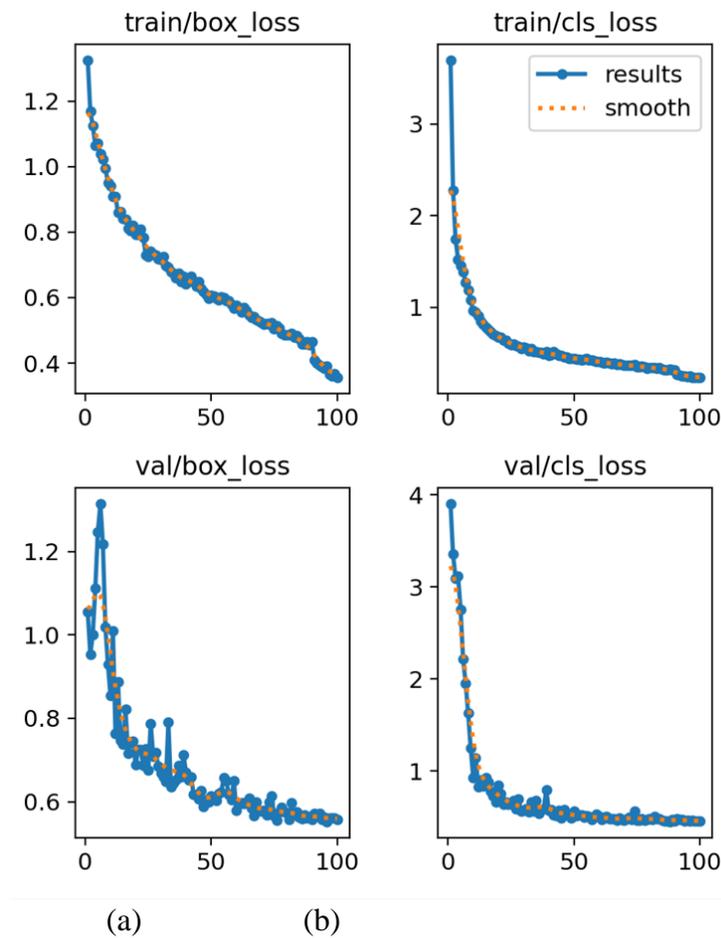
Fuente: Elaboración propia

#### 6.4. Resultados para el segundo conjunto con aumento de datos

El entrenamiento de este modelo se llevó a cabo con los parámetros mencionados en el apartado anterior. Además, se implementaron técnicas de aumento de datos que incluyeron zoom en un rango del 0 al 30%, rotación entre -15% y 15%, saturación entre -30% y 30%, brillo entre -25% y 25%, y desenfoque del 1.5%. Con estas técnicas de aumento, se generaron tres veces más imágenes, resultando en un total de 1210 imágenes para el entrenamiento.

**Figura 44.**

Perdidas de caja delimitadora y clases del segundo conjunto con aumentación de datos



**Fuente:** Elaboración propia

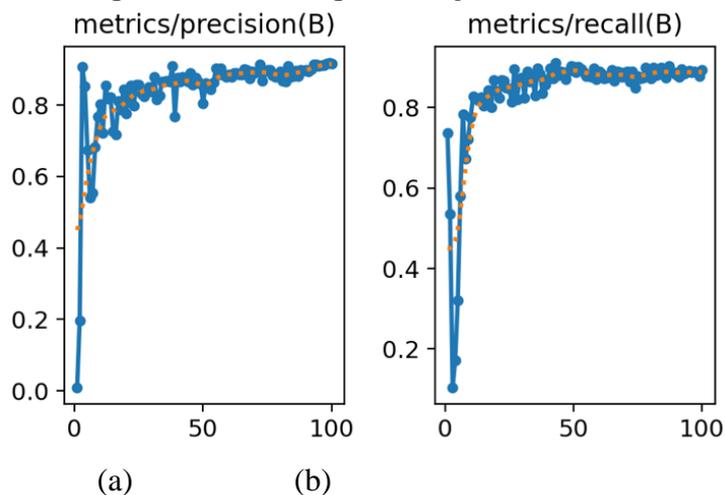
La curva de pérdidas de la caja delimitadora (a) en el conjunto de entrenamiento muestra un descenso constante hasta alcanzar las 100 iteraciones, lo cual indica que el modelo está mejorando de manera continua y consistente en su capacidad para predecir con precisión las coordenadas de las cajas delimitadoras a medida que progresa el entrenamiento. Por otro lado, en el proceso de validación, la curva muestra ciertas fluctuaciones, pero en general su tendencia es descendente y favorable. Aunque puede haber algunas fluctuaciones a lo largo del tiempo, la tendencia general es hacia una disminución en la pérdida de validación, lo que indica que el modelo pueda realizar predicciones precisas en nuevos datos y no simplemente memorizar los ejemplos de entrenamiento.

La pérdida de clasificación de clases (b) muestra una curva consistente tanto en el conjunto de entrenamiento como en el de validación, lo que indica que el modelo está aprendiendo de manera favorable a lo largo de las 100 iteraciones.

En el conjunto de entrenamiento, la curva tiene un decrecimiento favorable, lo que sugiere que el modelo está mejorando en su capacidad para clasificar correctamente las diferentes clases de objetos en las imágenes. De manera similar, en el conjunto de validación, la pérdida de clasificación de clases también muestra un decrecimiento favorable con muy pocas fluctuaciones. Esta consistencia la curva indica que el modelo está generalizando bien a datos que no ha visto durante el entrenamiento y que está aprendiendo patrones útiles en los datos de validación.

**Figura 45.**

Métricas generales con el segundo conjunto con aumentación de datos



**Fuente:** Elaboración propia

La curva de precisión (a) comienza en cero, lo que indica que inicialmente no se realizaron predicciones correctas. Sin embargo, aumenta favorablemente a medida que avanza el entrenamiento, lo que sugiere que el modelo está mejorando en su capacidad para realizar predicciones precisas. Aunque existen algunas fluctuaciones, el progreso general es ascendente y la curva logra estabilizarse rápidamente en un nivel satisfactorio.

Por otro lado, la curva de recall (b) también experimenta ciertas caídas al inicio del entrenamiento, lo que sugiere que inicialmente se están perdiendo algunas instancias positivas. Sin embargo, al igual que la curva de precisión, logra estabilizarse rápidamente a medida que avanza el entrenamiento, lo cual significa que el modelo está mejorando en su capacidad para recuperar instancias positivas y que está aprendiendo a hacerlo de manera consistente.

En la **Figura 46** se puede evidenciar que las predicciones del modelo tienen un alto nivel de detección y el porcentaje de confianza es considerablemente alto. Esto significa que el modelo está identificando correctamente objetos en las imágenes con una alta certeza en la precisión de sus predicciones.

**Figura 46.**  
Resultados del segundo conjunto con aumentación de datos



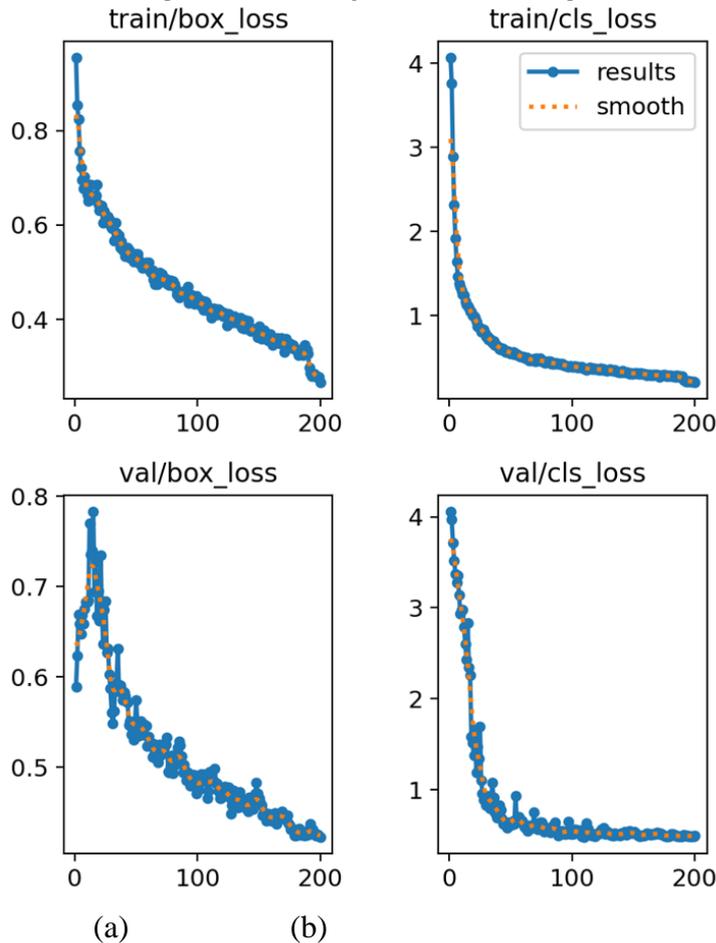
Fuente: Elaboración propia

### 6.5. Resultados para el tercer conjunto de datos

Para este tercer conjunto de datos, se consideraron los siguientes parámetros: se utilizaron un total de 1000 imágenes, de las cuales el 70% (700 imágenes) se asignaron para el conjunto de entrenamiento, el 20% (206 imágenes) se destinaron para la validación y el 10% restante (94 imágenes) se reservaron para la verificación del modelo.

**Figura 47.**

Perdidas de caja delimitadora y clases tercer conjunto de datos



**Fuente:** Elaboración propia

En el conjunto de entrenamiento, la curva de pérdidas de cajas delimitadora (a) muestra un descenso consistente a lo largo del tiempo, lo que indica que el modelo está mejorando constantemente su capacidad para predecir con precisión las coordenadas de las cajas delimitadoras. Este descenso consistente es una señal positiva de que el modelo está aprendiendo de manera efectiva de los datos de entrenamiento y está ajustando sus parámetros en la dirección correcta para minimizar las pérdidas asociadas con las cajas delimitadoras.

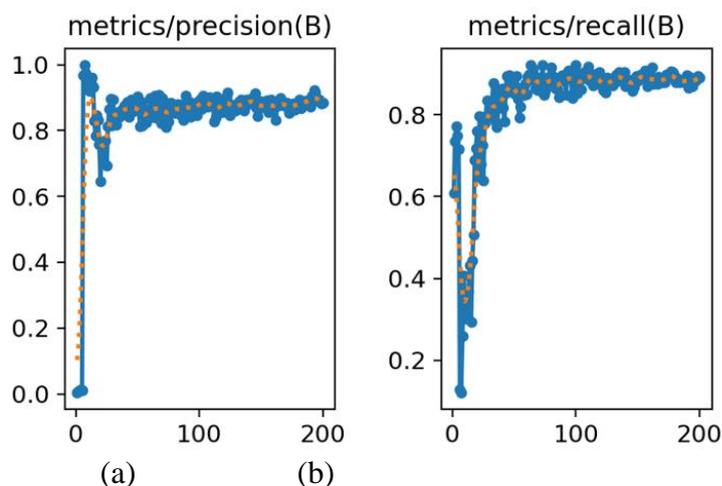
En el conjunto de validación, la curva muestra un pico al inicio, lo cual es común y puede atribuirse a la inicialización de los pesos del modelo o a la aleatoriedad en los datos de validación utilizados para evaluar el modelo. Sin embargo, luego de este pico inicial, la curva desciende de manera constante hasta llegar a las 200 épocas. Este descenso en la pérdida de validación indica que el modelo está generalizando bien a datos que no ha visto durante el entrenamiento, lo cual es un indicador positivo de que el modelo está aprendiendo patrones útiles en los datos y no simplemente memorizando los ejemplos de entrenamiento.

La curva de pérdida de la clasificación de clases (b) en entrenamiento muestra un descenso que alcanza la estabilización con pocas iteraciones. Esto indica que el modelo está aprendiendo de manera efectiva a clasificar correctamente las diferentes clases de objetos. La estabilización temprana de la curva de pérdida en entrenamiento es un indicador positivo de que el modelo está convergiendo hacia una solución óptima con rapidez.

Por otro lado, en validación, la curva de pérdida presenta mínimas fluctuaciones en comparación con los anteriores modelos. Esto sugiere que el modelo está reconociendo bien a datos que no ha visto durante el entrenamiento, ya que las pérdidas se mantienen relativamente estables a lo largo del tiempo. Además, al igual que en entrenamiento, la curva de pérdida de validación se estabiliza rápidamente, lo que indica que el modelo está clasificando correctamente las clases de objetos en nuevos datos.

**Figura 48.**

Métricas generales con el tercer conjunto de datos



**Fuente:** Elaboración propia

La curva de precisión (a) muestra algunas caídas, pero estas no son tan significativas, ya que se recupera rápidamente y alcanza la estabilización en pocas épocas. Esto sugiere que, aunque pueda haber cierta variabilidad en la precisión de las predicciones, el modelo es capaz de ajustarse y mejorar su rendimiento con relativa rapidez, alcanzando un nivel estable de precisión con pocas iteraciones.

Por otro lado, la curva de recall (b) también experimenta una caída considerable en las primeras iteraciones, pero logra recuperarse y alcanzar una estabilización temprana. Esto indica que el modelo es capaz de recuperar instancias positivas de manera efectiva, incluso después de experimentar una disminución en su capacidad de detección.

La **Figura 49** muestra los resultados de este modelo, donde se puede apreciar un alto porcentaje de aciertos en las predicciones realizadas. Esto indica que el modelo ha tenido un

rendimiento exitoso en la tarea específica para la cual fue entrenado, logrando detectar una gran cantidad objetos.

**Figura 49.**

Resultados del tercer conjunto de datos



**Fuente:** Elaboración propia

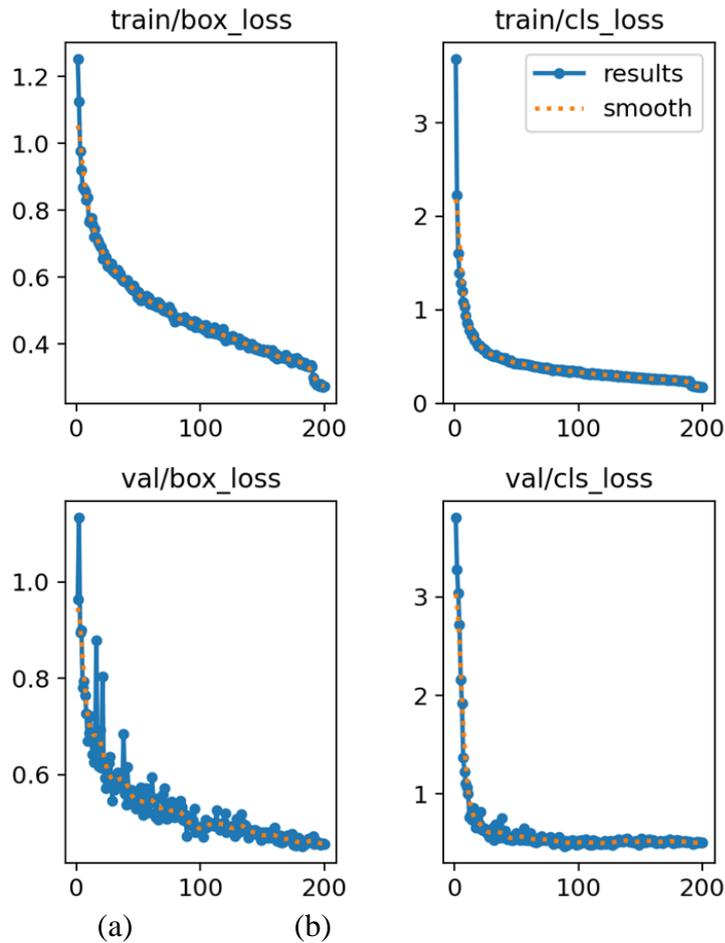
## 6.6. Resultados para el tercer conjunto con aumento de datos

El tercer conjunto de datos, se entrenó con los mismos parámetros del modelo anterior. Además, se implementaron técnicas de aumento de datos que incluyeron zoom en un rango del 0 al 30%, rotación entre -15% y 15%, saturación entre -30% y 30%, brillo entre -25% y 25%,

y desenfoque del 1.5%. Con estas técnicas de aumento, se generaron tres veces más imágenes, resultando en un total de 2400 imágenes para el entrenamiento.

**Figura 50.**

Perdidas de caja delimitadora y clases tercer conjunto con aumento de datos



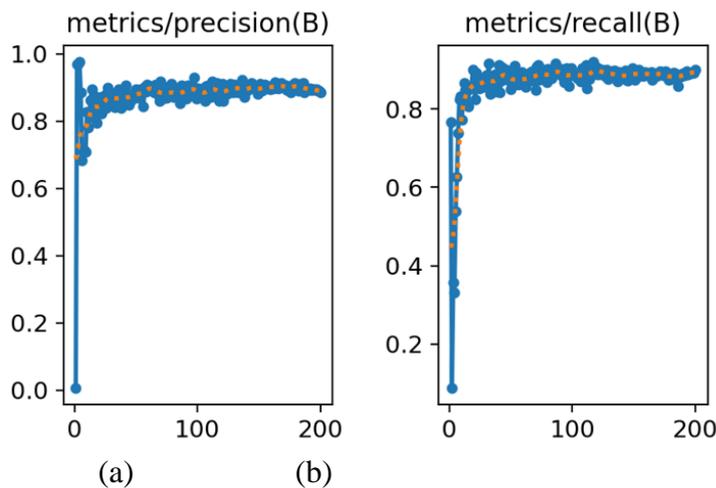
**Fuente:** Elaboración propia

Las pérdidas de la caja delimitadora (a) se representan mediante una curva. En el entrenamiento, esta curva es extremadamente consistente, sin presentar fluctuaciones, y desciende de manera favorable hasta alcanzar las pérdidas mínimas cerca de la iteración número 200. En la validación, la curva sí presenta fluctuaciones, pero en términos generales desciende de manera normal. Esto sugiere que tener grandes cantidades de datos y aplicar técnicas de aumento de datos contribuyó significativamente a reducir las pérdidas de manera favorable. Aunque la curva de validación presente algunas fluctuaciones, el descenso general indica que el modelo está generalizando bien a datos no vistos durante el entrenamiento, lo que es crucial para obtener resultados confiables en aplicaciones del mundo real.

Para la pérdida de clasificación de clases (b), se observa que ambas curvas descienden favorablemente tanto en entrenamiento como en validación. Además, ambas curvas se estabilizan durante las primeras 50 iteraciones y se mantienen así hasta el final del entrenamiento. Este comportamiento indica que el modelo aprendió de manera efectiva durante el entrenamiento, logrando reducir consistentemente la pérdida asociada con la clasificación de clases. La estabilización temprana de las curvas sugiere que el modelo ha alcanzado un nivel óptimo de aprendizaje en una etapa temprana del proceso de entrenamiento y que continúa manteniendo este nivel de rendimiento a lo largo del entrenamiento.

**Figura 51.**

Métricas generales con el tercer conjunto con aumento de datos



**Fuente:** Elaboración propia

La curva de precisión (a) muestra un inicio con valores bajos, pero estos ascienden rápidamente y se estabilizan antes de las 50 iteraciones, manteniéndose así hasta el final del entrenamiento. Esto sugiere que el modelo comienza con una baja precisión, pero mejora rápidamente durante las primeras iteraciones y luego alcanza un nivel estable de precisión, lo que indica que está realizando predicciones consistentes y precisas a lo largo del entrenamiento.

Por otro lado, la curva de recall (b) muestra algunos decaimientos que no son significativos en comparación con el resto de las iteraciones. A pesar del decaimiento inicial, la curva también alcanza una estabilización de forma temprana. Esto sugiere que el modelo tiene la capacidad de recuperar instancias positivas durante el entrenamiento y estabilizarse tempranamente.

En la Figura 52, se observa que la detección de objetos se realiza con una mayor eficacia. Los porcentajes de confiabilidad asociados a las detecciones son altos y varían desde 0.7 en el peor de los casos hasta 1.0, siendo este último el mejor de los casos. Esto indica que el modelo es capaz de detectar objetos con una alta certeza en la precisión de sus predicciones. Los porcentajes de confiabilidad cercanos a 100% indican un alto grado de certeza en las detecciones realizadas por el modelo, lo que sugiere que las predicciones son muy precisas y confiables.

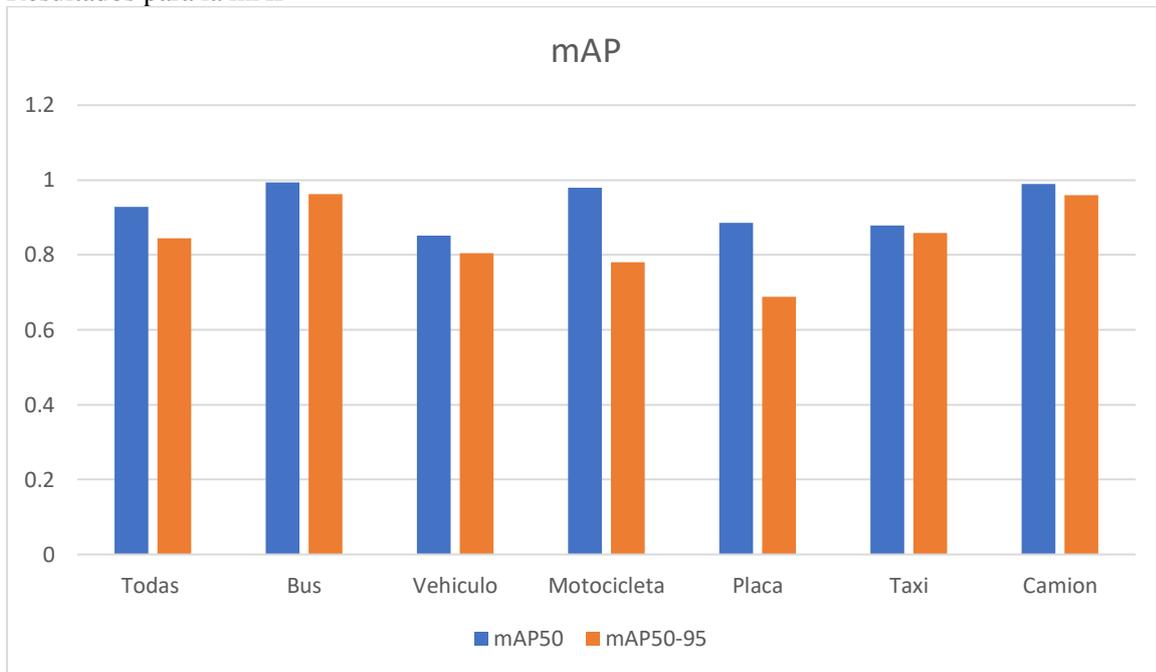
**Figura 52.**  
Resultados del tercer conjunto con aumento de datos



Fuente: Elaboración propia

Al ser este modelo uno con los mejores resultados se revisa una métrica adicional para evaluar el comportamiento del modelo la cual se puede visualizar a continuación.

**Figura 53.**  
Resultados para la mAP



**Fuente:** Elaboración propia

En base a los resultados de la **Figura 53** para media de la precisión promedio (*mean Average Precision*<sup>2</sup>) se puede destacar lo siguiente:

El modelo tiene un buen rendimiento general en la detección de todas las clases, con un mAP50 de 0.929 y un mAP50-95 de 0.844. Esto sugiere que el modelo es capaz de detectar una amplia variedad de objetos con una precisión considerable en una variedad de condiciones de IoU<sup>3</sup>.

La clase "Bus" tiene el mejor rendimiento en términos de mAP50 y mAP50-95, con valores de 0.994 y 0.963 respectivamente. Esto indica que el modelo es particularmente preciso en la detección de buses, tanto en condiciones de IoU más estrictas como en un rango más amplio.

Al igual que con la clase "Bus", la clase "Camión" muestra un rendimiento excepcionalmente alto, con un mAP50 de 0.989 y un mAP50-95 de 0.959.

La clase "Taxi" tiene un mAP50 de 0.878 y un mAP50-95 de 0.859, lo que indica un rendimiento bastante consistente en diferentes condiciones de IoU.

<sup>2</sup> **mAP:** Es una medida que combina la precisión y el recall de un modelo de detección en múltiples clases de objetos.

<sup>3</sup> **IoU:** Superposición dos regiones de interés

La clase "Vehículo" también muestra un buen rendimiento, con un mAP50 de 0.851 y un mAP50-95 de 0.804. Esto sugiere que el modelo es robusto en la detección de vehículos en general.

Aunque el mAP50 para la clase de objetos "Motocicleta" es alto (0.979), el mAP50-95 es significativamente más bajo (0.781). Esto indica que el modelo es muy preciso en la detección de motocicletas bajo condiciones de IoU menos estrictas, pero su precisión disminuye en condiciones más exigentes, como cuando el objeto está en un punto lejano o se cambian las condiciones luminarias.

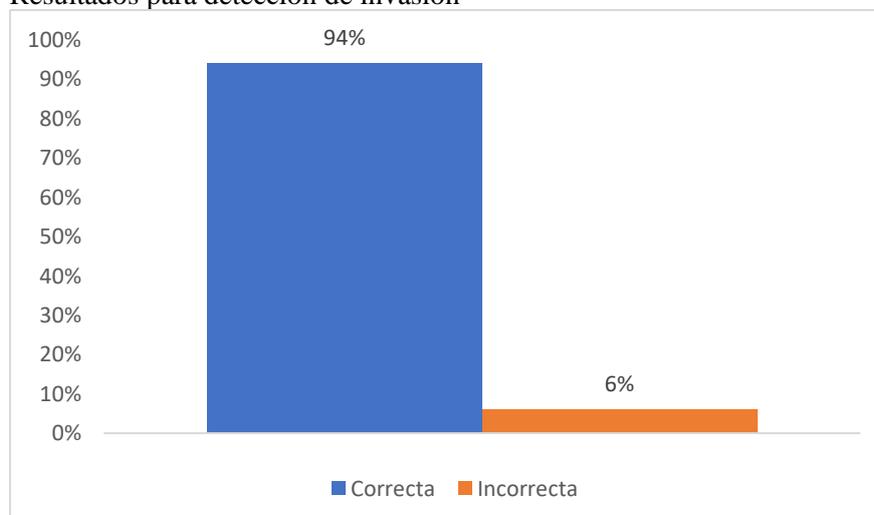
Finalmente, la clase "Placa" muestra un mAP50 relativamente alto (0.886), pero un mAP50-95 más bajo (0.688). Esto sugiere que el modelo puede detectar placas con alta precisión en condiciones de IoU menos estrictas, pero su precisión disminuye en condiciones de IoU más altas. Sin embargo, se debe tomar en consideración que el modelo será aplicado en una zona específica donde se espera que la detección de objetos, especialmente las placas de los vehículos, se realice en condiciones óptimas. Esta zona asegura una iluminación adecuada, ángulos de visión favorables y distancias apropiadas entre la cámara y los objetos.

### 6.7. Resultados de la detección de invasión

Durante un período de aproximadamente una hora, se realizaron pruebas del modelo, poniéndolo a prueba en tiempo real. Estos ensayos proporcionaron un conjunto de resultados que documentan el rendimiento y la eficacia del modelo.

#### Figura 54.

Resultados para detección de invasión



Fuente: Elaboración propia

El modelo logró una tasa de precisión del 94% en la detección de objetos, la tasa de detecciones incorrectas fue del 6%, lo que sugiere que hubo un pequeño porcentaje de errores

en las detecciones realizadas por el modelo durante las pruebas en tiempo real. Aunque se registraron algunas detecciones incorrectas, el modelo demostró ser eficaz en la mayoría de los casos, con una tasa de precisión por encima del 90%. Esto sugiere que el modelo es capaz de detectar una invasión en tiempo real con un alto grado de fiabilidad.

## 7. Discusión

En base a los resultados presentados en el apartado anterior, se determinó que el modelo entrenado con conjunto con aumento de datos tiene el mejor rendimiento como modelo para la detección de objetos. Por consiguiente, en este apartado se llevó a cabo una comparación con otros modelos similares con el fin de evaluar su efectividad en relación con investigaciones previas.

En la investigación de (Huu-Huy, 2023) se entrenó un modelo basado en la arquitectura YOLOv8 para detectar objetos de una carreta: bicicleta, motocicleta, carro, bus y camión. Obteniendo un valor de mAP50 muy alto y similar entre cada una de ellas, de 0,988. Mientras, el valor de mAP50-95 es diferente y también alto alcanzando 0,873. La clase de automóvil con el valor más alto de mAP50-95 fue 0.961. Por el contrario, la clase de motocicletas tuvo el valor más bajo de mAP50-95 de 0,702.

Nuestro modelo, el cual detecta 6 tipos de objetos: bus, carro, camión, motocicleta, taxi y placa; alcanzo una mAP50 de todas las clases es 0.929 y una mAP50-95 de 0.844, por otro lado, la clase que tiene una media más alta es “Bus” con una mAP50 de 0.994 y una mAP50-95 de 0.963 por otro lado la clase placa tiene una mAP50 de 0.886 y una mAP50-95 de 0.688, siendo esta la más baja.

Este análisis revela que el modelo propuesto en este estudio exhibe una mAP50 que se asemeja a la del modelo de referencia. Sin embargo, al considerar también la mAP50-95, se observa una disminución significativa en los valores totales. Este descenso en el desempeño se atribuye principalmente a la clase de placas, donde el modelo presenta dificultades para alcanzar altos niveles de precisión en un rango más amplio de umbrales de IoU. A pesar de esta disminución en la precisión a nivel de todas las clases, los resultados obtenidos siguen siendo considerados aceptables, lo que sugiere que el modelo desarrollado en este trabajo de investigación aún tiene un rendimiento competitivo en la detección de objetos.

Por otro lado, en lo que respecta a la detección de invasión de carril, es importante destacar que no se han encontrado investigaciones que se centren exclusivamente en el uso de visión por computadora para este propósito, la mayoría de los proyectos desarrollados en este ámbito tienden a utilizar prototipos electrónicos. Estas soluciones electrónicas pueden incluir

sensores de proximidad, cámaras adicionales u otros dispositivos de detección física para llevar a cabo la tarea de detección de invasión. Tal es el caso de (Abad Gómez & Guerrero Salas, 2017) quienes para realizar el proceso de detección crearon de un sistema electrónico que utiliza diodos infrarrojos para detectar la presencia de vehículos, un módulo de cámara Raspberry NoIR para la adquisición de imágenes, y una placa Raspberry Pi 3 para la programación y control del sistema. Además, este sistema emplea un algoritmo de red neuronal para la detección de vehículos, logrando una precisión aproximada del 78.04%. Sin embargo, esta precisión se considera insuficiente para la aplicación deseada.

En este sentido, el proyecto logra sobrepasar las expectativas establecidas para este sistema, ya que la detección implementada alcanza un 94% de precisión y fiabilidad en la identificación de vehículos invasores dentro de la zona especificada.

En base a estas observaciones, se puede concluir que este sistema es adecuado para su implementación en lo que respecta a la detección de invasión de carril, que fue su propósito original. Sin embargo, en cuanto a la detección de placas, la cual se incorporó como una funcionalidad adicional, se observa una tendencia a generar errores en la lectura. Por lo tanto, se hace evidente la necesidad de mejorar tanto el prototipo como las técnicas de procesamiento de imágenes para obtener mejores resultados en la lectura de las placas.

## 8. Conclusiones

- La técnicas y modelos de visión por computadora implementadas en este proyecto permitieron lograr el cumplimiento del objetivo principal que era la detección de invasión de carril. A través de esta investigación, se ha logrado una comprensión profunda de las diversas opciones disponibles para abordar este problema e implementarlas satisfactoriamente.
- Se generaron tres conjuntos de datos los cuales incluyen 250, 500 y 1000 imágenes utilizando el prototipo instalado en un tramo del carril de bus en la ciudad de Loja. Con la implementación de técnicas de aumento de datos, se logró expandir estos conjuntos a: 604, 1210 y 2400 imágenes, respectivamente. Este aumento significativo en la cantidad de datos resultó en mejoras favorables en el entrenamiento, puesto que los modelos entrenados con conjuntos de datos aumentados demostraron un rendimiento superior en comparación con los que usaron el conjunto de datos original. Destacando entre ellos el tercer conjunto de datos aumentado, el cual se utilizó para el desarrollo del software.
- La evaluación realizada al conjunto de datos reveló resultados prometedores en términos de la capacidad predictiva y la precisión de los modelos de visión por computadora entrenados. Las curvas de precisión y recall mostraron mejoras rápidas y estabilizaciones tempranas, asegurando predicciones consistentes. Además, la detección de objetos resultó altamente efectiva, evidenciada por los porcentajes de confiabilidad que reflejaban una precisión considerable en las predicciones del modelo.
- Es importante destacar que este proyecto pretende proporcionar apoyo a las autoridades tránsito de la ciudad de Loja en la identificación y sanción de conductores que invaden el carril de bus, una infracción catalogada como contravención de cuarta y sexta clase según el COIP. Dado que el tramo en cuestión es extenso, 12km aproximadamente, resulta difícil para un agente de tránsito cubrirlo en su totalidad, lo que hace que la implementación de este sistema permita facilitar este trabajo de manera efectiva. No obstante, es importante destacar que el software no puede trabajar de manera independiente, requiere la colaboración de una persona que verifique y valide los datos generados.

## 9. Recomendaciones

- Seguir ampliando en conjunto de datos de tal forma que se pueda proporcionar mayor variabilidad, en este sentido se podría considerar la inclusión de imágenes tomadas en diferentes momentos del día y en condiciones climáticas variadas. Asimismo, se podrían incluir imágenes con diferentes tipos de vehículos y en distintas posiciones para aumentar la diversidad del conjunto de datos.
- Explorar el uso de técnicas más avanzadas de procesamiento de imágenes para mejorar la calidad de las imágenes antes de la aplicación de OCR, de esta forma se podrán obtener imágenes de mayor calidad y claridad, mejorando precisión y confiabilidad del proceso de reconocimiento óptico de caracteres (OCR).
- Mejorar el prototipo instalado en la parada de bus, en base pruebas y verificación de los resultados incluyendo diferentes ángulos y alturas de montaje para simular diversas condiciones de visualización. Esto ayudaría a evaluar la validez del sistema frente a variaciones en la posición y orientación de la cámara, generando resultados más consistentes en diferentes escenarios del entorno real.
- El sistema opera eficientemente desde las 06:00 hasta las 18:00 horas, durante las cuales las detecciones se llevan a cabo con mayor eficacia. Sin embargo, fuera de este período, la capacidad del sistema se ve afectada debido a las condiciones de iluminación. Por esta razón, se sugiere el uso de una cámara de alta resolución con capacidad nocturna para extender el horario de funcionamiento del sistema.

## 10. Bibliografía

- Abad Gómez, A. A., & Guerrero Salas, E. L. (2017). *Desarrollo de prototipo de un sistema para compartir carril de la metrovía con vehículos livianos aplicando una tarifa, con el propósito de disminuir las invasiones al carril exclusivo.*
- Abad, J. (2022, July 25). *Transporte público: ¿Cómo influye dónde vivimos y las características socioeconómicas en nuestro patrón de movilidad?* - Moviliblog.  
<https://blogs.iadb.org/transporte/es/transporte-publico-como-influye-donde-vivimos-y-las-caracteristicas-socioeconomicas-en-nuestro-patron-de-movilidad/>
- Ahmed, M., Awel, M. A., & Abidi, A. I. (2019). REVIEW ON OPTICAL CHARACTER RECOGNITION. *International Research Journal of Engineering and Technology*, 3666.  
[www.irjet.net](http://www.irjet.net)
- Alegre, E., Pajares, G., & De La Escalera, A. (2016b). *Conceptos y Métodos en Visión por Computador.*
- Arkipov, A. (2023, June 5). *Firestore VS MongoDB: What To Choose?*  
<https://www.techmagic.co/blog/firebase-vs-mongodb/>
- AWS. (2023). *¿Qué es un IDE? - Explicación de los entornos de desarrollo integrado - AWS.*  
<https://aws.amazon.com/es/what-is/ide/>
- Boesch, G. (2023). *Computer Vision Model Performance Evaluation (Guide 2023) - viso.ai.*  
<https://viso.ai/computer-vision/model-performance/>
- Briceño, G. (2018). *Transporte terrestre | Qué es, características, historia, tipos, ventajas, usos.* <https://www.euston96.com/transporte-terrestre/>
- Brownlee, J. (2019). *Deep Learning for Computer Vision Image Classification, Object Detection and Face Recognition in Python.*
- Celi, S. (2018). *Análisis del comportamiento del transporte público a nivel mundial Behavioral analysis of public transportation worldwide.* 39.
- Chai, J., Zeng, H., Li, A., & Ngai, E. W. T. (2021). Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Machine Learning with Applications*, 6, 100134. <https://doi.org/10.1016/J.MLWA.2021.100134>

- Chamba Coronel, J. C. (2013). *Análisis de riesgos y seguridad vial, en el corredor exclusivo del sistema integrado de transporte urbano SITU en la ciudad de Loja*. Universidad Técnica Particular de Loja.
- Chawla, A., Gupta, A., & Shushrutha, K. S. (2022). *Intelligent Information Retrieval: Techniques for Character Recognition and Structured Data Extraction*.  
<https://doi.org/10.6084/m9.jetir.JETIR2207457>
- Clarke, M. (2023, July 24). *Jupyter Notebook vs PyCharm: Comparación de software | TechRepublic*. <https://www.techrepublic.com/article/jupyter-notebook-vs-pycharm/>
- Corrales Bastidas, B. E. (2022). *Sistema de visión artificial aplicado a la detección y conteo automático de objetos superpuestos de forma circular*.
- DB Engines. (2023). *Firestore Realtime Database vs. MongoDB vs. PostgreSQL Comparison*.  
<https://db-engines.com/en/system/Firebase+Realtime+Database%3BMongoDB%3BPostgreSQL>
- Díaz, Y. (2017, December 14). *SITU: Sistema que mejoran el servicio de transportación | Municipio de Loja*. <https://www.loja.gob.ec/noticia/2017-12/situ-sistema-que-mejoran-el-servicio-de-transportacion>
- Ghosh, A., Sufian, A., Sultana, F., Chakrabarti, A., & De, D. (2019). Fundamental concepts of convolutional neural network. *Intelligent Systems Reference Library*, 172, 519–567.  
[https://doi.org/10.1007/978-3-030-32644-9\\_36](https://doi.org/10.1007/978-3-030-32644-9_36)
- González, A., Martínez, F., Pernía, A., Alba, F., Castejón, M., Ordieres, J., & Vergara, E. (2006). *Técnicas y Algoritmos Básicos de Visión Artificial*. Universidad de la Rioja.
- Hikvision. (n.d.). *Compare Page - Hikvision*. Retrieved October 29, 2023, from  
<https://www.hikvision.com/es-la/compare-page/>
- Hurtado Torres, G. I. (2018). *UNIVERSIDAD NACIONAL DE LOJA FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES*.
- Huu-Huy, N. (2023). VEHICLE-DETECTION-BASED TRAFFIC DENSITY ESTIMATION AT ROAD INTERSECTIONS. *International Journal of Open Information Technologies*, 11(7). <https://cyberleninka.ru/article/n/vehicle-detection-based-traffic-density-estimation-at-road-intersections>

- IBM. (2022). *¿Que es Java? | IBM*. <https://www.ibm.com/mx-es/topics/java>
- Jiménez Rodríguez, M. J. (2019). *PROCESAMIENTO DE IMÁGENES DIGITALES*.
- Jocher, G., & Chaurasia, A. (2024, February 3). *Ultralytics YOLOv8 Docs*.  
<https://docs.ultralytics.com/es>
- Joyanes Aguilar, L. (2008). *FUNDAMENTOS DE PROGRAMACIÓN Algoritmos, estructura de datos y objetos Cuarta edición* (J. L. García & C. Sánchez, Eds.; Cuarta). MCGRAW-HILL.
- Kashyapa, R. (2022, December 24). *Accuracy of Machine Vision Systems: A Guide to Evaluating Performance*. <https://www.linkedin.com/pulse/accuracy-machine-vision-systems-guide-evaluating-raghava-kashyapa>
- Kirillov, A., He, K., Girshick, R., Rother, C., & Dollar, P. (2018). Panoptic Segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019-June*, 9396–9405. <https://doi.org/10.1109/CVPR.2019.00963>
- Kuncheva, L. I., & Whitaker, C. J. (2015). Pattern Recognition and Classification. *Wiley StatsRef: Statistics Reference Online*, 1–7.  
<https://doi.org/10.1002/9781118445112.STAT06503.PUB2>
- Lauren, S. (2022). *GitHub - HumanSignal/labelImg*.  
<https://github.com/HumanSignal/labelImg?tab=readme-ov-file>
- Leonard, H. (2021). *THE LATEST COMPUTER VISION*.
- Martínez, J. (2022, January 19). *Cómo Calcular Histogramas de Color en OpenCV - DataSmarts*. <https://www.datasmarts.net/como-calcular-histogramas-de-color-en-opencv/>
- MathWorks. (2023). *Computer vision - MATLAB & Simulink*.  
<https://la.mathworks.com/discovery/computer-vision.html>
- Ministerio de Defensa Nacional. (2021). *CÓDIGO ORGÁNICO INTEGRAL PENAL, COIP*.  
[www.lexis.com.ec](http://www.lexis.com.ec)
- Navarro Soler, Y. (2022, June). *Placa programable Spresense de Sony. Desarrollo de una aplicación de visión por computador*.

<https://upcommons.upc.edu/bitstream/handle/2117/372000/tfg-placa-programable-spresense-de-sony-desarrollo-de-una-aplicaci-n-de-visi-n-por-computador-yeray-navarro-soler.pdf?sequence=1&isAllowed=y>

Nick, A. (2020, August 6). *Jupyter (Lab) vs Colab vs VScode vs PyCharm* | by Nick Anthony | *Medium*. <https://nanthony007.medium.com/jupyter-lab-vs-colab-vs-deepnote-vs-vscode-vs-pycharm-4384c105b087>

Nieto, L. A. (2020, September 17). *DETECCIÓN DE PERSONAS / INTELIGENCIA ARTIFICIAL - OpenCV (HOG + SVM)*. <https://www.linkedin.com/pulse/detecci%C3%B3n-de-personas-inteligencia-artificial-opencv-nieto>

Pérez, P., & Valiente, M. (2018). *Fundamentos básicos del procesamiento de imágenes — documentación de Curso de imágenes médicas - 1.0*. <https://www.famaf.unc.edu.ar/~pperez1/manuales/cim/cap2.html>

Puga, B., Ramírez, A., & Soto, J. (2021, July). (PDF) *Un algoritmo de detección de bordes basado en reglas deterministas*. [https://www.researchgate.net/publication/353105130\\_Un\\_algoritmo\\_de\\_deteccion\\_de\\_bordes\\_basado\\_en\\_reglas\\_deterministas](https://www.researchgate.net/publication/353105130_Un_algoritmo_de_deteccion_de_bordes_basado_en_reglas_deterministas)

Reis, G., Lobato, W., Rosário, D., Cerqueira, E., & Villas, L. A. (2023). Avaliação de Técnicas de Detecção de Pedestres para Veículos Autônomos. *Anais Do Workshop Em Desempenho de Sistemas Computacionais e de Comunicação (WPerformance)*, 61–72. <https://doi.org/10.5753/WPERFORMANCE.2023.230699>

Ren, J., Wang, L., Park, Y., & Shin, Y. (2023). Gradual OCR: An Effective OCR Approach Based on Gradual Detection of Texts. *Mathematics 2023, Vol. 11, Page 4585, 11(22)*, 4585. <https://doi.org/10.3390/MATH11224585>

Roboflow. (n.d.). *Build Vision Models with Roboflow - Roboflow Docs*. Retrieved February 22, 2024, from <https://docs.roboflow.com/>

Sherer, T. (2022, September 7). *Selección de características (minería de datos)* | *Microsoft Learn*. <https://learn.microsoft.com/es-es/analysis-services/data-mining/feature-selection-data-mining?view=asallproducts-allversions>

- Sinha, G. R., & Bajaj, V. (2021). Data deduplication applications in cognitive science and computer vision research. *Data Deduplication Approaches: Concepts, Strategies, and Challenges*, 357–368. <https://doi.org/10.1016/B978-0-12-823395-5.00001-X>
- SourceForge. (n.d.). *Google Colab vs. Jupyter Notebook vs. PyCharm Comparison*. Retrieved October 29, 2023, from <https://sourceforge.net/software/compare/Google-Colab-vs-Jupyter-Notebook-vs-PyCharm/>
- Stolojescu-Crisan, C., Crisan, C., & Butunoi, B. P. (2022). Access control and surveillance in a smart home. *High-Confidence Computing*, 2(1), 100036. <https://doi.org/10.1016/J.HCC.2021.100036>
- Sucar, L. E. (2015). *Visión Computacional Gesture Therapy View project Semantic Segmentation of Images and Videos View project*. <https://www.researchgate.net/publication/267295870>
- Toro Bonilla, M. (2022). *FUNDAMENTOS DE PROGRAMACIÓN: PYTHON*. Universidad de Sevilla. <https://idus.us.es/bitstream/handle/11441/137796/720372-Python.pdf?sequence=1>
- Vélez, B. (2023, May 3). *¿Qué es un sistema de gestión de bases de datos (SGBD)?* <https://www.hostinger.mx/tutoriales/sgbd>

## 11. Anexos

### Anexo 1. Bitácora Transmisión de cámara

BITACORA NRO. 1		
DATOS INFORMATIVOS		
<b>Estudiante</b>	Ibeth Katherine Román	<b>Fecha:</b> 07/12/2023
<b>Tutor de proyecto</b>	Ing. Luis Rodríguez	
INFORMACIÓN GENERAL		
<b>Nombre del proyecto</b>	Desarrollo de un prototipo basado en Visión por Computadora para detectar la invasión del carril de buses en la ciudad de Loja.	
<b>Prueba</b>	Transmisión de cámara	
DETALLES		
<b>Descripción</b>	Leer la transmisión de una cámara IP en un entorno de desarrollo PyCharm.	
<b>Problema</b>	Se observa que las dimensiones de la cámara son excesivamente grandes, afectando la visualización y procesamiento de la transmisión.	
<b>Solución</b>	Redimensionar el tamaño de la transmisión dividiendo tanto la altura como el ancho a la tercera parte.	
<b>Observaciones</b>	La dimensión de los frames irán cambiando de acuerdo al rendimiento del computador.	

### Código

```
# Importar librerías
import cv2

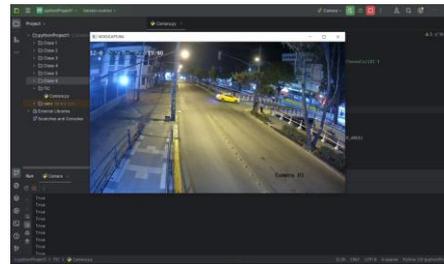
# Crear la Video Captura
cap = cv2.VideoCapture('rtsp://admin:ibeth2023@181.233.50.2:554/Streaming/Channels/101')

# Creamos un ciclo para ejecutar Frames
while True:
    # Leer los Frames
    ret, frame2 = cap.read()
    # Redimensionar frames
    print(ret)
    ancho = int(frame2.shape[1]/3)
    alto = int(frame2.shape[0]/3)
    frame = cv2.resize(frame2, (ancho, alto), interpolation=cv2.INTER_AREA)
    # Mostrar los Frames
    cv2.imshow("VIDEOCAPTURA", frame)
    # Cerrar con lectura de teclado
    t = cv2.waitKey(1)
    if t == 27:
        break
# Liberar la Video Captura
cap.release()
# Cerrar la ventana
cv2.destroyAllWindows()
```

### Evidencias



Antes



Después

## Anexo 2. Recorte de frames y selección de puntos de interés

---

<b>BITACORA NRO. 2</b>		
<b>DATOS INFORMATIVOS</b>		
<b>Estudiante</b>	Ibeth Katherine Román	<b>Fecha:</b>
<b>Tutor de proyecto</b>	Ing. Luis Rodríguez	
<b>INFORMACIÓN GENERAL</b>		
<b>Nombre del proyecto</b>	Desarrollo de un prototipo basado en Visión por Computadora para detectar la invasión del carril de buses en la ciudad de Loja.	
<b>Prueba</b>	Recorte de frames y selección de puntos de interés	
<b>DETALLES</b>		
<b>Descripción</b>	Esta prueba implica recortar la transmisión para procesar solo la zona de interés. Además, permite usar los clics del mouse para obtener el punto en coordenadas (x,y) de esta forma es más fácil obtener los puntos para posicionar la línea de captura.	
<b>Problema</b>	El procesamiento se ralentiza al procesar un frame completo de alta resolución y no se obtienen las coordenadas para posicionar la línea.	
<b>Solución</b>	Recortar específicamente la zona donde se determina que ocurre la invasión y realizar una captura de coordenadas.	
<b>Observaciones</b>	Al recortar la zona específica de interés antes de procesarla, se puede lograr una mejora significativa en el rendimiento del sistema. Además, al obtener las coordenadas por medio del código se puede asegurar un alto nivel de precisión.	
<b>Código</b>		
<pre>#Librerias import cv2 import numpy as np  #Zona de recorte NEW_ROI = np.array([     [400, 430],     [1400, 430],     [1400, 1400],     [400, 1400] ])  #Definición de clase class Coord:     def __init__(self, video_path: str, cap_type=cv2.CAP_FFMPEG):         self.cap = cv2.VideoCapture(video_path, cap_type)          cv2.namedWindow("Frame")         cv2.setMouseCallback("Frame", self.print_Coord)          self.video()      def print_Coord(self, event, x, y, flags, params):         if event == cv2.EVENT_LBUTTONDOWN:             print(f"[{x}, {y}],")      def video(self):         while True:             status, frame2 = self.cap.read()</pre>		

---

```

if not status:
    break

#Determinación de zona de recorte
frame = frame2[NEW_ROI[0][1]:NEW_ROI[2][1], NEW_ROI[0][0]:NEW_ROI[1][0]]

# Mostrar el frame recortado
cv2.imshow("Frame", frame)

#Detener transmisión
if cv2.waitKey(10) & 0xFF == ord('q'):
    break

def __del__(self):
    self.cap.release()
    cv2.destroyAllWindows()

#Ejecturar function principal
if __name__ == '__main__':
    c = Coord("Prueba.mp4")

```

---

## Resultados

---



Puntos de interes para la zona de invasión

```

[252, 73],
[286, 646],
[946, 648],
[487, 89],

```

Puntos de interes para la linea de captura

```

[256, 288],
[646, 292],

```

---

### Anexo 3. Entrenamiento del modelo

---

#### BITACORA NRO. 3

---

#### DATOS INFORMATIVOS

<b>Estudiante</b>	Ibeth Katherine Román	<b>Fecha:</b> 07/12/2023
<b>Tutor de proyecto</b>	Ing. Luis Rodríguez	

---

#### INFORMACIÓN GENERAL

<b>Nombre del proyecto</b>	Desarrollo de un prototipo basado en Visión por Computadora para detectar la invasión del carril de buses en la ciudad de Loja.
<b>Prueba</b>	Entrenamiento del modelo

---

#### DETALLES

<b>Descripción</b>	Entrenamiento del modelo de detección utilizando Google Colab
--------------------	---

<b>Problema</b>	La computadora en donde se está desarrollando el documento no tienen los recursos computacionales necesarios para realizar el entrenamiento.
<b>Solución</b>	Utilizar plataforma de Google Colab para realizar el entrenamiento, debido a su alta capacidad de procesamiento.
<b>Observaciones</b>	Es importante utilizar altos recursos computacionales para asegurar que el entrenamiento tenga resultados exitosos.

### CÓDIGO

```
#Verificación de tarjeta gráfica
!nvidia-smi

+-----+
| NVIDIA-SMI 535.104.05                Driver Version: 535.104.05   CUDA Version: 12.2   |
+-----+-----+-----+-----+-----+-----+
| GPU  Name                               Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf              Pwr:Usage/Cap |      Memory-Usage | GPU-Util  Compute M. |
|====+=====+====+=====+=====+=====+
|  0  NVIDIA A100-SXM4-40GB             Off          | 00000000:00:04.0 Off |             0        | |
| N/A   32C    P0               45W / 400W |  2MiB / 40960MiB |          0%    Default |
|                                           |                      |                      | MIG M. |
+-----+-----+-----+-----+-----+-----+

+-----+
| Processes:                               |
| GPU  GI   CI        PID   Type   Process name                      GPU Memory |
|   ID  ID                                     |            Usage |
+-----+-----+-----+-----+-----+-----+
| No running processes found              |
+-----+

#Conección con Google Colab
from google.colab import drive
drive.mount('/content/drive')

#Descarga de ultralytics
!pip install ultralytics
#Descargar modelo
!wget https://github.com/ultralytics/assets/releases/download/v8.1.0/yolov8n.pt

#Iniccializar archivo de configuración
!pip install PyYAML
import yaml

with open('/content/drive/MyDrive/Resultados3/M1000/Model1000ca/data.yaml')
as file:
    try:
        data = yaml.safe_load(file)
    except yaml.YAMLError as e:
        print(e)

#Entrenamiento del modelo
# Libraries
from ultralytics import YOLO

def main():
    # Model
    model = YOLO('/content/drive/MyDrive/Resultados3/yolov8n.pt')

    # Train

model.train(data='/content/drive/MyDrive/Resultados3/M1000/Model1000ca/data.yaml',
            epochs=200, batch=128, imgsz=640, device='cuda')
```

```
if __name__ == '__main__':
    main()
```

---

## Evidencias

---

```
Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.1.18 Python-3.10.12 torch-2.1.0+cu121 CUDA:0 (NVIDIA A100-SXM4-40GB, 40514MiB)
Model summary (fused): 168 layers, 3006818 parameters, 0 gradients, 8.1 GFLOPs
```

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95): 100%	1/1 [00:01<00:00, 1.11s/it
all	206	327	0.898	0.888	0.929	0.844	
Bus	206	56	0.982	0.954	0.994	0.963	
Vehiculo	206	52	0.806	0.827	0.851	0.804	
Motocicleta	206	58	0.944	0.948	0.979	0.781	
Placa	206	55	0.863	0.782	0.886	0.699	
Taxi	206	51	0.796	0.84	0.878	0.859	
Camion	206	55	1	0.977	0.989	0.959	

Speed: 0.1ms preprocess, 0.4ms inference, 0.0ms loss, 1.0ms postprocess per image  
Results saved to runs/detect/train

Resultados del entrenamiento

---

## Anexo 4. Conexión a la base de datos

---

### BITACORA NRO. 4

---

#### DATOS INFORMATIVOS

---

<b>Estudiante</b>	Ibeth Katherine Román	<b>Fecha:</b> 07/12/2023
<b>Tutor de proyecto</b>	Ing. Luis Rodríguez	

---

#### INFORMACIÓN GENERAL

---

<b>Nombre del proyecto</b>	Desarrollo de un prototipo basado en Visión por Computadora para detectar la invasión del carril de buses en la ciudad de Loja.
<b>Prueba</b>	Conexión con la base de datos

---

#### DETALLES

---

<b>Descripción</b>	Los datos recopilados por el software deben ser almacenados de manera que estén fácilmente disponibles cuando sea necesario.
<b>Problema</b>	Los datos son almacenados de forma local.
<b>Solución</b>	Subir los datos a una plataforma que permita su recuperación para futuras implementaciones.
<b>Observaciones</b>	Firestore se presenta como una solución viable debido a que permite la creación de la base de datos y el almacenamiento de las imágenes obtenidas del modelo.

---

#### CÓDIGO

---

```
#Librerias
import easyocr
from firebase import firebase
import firebase_admin
from firebase_admin import credentials, storage
from PIL import Image
import io
from datetime import datetime, timedelta
import pyrebase

# Inicializar la aplicación de Firebase con tus credenciales
cred = credentials.Certificate("serviceAccount.json")
firebase_admin.initialize_app(cred, {
    'storageBucket': 'deteccion-invasion.appspot.com'
})

firebase = firebase.FirebaseApplication("https://deteccion-invasion-default-rtdb.firebaseio.com/", None)
```

---

```

firebase1 = pyrebase.initialize_app(config)
db = firebase1.database()

# Obtener una referencia al bucket de Firebase Storage
bucket = storage.bucket()

#Carga de imagenes
image_path1 = 'captura_objeto_20240204202301479.png'
image_path = 'captura_placa_20240204202300900.png'
unique_number = datetime.now().strftime("%Y%m%d%H%M%S%f")[:-3]
nombre_imagen_firebase = f"placa_{unique_number}.png"

#Datos para subir
datos = {
    'id': new_id,
    'nombre_imagen': nombre_imagen_firebase,
    'imagen_url': url,
    'Placa': valor_placa,
    'Fecha': fecha_hora_actual_formateada,
}

#Llenar base de datos
resultado=firebase.post('/Resultados/Datos_post', datos)

```

---

## EVIDENCIAS

---



Resultados de la base de datos

---

### Anexo 5. Código completo

---

## BITACORA NRO. 5

---

### DATOS INFORMATIVOS

---

<b>Estudiante</b>	Ibeth Katherine Román	<b>Fecha:</b>
<b>Tutor de proyecto</b>	Ing. Luis Rodríguez	

---

### INFORMACIÓN GENERAL

---

<b>Nombre del proyecto</b>	Desarrollo de un prototipo basado en Visión por Computadora para detectar la invasión del carril de buses en la ciudad de Loja.
<b>Prueba</b>	Conexión con la base de datos

---

## CÓDIGO

---

---

**#Fecha de modificación: 11/02/2024 3:25**

**#Librerías Utilizadas**

```
from ultralytics import YOLO
from ultralytics.utils.plotting import Annotator
import matplotlib.path as mplPath
import time
from firebase import firebase
import firebase_admin
from firebase_admin import credentials, storage
from datetime import datetime, timedelta
import pyrebase
import easyocr
import cv2
import numpy as np
```

**# Configuración de base de datos**

```
cred = credentials.Certificate("serviceAccount.json")
firebase_admin.initialize_app(cred, {
    'storageBucket': 'deteccion-invasion.appspot.com'
})
```

**# ROI principal**

```
ROI = np.array([
    [560, 550],
    [1900, 550],
    [1900, 1900],
    [560, 990]
])
```

**# Delimitación de zona de invasión**

```
ZONA = np.array([
    [299, 104],
    [620, 143],
    [1153, 789],
    [296, 790]
])
```

**# Puntos para línea**

```
PUNTO_INICIO_LÍNEA = (300, 525)
PUNTO_FIN_LÍNEA = (950, 525)
```

**#Variables**

```
tipo=None
detecto=None
```

**#Función para obtener punto de bbox**

```
def obtener_centro(bbox):
    third_quarter_x = bbox[2] + (bbox[0] - bbox[2]) / 1.5
    third_quarter_y = bbox[1] + (bbox[3] - bbox[1]) / 1.5
    center_of_third_quarter = (third_quarter_x, third_quarter_y)
    return center_of_third_quarter
```

**#Función para verificar si el objeto esta dentro de la zona**

```
def validar_deteccion(xc, yc):
    return mplPath.Path(ZONA).contains_point((xc, yc))
```

**\*\*\*\*\*OCR\*\*\*\*\***

---

---

```

# Función para convertir caracteres
def convert_characters(text, conversion_dict):
    converted_text = ""
    for char in text:
        converted_text += conversion_dict.get(char, char)
    return converted_text

# Procesamiento de imagen
# Reconocer caracteres en la imagen
results1 = reader.readtext(binarizada)
print("Placa leida", results1)

# Reconocimiento de texto
for detection in results1:
    text = detection[1]
    formatted_text = license_complies_format(text)
    return formatted_text[:7]

# Función para la captura y procesamiento de video
def presentar():
    inicio_linea = np.array(PUNTO_INICIO_LINEA)
    fin_linea = np.array(PUNTO_FIN_LINEA)

    while True:
        captura_realizada = False
        # Intentar inicializar la captura de video
        cap = cv2.VideoCapture(0)
        #Graficar lineas
        cv2.polylines(img=frame, pts=[ZONA], isClosed=True, color=(255,
255, 255), thickness=4)
        cv2.line(frame, PUNTO_INICIO_LINEA, PUNTO_FIN_LINEA, color=(255,
255, 255), thickness=5)

        img = anotaciones.result()

        cv2.imshow('YOLO V8 Detection', img) # Mostrar la imagen resultante

        # Cerrar programa
        if cv2.waitKey(1) == 27:
            break

        cap.release()
        cv2.destroyAllWindows()
    else:
        print("Error al conectar. Intentando de nuevo en 5 segundos...")
        time.sleep(2) # Esperar 5 segundos antes de intentar nuevamente

# Modelo
model = YOLO("bestmil.pt")

# Iniciar el hilo de captura y procesamiento
presentar()

```

---

## EVIDENCIAS

---

▼ — -Nr1TKTprTxEuH1\_1Fr

Fecha: "2024-02-28 15:47:57"

Placa: "LBD5134"

id: 217

imagen\_url: <https://storage.googleapis.com/deteccion-invasion.>

nombre\_imagen: "captura\_placa\_20240228154753716.png"



**Anexo 6. Certificación emitida por la Unidad de Movilidad**

*Ing. Tania Verónica Vásquez Romero.*

**ING. ELECTRÓNICA Y DE TELECOM. DE LA DIRECCIÓN DE MOVILIDAD.**

**CERTIFICA:**

*Que, la Srta. IBETH ROMÁN, fue autorizada por la Dirección de Movilidad del Municipio de Loja, para la instalación de una cámara de video, en la parada del Sistema Intermodal de Transportación Urbana (SITU) ubicada en la Av. Universitaria y calle Alonso de Mercadillo, para el desarrollo de su tema de tesis denominado "DESARROLLO DE UN PROTOTIPO BASADO EN VISIÓN POR COMPUTADORA PARA DETECTAR LA INVASIÓN DEL CARRIL DE BUSES EN LA CIUDAD DE LOJA", sin embargo, para coadyuvar a temas de seguridad del sector, se solicitó que se realice la instalación de mencionada cámara en la parada ubicada en la Av. Manuel Agustín Aguirre y 10 de Agosto.*

*Certifico esto en honor a la verdad y autorizo a la interesada para que haga uso del presente documento para los fines pertinentes.*

*Loja, 22 de noviembre de 2023*



*Ing. Tania Vásquez Romero*

**INGENIERA ELECT. Y DE TELECOM. DE LA DMTT**

*Visto bueno,*



*Ing. Marlon Severino Maldonado*

**DIRECTOR DE MOVILIDAD**

## Anexo 7. Certificación de traducción del resumen

### Certificado de Traducción al Idioma Inglés

**Jenny Vicenta Sanmartín Loarte**, Licenciada en Ciencias de la Educación mención idioma Inglés, con C.I: 1105023772.

#### *Certifico:*

Que el texto traducido al idioma inglés que compone el **Resumen** del Trabajo de Titulación denominado: **“Desarrollo de una prototipo basado en visión por computadora para detectar la invasión del carril de buses en la ciudad de Loja”** correspondiente al **Srta. Ibeth Katherine Roman Carrión**, con número de cédula **1105736597**, fue realizado y verificado bajo mi supervisión.

Es todo en cuanto puedo indicar en honor a la verdad, facultando al interesado hacer uso del presente documento para los fines pertinentes.

Loja, 28 de noviembre de 2024



**Firma:** Lic. Jenny Sanmartín.

**C.I:** 1105023772

**Celular:** 0983546274

**Número de Registro Senescyt:** 1008-14-1269521