



Universidad del Azuay

Facultad de Ciencia y Tecnología

Escuela de Ingeniería Civil

**Aplicación de inteligencia artificial para la detección, conteo, clasificación y análisis del
flujo vehicular**

Trabajo de graduación previo a la obtención del título de:

INGENIERO CIVIL

Autores:

JOSUE LEONARDO BERMEO CABRERA

ARACELY PAOLA CÁRDENAS LOJA

Director:

DR. DIEGO CORREA BARAHONA

CUENCA ECUADOR

2025

AGRADECIMIENTO

Este proyecto representa el esfuerzo colectivo de quienes conformamos este grupo de titulación. Cada etapa recorrida fue una oportunidad para crecer, aprender y fortalecernos como equipo.

Durante este proceso entendimos que el camino se hace más llevadero cuando estás rodeado de personas comprometidas, con quienes compartir tanto los retos como las alegrías. Gracias a cada uno de ellos, este proyecto tomó forma y significado.

A la Universidad del Azuay y a sus docentes, nuestro reconocimiento por brindarnos una formación sólida y acompañarnos con paciencia, exigencia y compromiso a lo largo de estos años.

De manera especial, queremos agradecer al Dr. Diego Correa Barahona, director de nuestro proyecto, por su guía constante, sus valiosos aportes y su disposición para acompañarnos con profesionalismo y dedicación. Su confianza y acompañamiento han sido fundamentales para alcanzar los resultados que hoy presentamos con orgullo.

A nuestros amigos, que, con su apoyo incondicional, palabras de ánimo y presencia constante nos ayudaron a no rendirnos en los momentos más exigentes. Su compañía fue un sostén emocional que marcó la diferencia.

Y entre nosotros, compañeros de este proyecto, queremos reconocer el compromiso, la responsabilidad y el respeto mutuo con el que enfrentamos cada desafío. Este logro no hubiera sido posible sin la entrega y el trabajo de cada uno.

DEDICATORIA

Con el alma llena de gratitud, dedico este trabajo a quienes han sido mi fuerza, mi refugio y mi mayor inspiración a lo largo de este camino.

A mis amados padres, Leonardo y Verónica, por su amor inmenso, por cada sacrificio silencioso y por enseñarme, con su ejemplo, que todo se puede lograr con esfuerzo, humildad y perseverancia. Gracias por ser mi base, mi impulso y mi hogar.

A mi hermano Sebastián, por su cariño incondicional, por estar siempre a mi lado y por demostrarme que los hermanos son esos amigos que la vida nos regala para siempre. Gracias por tu apoyo constante y por ser parte esencial de este logro.

A mi familia, por cada palabra de aliento, cada abrazo y cada gesto que me recordó que nunca estoy solo.

A mis docentes, por compartir no solo su conocimiento, sino también su pasión y su guía durante este proceso.

A mis amigos y compañeros, por los momentos compartidos, las risas, los consejos y la compañía que hizo de este trayecto una experiencia inolvidable.

Josué L. Bermeo Cabrera.

DEDICATORIA

A Dios, por ser la fuente de mi fe, mi refugio en los momentos difíciles y la luz que ha guiado cada uno de mis pasos. Gracias por nunca soltar mi mano y por darme la fuerza necesaria para seguir adelante incluso cuando parecía imposible.

A mis padres, Yofre Cárdenas y Alicia Loja, con todo mi amor y profunda gratitud. Ustedes son mi mayor ejemplo de esfuerzo, valentía y entrega. Gracias por enseñarme, con hechos más que con palabras, el valor del trabajo honesto, la humildad, y la importancia de luchar por mis sueños sin rendirme. Cada logro que hoy celebro tiene sus raíces en los sacrificios que ustedes hicieron por mí. Este paso tan importante en mi vida es tan suyo como mío.

A mis hermanos, Maycol y Angie, por ser mi apoyo, mis confidentes y por compartir conmigo tantas etapas de la vida. Gracias por estar siempre, con una sonrisa, un consejo o simplemente su compañía que me reconforta y me hace sentir en casa, los amo mucho.

A mi abuelita María, que desde el cielo me cuida con amor infinito. Su ternura, su fortaleza y sus enseñanzas viven en mí. Su recuerdo me impulsa a ser una mejor persona, y dedico este logro con todo mi corazón a ella, porque sé que estaría orgullosa de este momento.

A mi pequeño primito Emilio, por llenar mis días de alegría con su sonrisa inocente y su luz. Eres una chispa de vida en nuestra familia, y verte crecer me inspira a ser un ejemplo para ti. Este logro también es para ti, con la esperanza de que un día persigas tus sueños con la misma determinación y fe.

Y a toda mi familia, gracias por su amor incondicional, por estar presentes en cada paso importante de mi vida y por ser ese soporte emocional que tantas veces me sostuvo sin siquiera pedirlo. En especial a mi tía Erika Cárdenas, por su cariño constante y su apoyo. A lo largo de los años, ella ha estado a mi lado, sin esperar nada a cambio. Este logro es también suyo, porque gracias a ella, siempre supe que no estaba sola en este camino.

Aracely P. Cárdenas Loja.

RESUMEN

Este trabajo de titulación presenta una solución basada en inteligencia artificial para la detección, conteo, clasificación y análisis del flujo vehicular en tiempo real mediante visión artificial y softwares de código abierto. Se desarrolló un algoritmo Python para identificar y clasificar motocicletas, vehículos livianos y pesados mediante el procesamiento de imágenes captadas por cámaras instaladas en la intersección de las avenidas Hernán Malo y Panamericana en Cuenca. El sistema incorpora técnicas de segmentación de imágenes, seguimiento de objetos y clasificación automática, adaptándose a las condiciones ambientales variables. Los resultados muestran una alta efectividad en la identificación y conteo de vehículos, lo que permite la generación de datos confiables para el análisis del comportamiento del tráfico. Este enfoque representa un aporte significativo al desarrollo de sistemas de transporte inteligentes, con potencial de aplicación en la planificación vial en ciudades intermedias del Ecuador.

Palabras claves: inteligencia artificial, visión por computadora, conteo vehicular, clasificación de vehículos, análisis de tráfico, Python.

ABSTRACT

This degree work presents a solution based on artificial intelligence for the detection, counting, classification and analysis of vehicular flow in real time using artificial vision and open source software. A Python algorithm was developed to identify and classify motorcycles, light and heavy vehicles by processing images captured by cameras installed at the intersection of Hernán Malo and Panamericana avenues in Cuenca. The system incorporates image segmentation techniques, object tracking and automatic classification, adapting to varying environmental conditions. The results show high effectiveness in vehicle identification and counting, allowing the generation of reliable data for traffic behavior analysis. This approach represents a significant contribution to the development of intelligent transportation systems, with potential application in road planning in intermediate cities of Ecuador.

Keywords: artificial intelligence, computer vision, vehicle counting, vehicle classification, traffic analysis, Python.

INDICE DE CONTENIDO

AGRADECIMIENTO	II
DEDICATORIA.....	III
DEDICATORIA.....	IV
RESUMEN.....	V
ÍNDICE DE FIGURAS	IX
ÍNDICE DE ECUACIONES	XII
ÍNDICE DE TABLA	XIII
OBJETIVOS.....	XIV
CAPITULO I.....	15
1.1. Introducción.....	15
1.2. Descripción del problema.....	15
1.2.1. Importancia del control del tráfico de vehículos	16
1.3. Alcance y limitaciones.....	16
1.3.1. Alcance del análisis para vehículos.	17
CAPITULO II.....	18
2.1. Conceptos de congestión del tráfico y su impacto.	18
2.2. Procesamiento de vídeo	18
2.3. Inteligencia Artificial y su aplicación en el tráfico.....	19
2.3.1. Redes neuronales y algoritmos de clasificación.	20
2.3.2. Redes neuronales convolucionales (CNN)	21
2.3.2.1 YOLO – You Only Look Once	23
2.3.2.2 YOLOv8n – You Only Look Once	24
2.4. Visión artificial	24
	VI

2.4.1. Reconocimiento y detección de objetos en vídeo.....	27
2.5. Tecnologías y herramientas utilizadas.....	28
2.5.1. OpenCV y Pytorch.	29
CAPITULO III	33
3.1 Arquitectura del proyecto	33
3.2 Revisión bibliográfica.....	34
3.3 Área de estudio	34
3.4 Instalación del sistema de video vigilancia	35
3.5 Recopilación de Datos	37
3.5.1. Software y versiones empleadas.....	38
3.5.2. Generación de la base de datos.....	39
3.6 Desarrollo de la red neuronal.....	46
3.6.1 Preparación del Dataset	48
3.6.2 Implementación de YOLOv8n	50
3.6.3 Entrenamiento y evaluación de la arquitectura.....	50
3.7 Procesamiento.....	60
3.7.1 Detección, conteo y clasificación vehicular	62
CAPITULO IV	67
4.1 Comparación de AlexNet y Yolov8n	67
4.2 Resultados de clasificación de vehículos.....	69
4.2.1 Conteo de vehículos en la intersección seleccionada	71
4.3 Evaluación de la precisión del algoritmo	79
4.3.1 Análisis de la efectividad del modelo de clasificación.....	92
CAPITULO V	95

5.1 Interpretación de los resultados	95
5.1.1 Factores que podrían afectar los resultados	95
CAPITULO VI	97
6.1 Conclusiones.....	97
6.2 Recomendaciones	99
Bibliografía.....	101
Anexos	106

ÍNDICE DE FIGURAS

Figura 1	Red neuronal	20
Figura 2	Arquitectura de AlexNet	22
Figura 3	Arquitectura de YOLO.....	23
Figura 4	Filtro Gaussiano de imágenes	25
Figura 5	Contorno de imagen	26
Figura 6	Sustracción de fondo.....	27
Figura 7	Logo de OpenCV	29
Figura 8	Logo de Tensor Flow	30
Figura 9	Logo de Pytorch.....	32
Figura 10	Arquitectura del Proyecto	33
Figura 11	Área de estudio seleccionado	35
Figura 12	Cámara	36
Figura 13	Instalación de cámara.....	36
Figura 14	Disco de estado sólido de 1TB.....	37
Figura 15	Logo de Python	38
Figura 16	Logo de entorno Anaconda.....	39
Figura 17	Interfaz gráfica de Python	39
Figura 18	Sustracción de fondo con MOG2.....	41
Figura 19	Operaciones morfológica-apertura.....	42
Figura 20	Contornos en vehículos.....	43
Figura 21	Rectángulo delimitador para la detección.....	43
Figura 22	Fotogramas recolectados.....	44
Figura 23	Numero de imágenes por categoría.....	45
Figura 24	Diagrama de flujo de la arquitectura de red neuronal.....	47
Figura 25	Redimensión 640 x 640	48
Figura 26	Etiqueta de imágenes	49
Figura 27	Iteración con 10 épocas de entrenamiento	51
Figura 28	Iteración con 20 épocas de entrenamiento	52
Figura 29	Iteración con 30 épocas de entrenamiento	52
Figura 30	Iteración con 50 épocas de entrenamiento	53

Figura 31	Métricas de evaluación	53
Figura 32	Matriz de confusión	54
Figura 33	Matriz de confusión normalizada.....	55
Figura 34	Curva precisión vs recall.....	57
Figura 35	Curva precisión vs confianza	57
Figura 36	Curva F1 vs confianza.....	58
Figura 37	Curva recall vs confianza	58
Figura 38	Visualización de predicciones.....	59
Figura 39	Modelo guardado en formato pt.....	60
Figura 40	Diagrama de flujo de la arquitectura del procesamiento.....	61
Figura 41	Zona de interés en el video (Hernán Malo y Panamericana)	62
Figura 42	Línea de conteo en el video (Hernán Malo y Panamericana)	63
Figura 43	Zona de interés en el video (Panamericana)	64
Figura 44	Línea de conteo en el video (Panamericana)	64
Figura 45	Esquema de flujos en la intersección	66
Figura 46	Resultados de grafica de precisión en el entrenamiento del modelo AlexNet...	67
Figura 47	Resultados de grafica de precisión en el entrenamiento del modelo YOLOv8n ..	68
Figura 48	Detección con etiquetas de clasificación vehicular (Hernán Malo y Panamericana)	69
Figura 49	Detección con etiquetas de clasificación vehicular (Hernán Malo y Panamericana)	70
Figura 50	Detección con etiquetas de clasificación vehicular (Panamericana)	70
Figura 51	Detección con etiquetas de clasificación vehicular (Panamericana)	71
Figura 52	Grafica de flujo en la intersección Hernán M y Panamericana (07:30-8:00am)	73
Figura 53	Grafica de flujo en la intersección Hernán M y Panamericana (17:30-18:00pm)	74
Figura 54	Pantalla final de video Hernán M y Panamericana, conteo IA-07:30 a 8:00 am.	75
Figura 55	Pantalla final de video Hernán M y Panamericana, conteo IA-17:30 a 18:00 pm.	75
Figura 56	Grafica de flujo vehicular en la calle Panamericana 07:30 a 8:00 am.....	77

Figura 57 Grafica de flujo vehicular en la calle Panamericana 17:30 a 18:00 pm.....	77
Figura 58 Pantalla final de video Panamericana con conteo IA de 07:30 a 08:00 am.....	78
Figura 59 Pantalla final de video Panamericana con conteo IA de 17:30 a 18:00 pm.....	78
Figura 60 Comparación entre conteo IA vs conteo manual Descenso Panamericana- Hernán Malo.....	82
Figura 61 Comparación entre conteo IA vs conteo manual Ascenso Hernán Malo- Panamericana.....	83
Figura 62 Comparación entre conteo IA vs conteo manual Panamericana.....	84
Figura 63 Comparación entre conteo IA vs conteo manual Descenso Panamericana- Hernán Malo.....	85
Figura 64 Comparación entre conteo IA vs conteo manual Ascenso Panamericana- Hernán Malo.....	86
Figura 65 Comparación entre conteo IA vs conteo manual Panamericana.....	87
Figura 66 Error presentado por el contador IA en la Hernán Malo y Panamericana de 07:30 a 08:00 am.	90
Figura 67 Error presentado por el contador IA en la Hernán Malo y Panamericana de 17:30 a 18:00 pm	91
Figura 68 Error presentado por el contador IA en la Panamericana de 07:30 a 08:00 am..	91
Figura 69 Error presentado por el contador IA en la Panamericana de 17:30 a 18:00 pm	92
Figura 70 Resultados del análisis de flujos vehiculares en dos horarios 07:30 a 08:00 am	93
Figura 71 Resultados del análisis de flujos vehiculares en dos horarios 17:30 a 18:00 pm	94

ÍNDICE DE ECUACIONES

Ecuación 1 : Función ReLU	21
Ecuación 2: Función softmax	22
Ecuación 3: Función gaussiana	25
Ecuación 4: Matriz Kernel.....	26
Ecuación 5: Precisión	56
Ecuación 6: Recall	56
Ecuación 7: F1-score	56
Ecuación 8: Flujo vehicular.....	66
Ecuación 9: Efectividad.....	88

ÍNDICE DE TABLA

Tabla 1	Comparación de la red neuronal AlexNet y YOLOv8n	68
Tabla 2	<i>Conteo vehicular por clase en la Intersección Hernán Malo y Panamericana horario de 07:30 a 8:00 am.....</i>	<i>72</i>
Tabla 3	Conteo vehicular por clase en la Intersección Hernán Malo y Panamericana horario de 17:30 a 18:00 pm.	72
Tabla 4	Conteo de vehículos por clase en la calle Panamericana de 07:30 a 8:00 am.	76
Tabla 5	Conteo de vehículos por clase en la calle Panamericana de 17:30 a 18:00 am.	76
Tabla 6	Resultados de conteo real de vehículos (video) en la intersección Hernán Malo y Panamericana de 07:30 a 08:00 am.	79
Tabla 7	Resultados de conteo real de vehículos (video) en la intersección Hernán Malo y Panamericana de 17:30 a 18:00 am.	80
Tabla 8	Resultados de conteo real de vehículos (video) en la calle Panamericana de 07:30-8:00 am.	80
Tabla 9	Resultados de conteo real de vehículos (video) en la calle Panamericana de 17:30-18:00 pm.....	81
Tabla 10	Comparación de resultados obtenidos en la intersección Hernán Malo y Panamericana 07:30 a 08:00 am.....	81
Tabla 11	Comparación de resultados obtenidos en la calle Panamericana	83
Tabla 12	Comparación de resultados obtenidos en la intersección Hernán Malo y Panamericana.....	85
Tabla 13	Comparación de resultados obtenidos en la calle Panamericana	86
Tabla 14	Resultados de efectividad del conteo automático en la intersección Hernán Malo y Panamericano tanto en descenso como ascenso de 07:30 a 08:00 am.	88
Tabla 15	Resultados de efectividad del conteo automático en la intersección Hernán Malo y Panamericano tanto en descenso como ascenso de 17:30 a 18:00 pm.	88
Tabla 16	Resultados de efectividad del conteo automático en la calle Panamericana de 07:30 a 08:00 am.	89
Tabla 17	Resultados de efectividad del conteo automático en la calle Panamericana de 17:30 a 18:00 pm.	89
Tabla 18	Resultados del análisis de flujos vehiculares en dos horarios	93

OBJETIVOS

Objetivo General

Desarrollar una solución basada en inteligencia artificial que permita la detección, conteo, clasificación y análisis integral del flujo vehicular, con el fin de mejorar la gestión del tránsito y optimizar los procesos de toma de decisiones en entornos urbanos.

Objetivos específicos

- Realizar una revisión bibliográfica de la literatura existente sobre el uso de algoritmos de visión e inteligencia artificial.
- Desarrollar e implementar un sistema de visión artificial para la detección y seguimiento automático de vehículos.
- Aplicar un algoritmo de inteligencia artificial para clasificar vehículos en tiempo real según su tipo.
- Redactar el proceso de desarrollo del sistema de inteligencia artificial, analizando los resultados obtenidos y su impacto en la optimización de la gestión del tráfico.

CAPITULO I

1.1. Introducción

En la actualidad, el crecimiento poblacional y del número de vehículos han generado importantes desafíos en el sistema vial. Según estimaciones recientes, el número de vehículos ha aumentado con una tasa de 1,78% anual. (Instituto Nacional de Estadísticas y Censos, 2024), lo que incrementa los niveles de congestión, los accidentes y el impacto ambiental son problemas comunes en la ciudad de Cuenca, que demandan soluciones innovadoras y tecnológicas. En este contexto, la Inteligencia Artificial (IA) surge como una de las herramientas principales para mejorar y transformar la manera en que se monitorea y analiza el tráfico vehicular optimizando la gestión del tráfico y contribuyendo a una planificación más efectiva de las infraestructuras.

En el presente trabajo investigativo propone desarrollar una solución basada en IA para una adecuada detección, conteo, clasificación y análisis del flujo vehicular, empleando tecnologías para recopilar y procesar grandes volúmenes de datos en tiempo real sobre vehículos. A través de la instalación de cámaras de video en la intersección Hernán Malo y Panamericana, la aplicación de un algoritmo estructurado en Python, con el objetivo de automatizar el proceso de conteo vehicular, eliminando la dependencia de métodos manuales que pueden ser ineficaces (Caiza, 2019). Con esta información, se podrá identificar patrones de circulación y puntos críticos de congestión, lo que contribuirá al desarrollo ordenado de las ciudades y mejorará la seguridad en las vías. La modernización de este proceso permitirá gestionar el tráfico de manera más eficiente, facilitando una planificación urbana más efectiva. De este modo, la integración de estas tecnologías en las intersecciones clave de la ciudad se convierte en un paso esencial hacia la creación de un entorno urbano más dinámico, seguro y sostenible (Ministerio de Transporte y Obras Públicas, 2023)

1.2. Descripción del problema

El tráfico vehicular se ha convertido en uno de los mayores problemas para la ciudad de Cuenca (Instituto Nacional de Estadísticas y Censos, 2024) ,impulsado por el crecimiento poblacional y el atractivo de la ciudad como patrimonio y estilo de vida. Este parámetro no sólo afecta a la movilidad de los ciudadanos, sino también a la calidad del aire y a la eficiencia

de las actividades económicas, provocando retrasos y aumentando el riesgo de accidentes y consumo excesivo de combustible.

Hasta donde se conoce, en Cuenca todavía no se ha implementado tecnologías basadas en IA y visión por computadora que permitan el análisis inteligente del tráfico vehicular. Las incorporaciones de estas herramientas resultan fundamentales para avanzar hacia un sistema de transporte eficiente y alineado con los principios de sostenibilidad y resiliencia. Estos estudios optimizan la movilidad urbana, reducen emisiones contaminantes y mejoran la seguridad vial.

1.2.1. Importancia del control del tráfico de vehículos

Un sistema de control adecuado no solo optimiza el flujo vehicular, sino que también reduce la congestión, mejora la seguridad vial y disminuye ruido, emisiones de gases contaminantes.

Al sustituir los métodos manuales por soluciones automatizadas, se minimizarán los errores asociados con la recolección de datos en condiciones ambientales variables y el tráfico intenso. Esta transición no solo permitirá obtener datos más precisos y en tiempo real, sino que también facilitará un análisis más exhaustivo del flujo vehicular, contribuyendo a la identificación de patrones y a la toma de decisiones informadas.

1.3. Alcance y limitaciones

El presente trabajo de titulación está enfocado en el diseño e implementación de un algoritmo de aprendizaje progresivo, para el procesamiento de imágenes digitales, optimizado para la detección, conteo, clasificación y análisis automatizado de vehículos en tiempo real. Se empleará software de código abierto, lo que nos permitiría reducir considerablemente a los costos asociados al desarrollo e implementación, sin comprometer la funcionalidad ni la precisión del sistema.

El algoritmo integrara técnicas de visión por computadora, segmentación de imágenes, garantizando un rendimiento óptimo en entornos dinámicos y diversos factores como iluminación, accesibilidad y las condiciones climáticas. Además, debe ser capaz de clasificar de manera precisa y eficiente el tráfico vehicular, teniendo la capacidad de diferenciar entre vehículos livianos y pesados en la intersección de estudio. Para garantizar el mejor de los rendimientos, es fundamental establecer una ubicación estratégica de la cámara,

considerando parámetros técnicos como la altura de instalación de la cámara, el campo de visión necesario para abarcar todos los carriles de la vía para evitar puntos ciegos, así como la orientación evitar obstrucciones visuales ocasionadas por sombras. Además, se deben garantizar las mejores condiciones de iluminación ya sea natural o artificial, especialmente en horarios nocturnos.

Por otro lado, se debe realizar un análisis previo del entorno donde se pueda identificar posibles obstáculos físicos que puedan bloquear la visibilidad de la cara.

Se utilizarán plataformas de código abierto para la implementación del proyecto, específicamente el lenguaje de programación Python en conjunto con la biblioteca Open para el procesamiento y adquisición de datos.

El uso de herramientas de código abierto en proyectos tecnológicos favorece tanto la innovación como la colaboración en comunidades académicas y profesionales, lo que refuerza el potencial de este proyecto como una o plataforma de desarrollo continuo (Urrego *et al.*, 2009).

1.3.1. Alcance del análisis para vehículos.

El análisis estará enfocado a obtener información precisa sobre el flujo vehicular en una intersección de la ciudad de Cuenca, priorizando el conteo y clasificación entre vehículos livianos y pesados. Este enfoque permitirá evaluar directamente los patrones de tráfico en momentos específicos, facilitando la identificación de puntos críticos de congestión y ofreciendo una base sólida para futuras propuestas de mejora de la movilidad urbana.

Un aspecto relevante del alcance es la capacidad del sistema para operar en tiempo real, lo que permite responder rápidamente ante cambios dinámicos en el flujo vehicular. Sin embargo, también se contempla la generación de datos históricos, lo que posibilita estudios más profundos de patrones de tráfico a lo largo del tiempo.

CAPITULO II

Capítulo 2: Marco teórico

2.1. Conceptos de congestión del tráfico y su impacto.

El crecimiento demográfico y económico ha aumentado la demanda de las redes de carreteras, que deben soportar volúmenes cada vez mayores de transporte de personas y mercancías. A nivel global, la capacidad de estas infraestructuras se supera en las horas pico, cuando el flujo de vehículos alcanza su punto máximo. A pesar de los efectos negativos de la congestión, como la contaminación, la reducción de la capacidad de las carreteras, los atascos y la variabilidad en los tiempos de viaje, la propiedad de automóviles sigue aumentando cada año (Agencia Europea de Medio Ambiente, 2019). Esta creciente presión sobre las redes de transporte representa un desafío constante para las autoridades viales, los planificadores y los ingenieros de tránsito, quienes deben desarrollar estrategias efectivas para mitigar la congestión y mejorar la movilidad urbana.

La congestión vehicular en entornos urbanos afecta la movilidad negativamente a diversos sectores de la sociedad, reduciendo la calidad de vida y dificultando el comercio (Campos, 2011). Además, impactos como la contaminación acústica, accidentes viales y la salud publica especialmente en enfermedades cardiorrespiratorias, el principal efecto en los usuarios es el aumento en tiempos de viaje y su variabilidad.

2.2. Procesamiento de vídeo

El procesamiento de video es una disciplina dentro del campo de la visión por computadora y el procesamiento de imágenes que se enfoca en la manipulación, análisis e interpretación de secuencias de imágenes en movimiento. Su aplicación abarca diversas áreas, como seguridad, control de tráfico, automatización industrial y medicina, entre otras (Zurita *et al.*, 2024).

El procesamiento de vídeo implica la transformación y análisis de datos visuales con el objetivo de extraer información relevante. Los conceptos clave en este campo incluyen:

- ❖ **Adquisición de video:** Se refiere a la captura de imágenes en secuencia a través de cámaras digitales o sensores especializados.

- ❖ **Preprocesamiento:** Incluye técnicas como conversión de escala de grises, filtrado de ruido, mejora del contraste y normalización de la iluminación para facilitar la interpretación de la imagen.
- ❖ **Segmentación:** Consiste en la identificación y separación de objetos dentro del vídeo mediante algoritmos como umbralización, detección de bordes y segmentación basada en movimiento.
- ❖ **Seguimiento de objetos:** utiliza técnicas como el flujo óptico o filtros de Kalman para estimar la trayectoria de un objeto en movimiento dentro de una secuencia de vídeo.
- ❖ **Extracción de características:** se centra en la identificación de propiedades relevantes de los objetos detectados, como forma, color, textura o patrones de movimiento.
- ❖ **Clasificación y reconocimiento:** Aplicación de modelos de inteligencia artificial y aprendizaje automático para categorizar los objetos detectados y diferenciarlos según su naturaleza.

En el contexto de esta investigación se utilizará el procesamiento de video para la detección, conteo y clasificación de vehículos en la intersección Hernán Malo y Panamericana.

El uso del procesamiento de vídeo en el análisis del flujo de vehículos permite una evaluación automatizada y precisa del tráfico, contribuyendo a la planificación urbana y a la optimización de los tiempos de los semáforos en la ciudad (Rosero, 2019).

2.3. Inteligencia Artificial y su aplicación en el tráfico

La Inteligencia Artificial (IA) ha transformado significativamente diversos sectores y el tráfico vehicular no es una excepción (Coello , 2024) . Su aplicación en el análisis y gestión del tráfico ha permitido optimizar la movilidad urbana, mejorar la seguridad vial y reducir el impacto ambiental de la congestión vehicular (Diaz, 2024). Este capítulo discutirá los principales enfoques de la IA aplicada al tráfico, así como sus beneficios y desafíos.

La IA se utiliza en varias soluciones de gestión del tráfico. Algunas de las aplicaciones más relevantes incluyen:

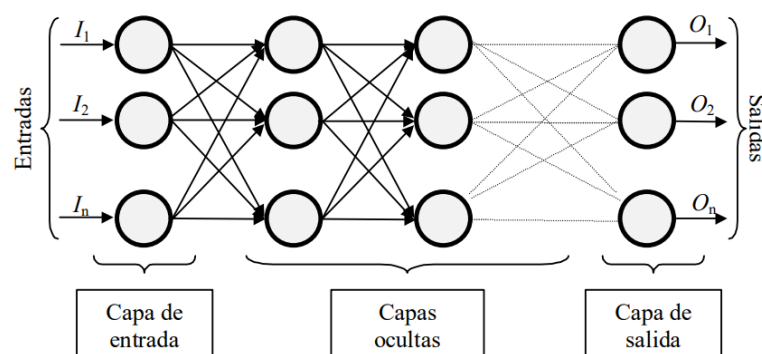
- ❖ **Sistemas Inteligentes de Control de Tráfico:** Utilizan algoritmos de aprendizaje automático para ajustar el tiempo de los semáforos en tiempo real, en función del flujo de tráfico, reduciendo así los tiempos de espera y la congestión.
- ❖ **Detección y clasificación de vehículos:** a través de visión por computadora y redes neuronales, los sistemas pueden identificar y clasificar vehículos en intersecciones y carreteras, proporcionando datos precisos para el análisis del tráfico.
- ❖ **Predicción del flujo de vehículos:** los modelos de IA analizan patrones de tráfico históricos y condiciones climáticas para predecir posibles atascos, lo que permite a los conductores y a las autoridades tomar decisiones informadas.

2.3.1. Redes neuronales y algoritmos de clasificación.

Las redes neuronales artificiales han demostrado ser herramientas muy efectivas en el campo del aprendizaje automático y la inteligencia artificial (Pino *et al.*, 2001), especialmente en tareas de clasificación. Su estructura está inspirada en el funcionamiento del cerebro humano, utilizando capas de neuronas interconectadas para procesar datos y extraer patrones complejos. A continuación, se evidencia en la figura 1, un esquema de los elementos básicos en la red neuronal.

Figura 1

Red neuronal



Nota. La figura muestra el esquema de los elementos básicos de la red neuronal. Fuente:

(Matich, 2001).

Existen varios algoritmos de clasificación que utilizan redes neuronales para identificar y categorizar datos. Algunos de los más utilizados incluyen:

- ❖ **Redes Neuronales Artificiales (ANN):** Modelos básicos con múltiples capas totalmente conectadas, ideales para problemas generales de clasificación.
- ❖ **Redes Neuronales Convolucionales (CNN):** Especializadas en procesamiento de imágenes, útiles en visión por computadora para reconocimiento de patrones y objetos.
- ❖ **Redes neuronales recurrentes (RNN):** Utilizados para datos secuenciales, como series temporales o reconocimiento de texto.
- ❖ **Perceptrón multicapa (MLP):** Un tipo de ANN con múltiples capas ocultas y funciones de activación no lineales.

2.3.2. Redes neuronales convolucionales (CNN)

Las redes neuronales convolucionales disponen de múltiples capas ocultas para la identificación de curvas, líneas, texturas y formas de los objetos, es una arquitectura de redes neuronales diseñadas para trabajar con una extensa base de datos de imágenes (Coanqui *et al.*, 2022). Las CNN están compuestas de capas de entrada y salida para el respectivo aprendizaje de las imágenes, las capas están compuestas por varios elementos, los cuales se detallan a continuación:

Capa convolucional: Esta capa realiza operaciones en las imágenes de entrada, permitiendo identificar las texturas, bordes, formas de las imágenes, el filtro Kernels se usa para extraer estas características de entrada.

Función ReLU: Es una función de activación en las redes neuronales, es decir son el “cómo piensan” las neuronas utilizado para la aplicación de redes neuronales y se muestra en la ecuación 1. Esta función es eficiente debido a que solo requiere una comparación y realiza el paso de todos los valores positivos, mientras que cambia a cero todos aquellos valores negativos (Llerena, 2019).

Ecuación 1 : Función ReLU

$$\varphi(v) = \max(0, v)$$

Capa de pooling: Esta capa realiza la reducción de tamaño de las imágenes a través de operaciones Max-pooling, ayuda a que el modelo sea más robusto y eficiente.

Finalmente es portante mencionar las funciones sigmoideas que eliminan elementos abstractos que pueden afectar a los parámetros y hiperparámetros, dentro de esta se encuentra la función softmax conocida como exponencial normalizada que re-escala las n dimensiones de la entrada en un rango de 0 a 1 (Bilbao, 2022), evidenciado en la ecuación 2.

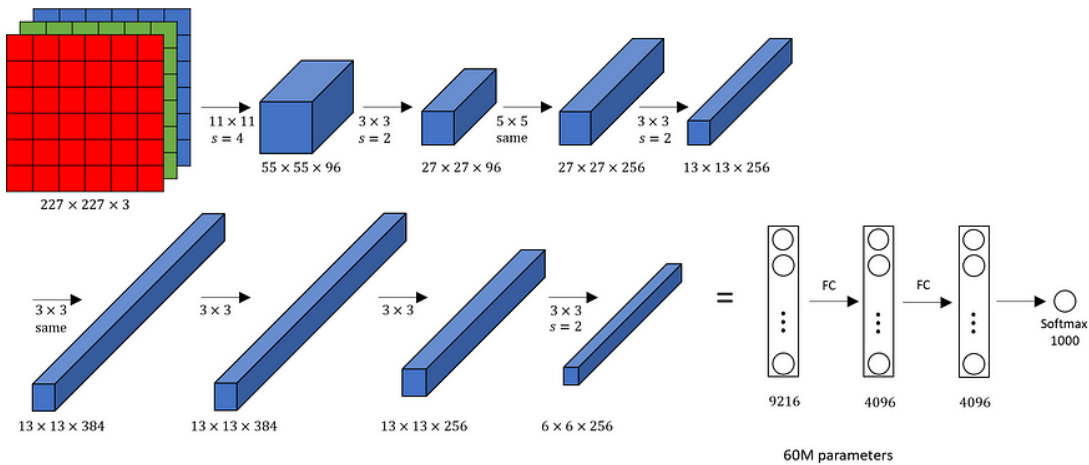
Ecuación 2: Función softmax

$$S(xi) = \frac{\exp(xi)}{\sum_j \exp(xj)}$$

Dentro del campo de las redes neuronales convencionales CNN, AlexNet ha sido un modelo fundamental en la evolución de las CNN. AlexNet marcó un hito en el reconocimiento de imágenes al ganar el desafío ImageNet con una precisión sin precedentes en ese momento. Su arquitectura caracterizada por tener múltiples capas convolucionales (Maeda *et al.*, 2018), seguidas de capas completamente conectadas. En el contexto del recuento y clasificación de vehículos, las redes neuronales han mejorado la precisión en la identificación de diferentes tipos de vehículos a partir de imágenes y vídeos. Utilizando CNN como AlexNet evidenciado en la figura 2, es posible extraer características avanzadas como forma, tamaño y color.

Figura 2

Arquitectura de AlexNet



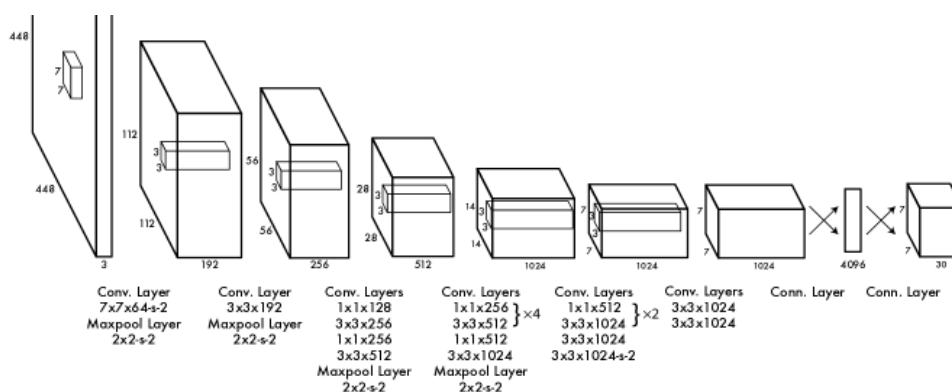
Nota. La figura muestra detalladamente la distribución de AlexNet. Fuente: (Picazo

Montoya, 2018)

Además, como se muestra en la figura 2, la capa de entrada tiene como dimensión: 227x227x3,5 capas convolucionales,3 redes totalmente conectadas y una salida de la capa final con 1000 salidas, estas entrenan redes neuronales con grandes volúmenes de datos permite optimizar la precisión del sistema, minimizando errores en la clasificación y mejorando la eficiencia del análisis del tráfico urbano (Pino *et al.*, 2001).

Dentro del campo de las CNN, YOLO ha sido un modelo fundamental en la evolución de las redes neuronales convolucionales. Fue propuesto por J. Redmon y se caracteriza por ser eficiente y preciso en la detección de objetos ya que tiene una gran capacidad para realizar la predicción de manera precisa, hace la tarea en un solo paso, lo que genera menos tiempo de procesamiento

Figura 3



Divvala, Girshick, & Farhadi, 2016)

Además, como se muestra en la figura 3, consta de 24 capas convolucionales, la capa de entrada tiene como dimensión: 448x448x3 capas convolucionales y una salida de la capa final que predice coordenadas (Redmon *et al.*, 2016), el modelo como YOLO han demostrado ser eficaces en tareas de reconocimiento visual de alto rendimiento, conteo y clasificación vehicular en tiempo real lo que lo convierte en una opción valiosa para aplicaciones de movilidad y transporte inteligentes.

2.3.2.2 YOLOv8n – You Only Look Once

YOLOv8n es una versión mejorada de YOLO, diseñada para la detección de objetos en términos de velocidad y precisión, desarrollado por Ultralytics, este modelo ofrece mejoras en la detección de los objetos como la eliminación de anclas para la predicción del centroide de los objetos y aumento de mosaico durante el entrenamiento lo que hace que el modelo no generalice las imágenes, si no que aprenda características diferentes de los objetos como las ubicaciones y ángulos (Coanqui *et al.*, 2022).

El modelo YOLOv8n puede ser utilizado en una amplia gama de aplicaciones, gracias a su arquitectura optimizada en comparación con versiones anteriores, siendo ampliamente utilizado para realizar el conteo y clasificación de vehículos.

2.4. Visión artificial

La visión artificial comprende un conjunto de técnicas y procesos destinados a la adquisición, procesamiento y análisis de imágenes mediante sistemas informáticos. Esta disciplina busca replicar la capacidad humana para interpretar datos visuales, abordando problemas complejos a partir de una representación bidimensional, como puede ser una imagen, se pretende inferir información tridimensional sobre la escena capturada. Este proceso incluye la reconstrucción de estructuras espaciales, la identificación de objetos y patrones y el análisis del movimiento. (Loaiza *et al.*, 2012).

Además de su aplicación en el tráfico, la visión artificial es importante en campos como la medicina, ya que ayuda a la detección de anomalías en imágenes médicas, los sistemas de vigilancia y en la automatización industrial con la inspección de calidad. Finalmente, la complejidad de la visión artificial también radica en la necesidad de interpretar correctamente las variaciones de luz, perspectiva y ruido en las imágenes, utilizando algoritmos sofisticados para garantizar un análisis preciso.

La visión artificial consta de elementos que son importantes para su configuración, como un sensor óptico que permite captar las imágenes mediante una cámara de video y de un computador que permite la creación y ejecución de los algoritmos de visión artificial. También consta de áreas principales para su implementación como adquisición y representación en el cual las imágenes se encuentran compuestas por pixeles que desarrollan la información importante como el color, modelo de color, profundidad de color y el brillo de una región delimitada de aquellas imágenes.

En el filtrado se realizan cambios y operaciones a las imágenes que pasan de ser analógicas a digitales, destacando las operaciones sobre pixeles, sobre el histograma, morfológicas y en imágenes con ruido o condiciones de iluminación difíciles, se pueden aplicar filtros como el filtro gaussiano, es decir se aplica una convolucion para difuminar y reducir el ruido de las imágenes (Guaman Tisalema, 2015) ,como se muestra en la ecuación 3 y en la figura 4.

Ecuación 3: Función gaussiana

$$G\sigma(x,y) = e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

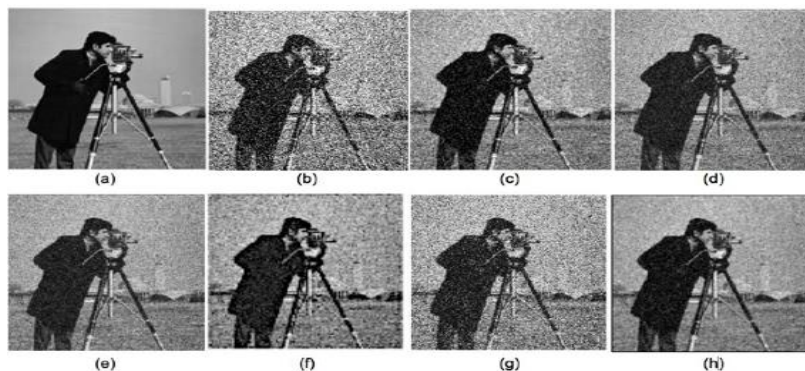
Donde:

$G\sigma(x,y)$ = Es una función gaussiana en dos dimensiones

σ = Varianza para x, y

Figura 4

Filtro Gaussiano de imágenes



Nota. Fuente: (Guaman Tisalema, 2015)

También incluye segmentación el cual mediante una imagen se genera otra en la que se tiene una etiqueta referencial que lo distinga y reconocimiento de patrones que es primordial disponer de las características importantes de las imágenes, permitiendo identificar objetos, formas, texturas, contornos o cualquier otra información relevante evidenciado en la figura 5.

Figura 5

Contorno de imagen



Nota. Fuente: (De los Santos, 2016)

Dentro de la visión artificial, el filtrado para el procesamiento de imágenes ayuda a eliminar el ruido. La matriz Kernel es una estructura en el procesamiento de imágenes que permite realizar convoluciones y filtrado en los píxeles de fotogramas, ayuda a resaltar y suavizar los bordes, aplicando operaciones matemáticas logra transformar la imagen original mejorando la misma (Cueva *et al.*, 2017). En la ecuación 4 se evidencia la matriz con un tamaño de 5x5 o 3x3.

Ecuación 4: Matriz Kernel

$$K = \frac{1}{\text{Kalto} \times \text{Kancho}} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Donde:

K=Matriz Kernel

Kalto=N° filas de la matriz

Kancho= N° columnas de la matriz

Las operaciones morfológicas se aplican utilizando la matriz kernel el cual define los pixeles en los fotogramas, las más comunes son, la erosión la cual reduce el tamaño de los objetos en movimiento, mientras que la dilatación aumenta el tamaño, la combinación de ambas mejoran la calidad de las imágenes. (Cueva *et al.*, 2017).

2.4.1. Reconocimiento y detección de objetos en vídeo.

El reconocimiento y detección de objetos es clave en la visión artificial, permitiendo identificar ,rastrear, clasificar elementos en movimiento dentro de una secuencia de imágenes , utilizando algoritmos de aprendizaje automático y redes neuronales convolucionales (CNN) , que realizan el análisis de cada fotograma del video y extraen características importantes, tecnologías como AlexNet han revolucionado la detección en tiempo real, mejorando la precisión y reduciendo el tiempo de procesamiento. (Maeda, *et al.*, 2018).

La detección de objetos de video tiene aplicaciones clave en la video vigilancia inteligente, donde se puede identificar actividades sospechosas en tiempo real. Es por ello importante mencionar una técnica para la detección de objetos denominada Background subtraction o sustracción de fondo, evidenciada en la figura 6, el cual separa los objetos en movimiento del fondo estático generando una máscara donde dichas áreas cambiantes representan movimiento, es un modelo que se adapta al seguimiento real con las distribuciones de la función kernel 3x3 a 5x5 por cada pixel de fondo. (Cueva *et al.*, 2017).

Figura 6

Sustracción de fondo



Nota. Fuente: Elaboración propia

La detección y el análisis del tráfico mediante IA y visión por computadora contribuyen a la optimización del flujo de vehículos y la reducción de la congestión. El conteo y la clasificación de vehículos son esenciales para comprender el flujo de tráfico ya que permite no sólo contar vehículos, sino también clasificarlos en diferentes categorías (vehículos, camiones, motos, etc.).

Por lo tanto, para dar un correcto funcionamiento del algoritmo estructurado se debe tener en cuenta las recomendaciones brindadas por la norma técnica ecuatoriana (Norma técnica ecuatoriana INEN 2656, 2016), la cual establece las definiciones:

- a) Vehículo liviano: es aquel vehículo automotor tipo automóvil o derivado de éste, diseñado para transportar hasta 12 pasajeros.
- b) Vehículo mediano: es aquel automotor cuyo peso bruto es menor o igual a 3860kg o cuyo peso neto es menor o igual a 2724kg, o cuya área frontal no excede de 4,18m². Este vehículo debe estar diseñado para el transporte de carga y el transporte de más de 12 pasajeros y ser utilizado fuera de carreteras o autopistas y contar para ello con características especiales.
- c) Vehículo pesado: es aquel vehículo automotor cuyo peso bruto del vehículo sea superior a 3860kg, o cuyo peso neto del vehículo sea superior a 2724kg, o cuya área frontal exceda de 4,18m². (NTE INEN 2207, 2016).

2.5. Tecnologías y herramientas utilizadas.

Las tecnologías han revolucionado el mundo, gracias a su conjunto de conocimientos, herramientas y de procesos que permiten crear soluciones innovadoras para mejorar la calidad de vida, esta abarca desde la creación de simples mecanismos hasta el desarrollo de inteligencia y visión artificial ,su gran aporte ha sido a la comunicación, industria, medicina, educación, ingeniería (Cascajares Ruperez, 2021), generando de esta manera cambios profundos en la sociedad ,facilita la automatización de tareas ,optimiza los recursos y permite el acceso inmediato a la información. Además, impulsa la investigación científica, acelerando el descubrimiento en distintas disciplinas y brindando nuevas oportunidades laborales.

La tecnología es de acceso inmediato y desde cualquier parte del mundo, simplemente el usuario accede empleando un navegador web, encuentra información actualizada y en el

campo de la ingeniería ha evolucionado significativamente en los últimos años, especialmente con el auge de la visión artificial y la IA, estas herramientas están cambiando la manera en que construye, diseña y se mantiene las infraestructuras, permitiendo un análisis más eficiente. Las implementaciones de estas herramientas innovadoras están mejorando la toma de decisiones y optimizando los recursos. (Terrones *et al.*, 2024).

Dentro de este marco, se utilizará Python como lenguaje de programación principal debido a su amplia adopción en el desarrollo de aplicaciones, también se emplearán bibliotecas de código abierto utilizadas en el entorno de Python como Pytorch para la IA y OpenCV para la visión artificial. Ambas tecnologías pueden trabajar conjuntamente, ya que Open CV se basa en el procesamiento de imágenes mientras que Pytorch es la que se centra en el aprendizaje automático y entrenamiento de modelos. Juntas estas herramientas realizan aplicaciones avanzadas en el campo de la inteligencia y visión artificial.

2.5.1. OpenCV y Pytorch.

OpenCV es una biblioteca de software de código abierto enfocada en la visión por computadora y el aprendizaje automático, su propósito es ofrecer una infraestructura común para aplicaciones de visión por computadora y facilitar la integración de la percepción automática en productos comerciales, OpenCV permite a individuos y empresas utilizar y modificar el código de manera sencilla. (Cuzco Cualchi, 2020).

Figura 7

Logo de OpenCV



Nota. Fuente: (OpenCV.org, 2016)

OpenCV contiene distintos tipos de lenguaje de programación como: Python, Java, C++, C y las principales actividades se enfatizan en el procesamiento de imágenes basadas en tiempo real, sus capacidades incluyen capturar videos e imágenes, procesarlos y generar nuevos

archivos con características modificadas. Además, esta biblioteca virtual está organizada de manera modular, ofreciendo distintos componentes que desempeñan funciones específicas. Algunos de estos módulos son especialmente relevantes para diversas tareas en el procesamiento de imágenes:

- ❖ **Módulo Core:** Proporciona funciones básicas para operaciones matemáticas y manejo de matrices.
- ❖ **Módulo Features 2D:** Especializado en la detección, descripción y emparejamiento de características en imágenes.
- ❖ **Módulo Highgui:** Facilita la visualización de imágenes y videos, además del manejo de interfaces graficas simples.
- ❖ **Módulo Calib3b:** Enfocado en la calibración de cámaras y la reconstrucción tridimensional.
- ❖ **Módulo Imgproc:** Incluye herramientas avanzadas para el procesamiento y análisis de imágenes.

Por su parte, Tensor Flow es una biblioteca de código abierto para computación numérica mediante gráficos de flujo de datos, en estas gráficas, los nodos representan operaciones matemáticas y los bordes simbolizan matrices de datos, conocidas como tensores ya que esta plataforma es ideal para desarrollar redes neuronales, permitiendo identificar patrones y correlaciones similares a cómo los humanos aprenden. (Tito *et al*, 2023).

Figura 8

Logo de Tensor Flow



Nota. Fuente: (TensorFlow.org, 2015)

Tensor Flow es aquel que tiene la opción de probar nuevos algoritmos avanzados de entrenamiento y de mejora ,la cual a pesar de ser una técnica relativamente reciente ,ha ganado gran popularidad en los últimos años , ya que se destaca por su clasificación de grandes volúmenes de datos , ofreciendo de esta manera resultados precisos y en un tiempo reducido ,además una de las grandes ventajas de Tensor Flow es que permite ejecutar cálculos en distintos entornos heterogéneos sin requerir de grandes modificaciones ,ya que puede utilizarse tanto en móviles ,como smartphones y tablets, como en sistemas de computación de gran escala . (Falconí Punguil, 2021).

Pytorch es una biblioteca de código abierto desarrollado por Facebook AI Research (FAIR) que se ha convertido en una de las herramientas más utilizadas en el ámbito del aprendizaje profundo. Esta biblioteca proporciona una interfaz flexible y dinámica basada en Python, lo que facilita la implementación y experimentación con redes neuronales (Mazzola, 2021). Su arquitectura orientada a gráficos computacionales dinámicos permite que los investigadores modifiquen estructuras de modelos en tiempo real (Rodríguez R. , 2004), a diferencia de enfoques más estáticos como Tensor Flow en sus primeras versiones.

Una de las principales características de Pytorch es su capacidad para trabajar con tensores multidimensionales, los cuales representan los datos en estructuras matemáticas optimizadas para el cálculo numérico. Estos tensores pueden aprovechar la aceleración por GPU (Unidad de Procesamiento Gráfico) (García, 2023), lo que permite realizar cálculos de alto rendimiento en aplicaciones como el procesamiento de imágenes y el análisis de datos masivos (Rodríguez & Pérez, 2024) . Además, Pytorch ofrece operaciones eficientes que permiten la manipulación y transformación de datos con facilidad, facilitando así la implementación de modelos complejos.

Otro aspecto relevante de Pytorch es su módulo Torch. nn, que proporciona herramientas para construir modelos de aprendizaje profundo a través de capas predefinidas, funciones de activación y mecanismos de optimización. En este sentido, Pytorch simplifica la implementación de arquitecturas de redes neuronales como convolucionales (CNNs), recurrentes (RNNs) y transformadores (Ortiz & Flores, 2024), las cuales han demostrado ser altamente efectivas en tareas como el reconocimiento de imágenes y la generación de lenguaje natural.

Figura 9

Logo de Pytorch



Nota. Fuente: (Pytorch.org, 2015)

CAPITULO III

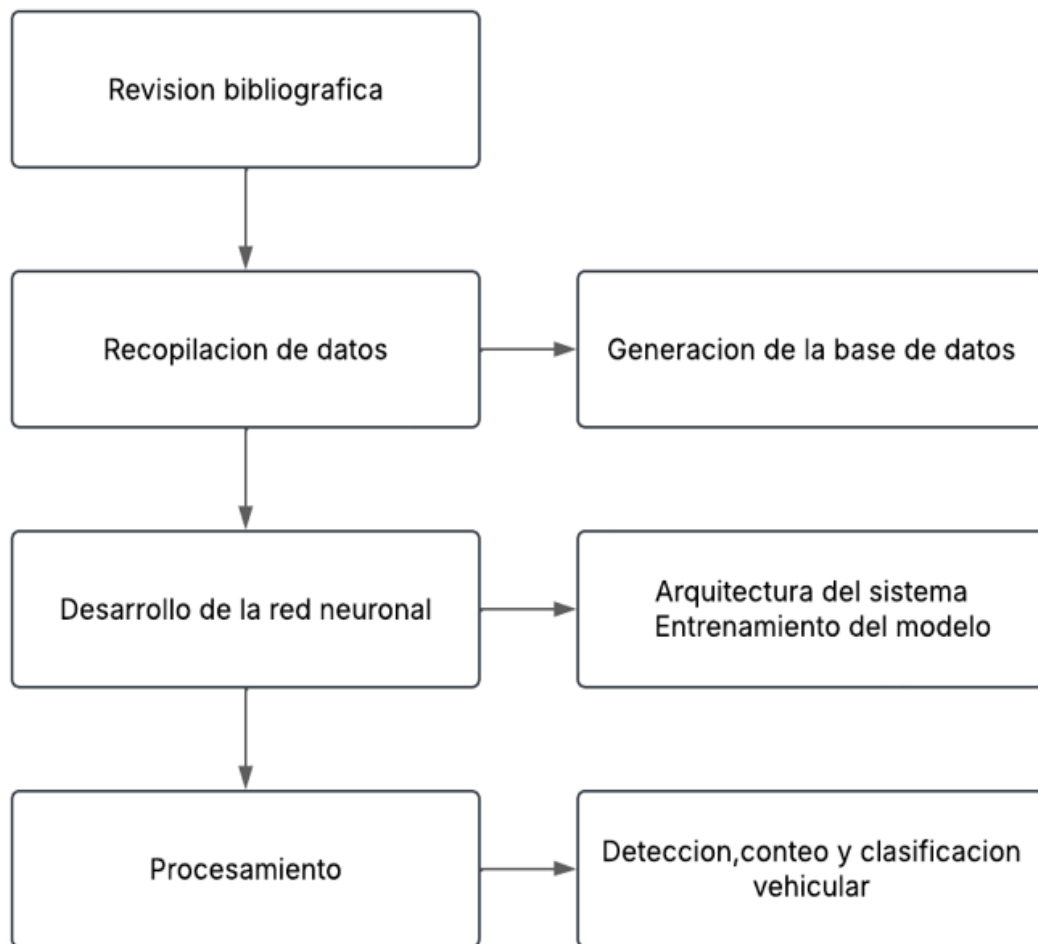
Capítulo 3: Metodología

3.1 Arquitectura del proyecto

La arquitectura del proyecto se compone de fases bien definidas que desempeñan un papel fundamental para lograr la aplicación de inteligencia artificial para la detección, conteo, clasificación y análisis del flujo vehicular, garantizando un procesamiento eficiente para obtener resultados precisos y fiables. A continuación, se evidencia en la figura 10 la arquitectura del proyecto.

Figura 10

Arquitectura del Proyecto



Nota. La figura muestra las fases que definen el proyecto. Fuente: Elaboración propia

3.2 Revisión bibliográfica

La revisión de literatura es importante en cualquier investigación, ya que permite comprender y examinar el conocimiento sobre el tema de estudio. En el caso de esta investigación, se exploraron antecedentes y estudios previos en bases de datos académicos, relacionados con la aplicación de inteligencia artificial en la detección, conteo, clasificación y análisis del flujo vehicular.

La revisión se centró en temas relacionados con el uso de OpenCV para la visión artificial y Pytorch para la inteligencia artificial, la implementación de redes neuronales profundas y específicamente en la arquitectura Yolo para la clasificación de imágenes. Además, se seleccionaron artículos relevantes para la detección y clasificación de vehículos mediante videos.

La presente investigación se enmarca dentro de un enfoque descriptivo y experimental, debido a que busca analizar el impacto de la aplicación de inteligencia artificial en la detección, conteo, clasificación y análisis del flujo vehicular. La fase descriptiva permitirá documentar el estado actual de las herramientas tecnológicas utilizadas en el análisis vehicular, identificando ventajas y limitaciones. A través de esta fase se examinarán estudios previos relacionados con la IA, visión artificial y redes neuronales.

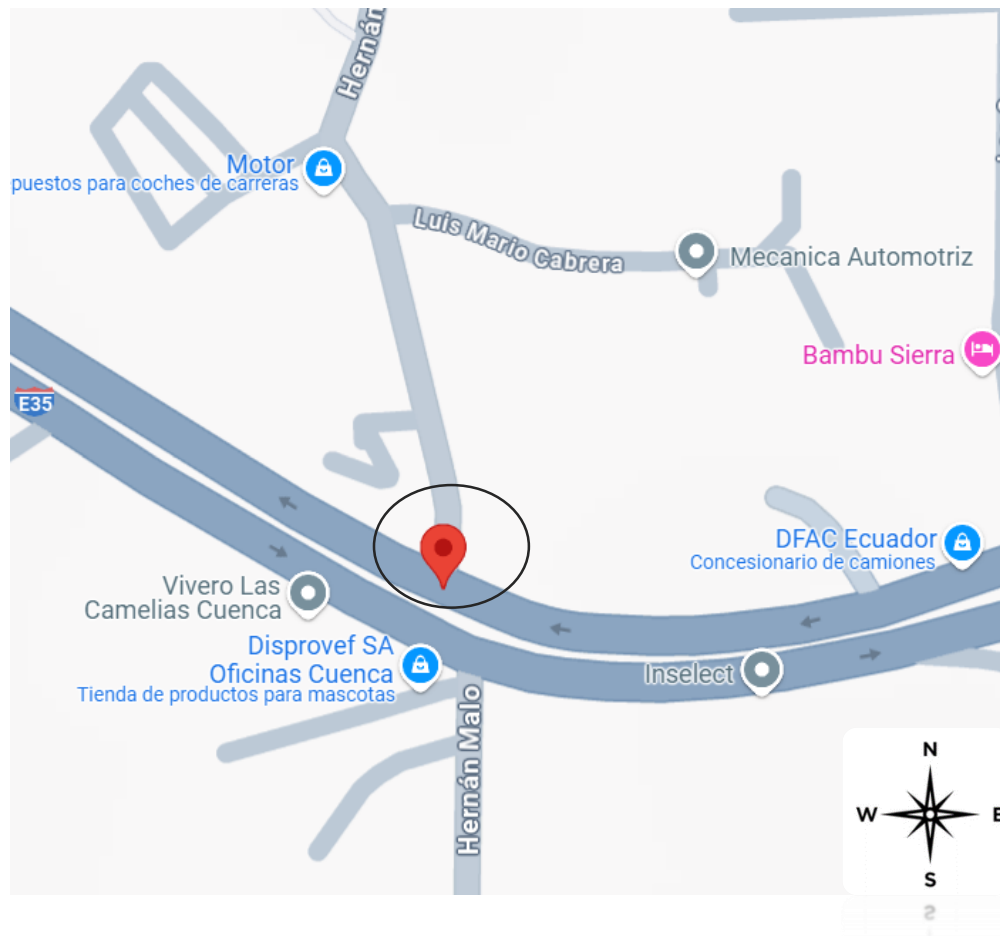
Por otra parte, el enfoque experimental será importante para evaluar el desempeño de la IA en la detección y clasificación vehicular en tiempo real, se implementará un algoritmo basado en aprendizaje automático para determinar la precisión y eficiencia del sistema en distintas condiciones de tráfico, con el fin de optimizar el rendimiento del sistema.

3.3 Área de estudio

La población objeto de estudio en este proyecto abarca la totalidad del parque automotor de Cuenca, en la provincia de Azuay, debido que, en los últimos años, se ha observado un considerable aumento del parque automotor, (Instituto Nacional de Estadísticas y Censos, 2024). No obstante, el enfoque se centra en el análisis del tráfico en una de las áreas con mayor afluencia vehicular de la ciudad, la zona seleccionada para la captura de imágenes a través del sistema de video es la intersección Hernán Malo y Panamericana, como se puede observar en la figura 11.

Figura 11

Área de estudio seleccionado



Nota. Fuente: Google maps

La zona seleccionada Hernán Malo y Panamericana se justifica por su estratégica ubicación cerca de la Universidad del Azuay, lo que genera alto flujo de estudiantes y personal académico, aumentando significativamente la afluencia vehicular en el área. Además, esta intersección es un punto clave para vehículos que se dirigen tanto al centro de la Ciudad de Cuenca como a las zonas periféricas, presentando un flujo vehicular alto. La combinación de estos factores convierte a esta intersección en un punto crítico para el análisis del tráfico.

3.4 Instalación del sistema de video vigilancia

Para este proyecto, la cámara utilizada es un dispositivo de alta resolución, capaz de capturar imágenes y videos en tiempo real para la base de datos del sistema de IA. Se trata de una

cámara con capacidad para grabación en Full HD, que permite un rendimiento óptimo en diversas condiciones. La cámara utilizada se muestra en la figura 12.

Figura 12

Cámara



Nota. Fuente: Elaboración propia

La instalación de la cámara en la intersección Hernán Malo y Panamericana se realizó con el único objetivo de captar el flujo vehicular en tiempo real, la cámara fue colocada sobre un trípode el cual fue cuidadosamente colocado en un punto estratégico, asegurando el ángulo de inclinación y una visión clara y amplia de los vehículos que transitan por la vía. Como se evidencia en la figura 13.

Figura 13

Instalación de cámara



Nota. Fuente: Elaboración propia

Por otra parte, el sistema de procesamiento está compuesto por una computadora de alto rendimiento equipada con una tarjeta gráfica potente para la ejecución eficiente de modelos de IA y para el almacenamiento de los datos recolectados, se incorpora un disco de estado sólido de 1TB, lo que permite el acceso rápido y seguro a la información sin comprometer la velocidad del sistema. Se evidencia en la figura 14.

Figura 14

Disco de estado sólido de 1TB



Nota. Fuente: Elaboración propia

3.5 Recopilación de Datos

La recolección de datos es una fase fundamental en el desarrollo de este estudio, ya que permite obtener la información necesaria para el análisis y validación de los resultados. Este capítulo describe las fuentes de información utilizadas, los métodos utilizados para la recopilación de datos y los criterios de selección de información.

Para garantizar la confiabilidad y validez de los datos recopilados, se han utilizado diversas fuentes de información, las cuales se pueden clasificar en:

- ❖ **Fuente manual:** Datos obtenidos directamente mediante observación en campo, con la recolección de video se analizó personalmente el flujo vehicular del video.
- ❖ **Fuente video:** Información recopilada del estudio del código establecido.

Las herramientas computacionales (Software) utilizadas en la investigación y las versiones específicas de cada software. La selección de estas herramientas se basó en su capacidad de procesamiento de vídeo, visión por ordenador y aprendizaje profundo, aspectos esenciales para el desarrollo del algoritmo de detección y clasificación de vehículos.

3.5.1. Software y versiones empleadas

Al momento de analizar el software de visión artificial, es fundamental considerar las características que garanticen el cumplimiento de los requerimientos del sistema. Es esencial que el programa incluya filtros para analizar aspectos como la iluminación y el color de las imágenes, lo que permite diferenciar aquellos en movimiento de los que permanecen estáticos. Por esta razón, se ha optado por utilizar Python con OpenCV, ya que es una herramienta de código abierto con amplia variedad de bibliotecas para el procesamiento de imágenes.

La versión utilizada de Python es 3.11.11 debido a sus mejoras en rendimiento y optimización de memoria (Guevara *et al.*, 2024), facilitando la implementación de algoritmos eficientes para el procesamiento de imágenes y videos en la presente investigación. El logo de Python se evidencia en la figura 15.

Figura 15

Logo de Python



Nota. Fuente: (python.org, 2016)

Como se mencionó anteriormente el script programado para la recolección de los datos será Python con OpenCV, en este caso la versión utilizada de OpenCV es 4.11.0 ya que proporciona mejoras en la detección de objetos, optimización de algoritmos y soporte para hardware especializado, lo que permite una mayor eficiencia en la identificación y clasificación de vehículos. (Arévalo *et al.*, 2004).

El entorno de desarrollo para este proyecto es “Anaconda. Navigator”, una plataforma que permite el desarrollo de las aplicaciones con Python relacionado al análisis, procesamiento y programación de grandes volúmenes de datos, el cual realiza la recolección de datos en una base de datos, la actualización de software de datos mediante Conda y la creación de trabajos

de aprendizaje profundo (Jarrin & Vinueza, 2020).Dentro de este entorno se implementa Jupyter, el cual es una herramienta que permite la ejecución de un código en un cuaderno, facilitando la visualización de los datos. El logo de anaconda se evidencia en la figura 16.

Figura 16

Logo de entorno Anaconda

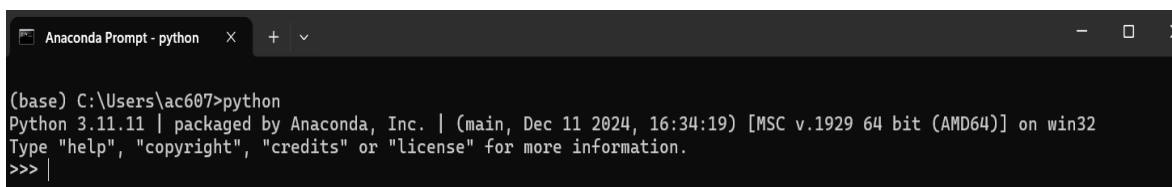


Nota. Fuente: (Anaconda.com, s.f.)

La interfaz gráfica de Python se muestra en la figura 17, el cual fue ejecutado dentro del entorno de Anaconda, una plataforma que permite la gestión de paquetes desarrollados en Python, el cual incluye bibliotecas y herramientas para el desarrollo de datos e inteligencia artificial .En este contexto se evidencia la versión activa de Python que fue instalada mediante el comando “Python” en el terminal de Anaconda, lo que garantiza que es compatible con una amplia variedad de bibliotecas para gestionar volúmenes de datos.

Figura 17

Interfaz gráfica de Python



```
Anaconda Prompt - python  x  +  v
(base) C:\Users\ac607>python
Python 3.11.11 | packaged by Anaconda, Inc. | (main, Dec 11 2024, 16:34:19) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> |
```

Nota. Fuente: Elaboración propia

3.5.2. Generación de la base de datos

La generación de bases de datos es fundamental para el entrenamiento y desarrollo del modelo de recuento y clasificación de vehículos. En este proyecto la base de datos se crea a partir de los videos captados en la carretera panamericana y Hernán Malo. Estos vídeos constituyen los principales insumos para entrenar y probar el modelo, asegurando que esté disponible una amplia variedad de escenarios de tráfico, condiciones de iluminación y tipos de vehículos.

Para construir la base de datos se utiliza OpenCV para la segmentación, sustracción y preprocesamiento de los vídeos, permitiendo extraer fotogramas clave en los que se identifican y etiquetan los vehículos. Posteriormente se aplican técnicas de anotación manual y automática para clasificar los vehículos según categorías como vehículos livianos, vehículos pesados y motos.

Además, se utilizan técnicas de mejora de imagen y reducción de ruido para optimizar la calidad de los datos antes de alimentar el modelo. Esta base de datos es esencial para entrenar el algoritmo de visión artificial, asegurando que pueda detectar y clasificar vehículos con precisión en diferentes condiciones de tráfico urbano. En la realización del algoritmo mediante visión artificial se importaron librerías utilizando el comando `import` seguido del nombre de la librería.

Las librerías empleadas indican lo siguiente:

- ❖ **CV2:** Pertenece a OpenCV y se utiliza para el procesamiento de imágenes y videos en Python.
- ❖ **Os:** Es un módulo de Python que permite interactuar con el sistema operativo para manipular archivos y directorios.

Posteriormente, se utiliza el video de entrada, con el único fin de extraer imágenes a partir de él, utilizando la función `“cv2.VideoCapture()”` que permite cargar el contenido del video también, se realizó una reducción al 50% de la resolución de los fotogramas extraídos utilizando la función `“cv2.resize()”` con el único fin de mejorar el procesamiento en cuanto al tiempo y carga. Es importante mencionar que para la detección de vehículos se utilizó el clasificador Haar Cascade, esta técnica de aprendizaje automático es conocido por su capacidad de detección eficiente, las características de Haar son usadas para detallar los bordes y cambios en el brillo de los objetos en movimiento. (Rivera & Zambrano, 2022). Este clasificador carga `“cars.xml”`, un archivo cargado con la función `“cv2.CascadeClassifier()”`, el cual es un modelo preentrenado para la detección de vehículos aplicada a la imagen en una escala de grises para mejorar la eficiencia.

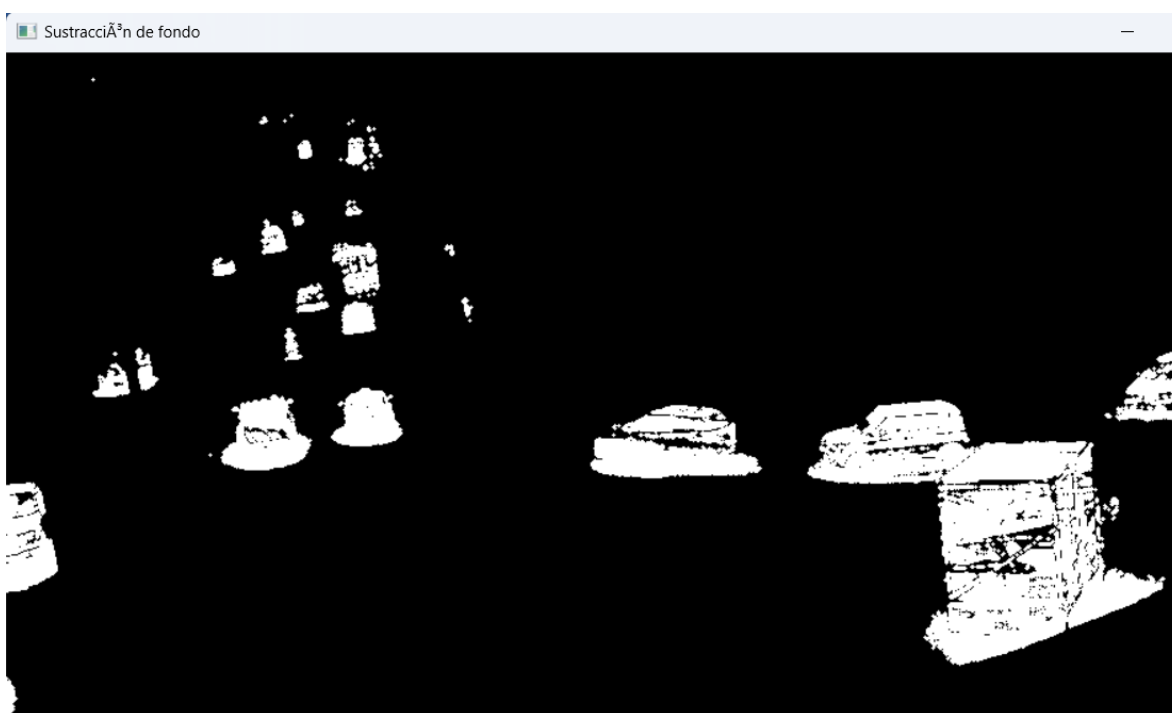
Además, para una mejor detección de los vehículos es importante combinar el clasificador haar con la utilización de la sustracción de fondo, debido a que permite identificar mejor los

videos con fondos dinámicos, la sustracción de fondo elimina el fondo estático, para ello se utilizó el algoritmo de Background Subtractor MOG2 de la librería de visión artificial OpenCV evidenciado en la figura 18, el cual funciona de la siguiente manera:

- ❖ MOG2: Este modelado funciona con una mezcla de modelos gaussianos para el representar el fondo del video, realizando de esta manera la detección de objetos en movimiento en un video.
- ❖ Mascara en primer plano: Se realiza el cálculo de la diferencia entre la imagen actual y el modelo de fondo y los objetos en movimiento se marcan en blanco en binaria, mientras que todo el fondo permanece de color negro.

Figura 18

Sustracción de fondo con MOG2



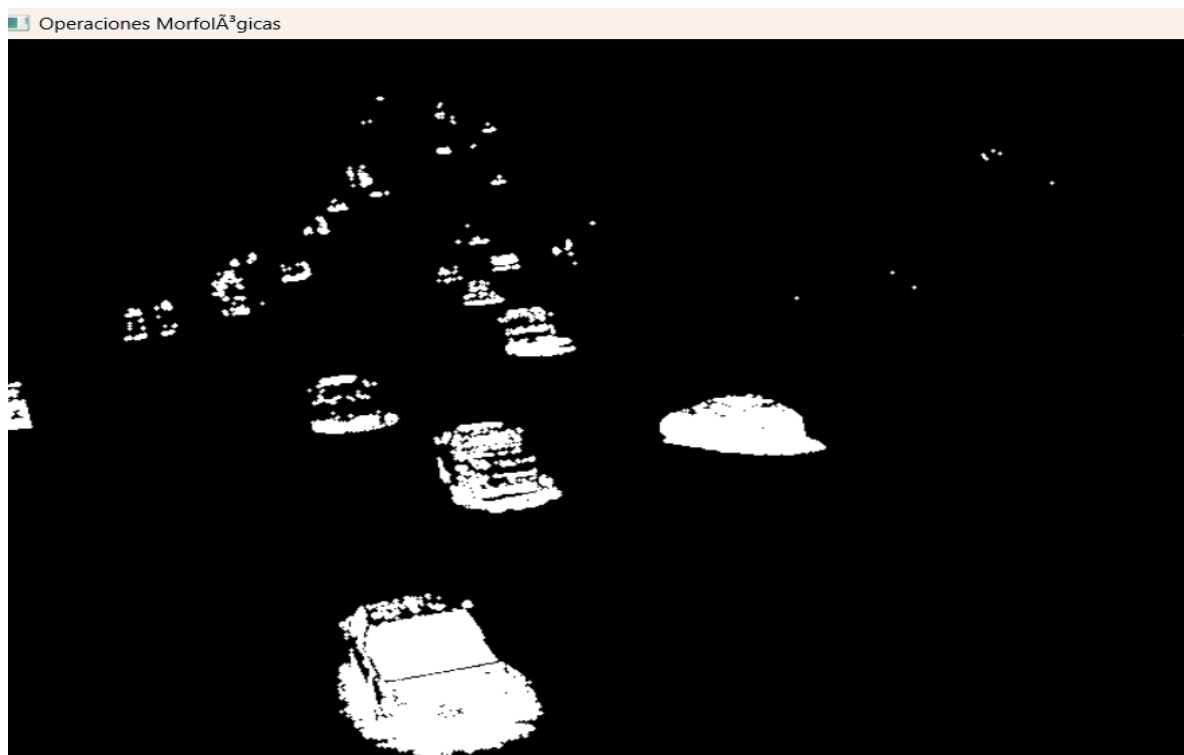
Nota. Fuente: Elaboración propia

Así mismo, se aplican filtros a las imágenes para eliminar el ruido, el cual se denominan operaciones morfológicas, que transforman a la imagen binaria para eliminar pequeños detalles que no son relevantes en la imagen. En este caso se utilizó una apertura “cv2.MORPH_OPEN”, la cual se muestra en la figura 19, que consiste en la aplicación de

una erosión seguida de una dilatación el cual mejora la suavización de los bordes y elimina puntos de ruido en las zonas blancas. Una vez realizadas estas operaciones, se aplica la función “cv2.findContours()”, evidenciada en la figura 20, la cual detecta los contornos en la imagen binaria y después de obtener dichos contornos cada uno es procesado para extraer su rectángulo delimitador utilizando la función cv2.boundingRect(), que se muestra en la figura 21, el cual devuelve coordenadas de dicho rectángulo y luego se verifica que el ancho y alto del contorno sean mayores a 100 píxeles para una mejor eficiencia de la detección de vehículos.

Figura 19

Operaciones morfológica-apertura



Nota. Fuente: Elaboración propia

Figura 20

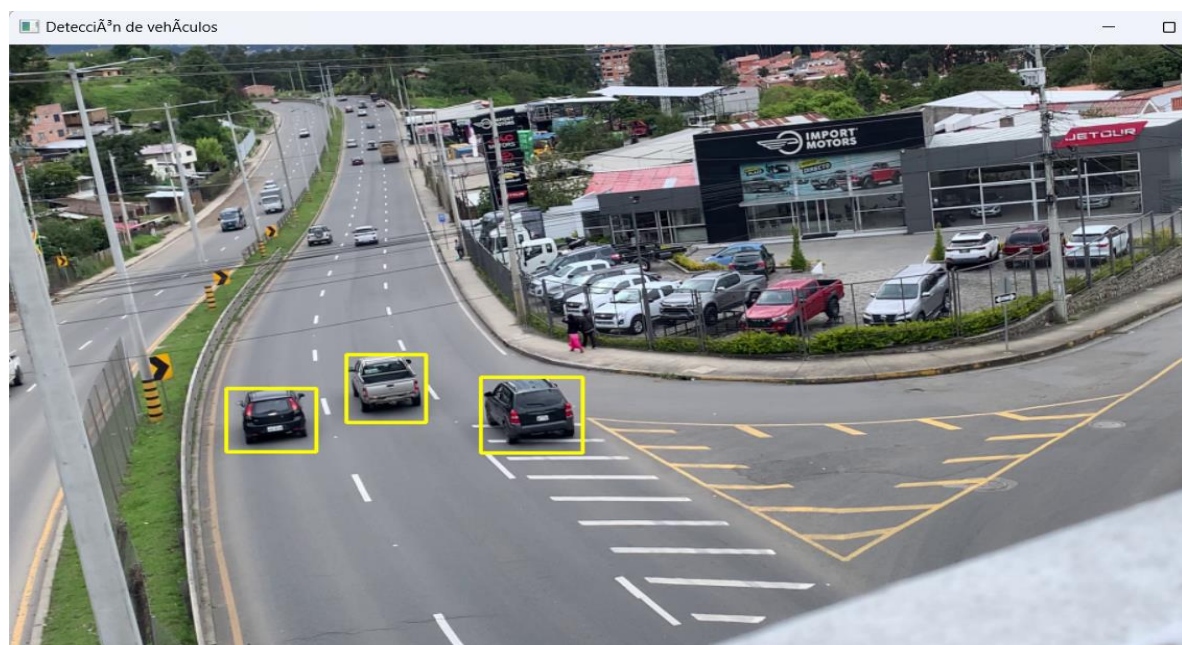
Contornos en vehículos



Nota. Fuente: Elaboración propia

Figura 21

Rectángulo delimitador para la detección



Nota. Fuente: Elaboración propia

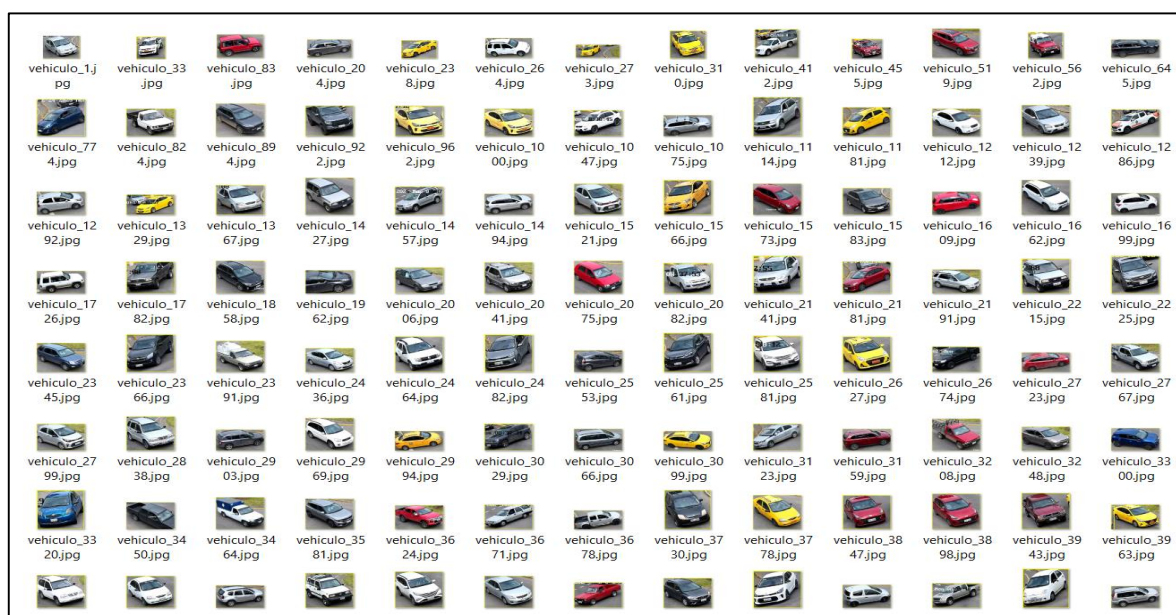
También, se crea una carpeta para los frames extraídos, es decir para almacenar las imágenes extraídas del video, mediante la función “os. makedirs ()”.

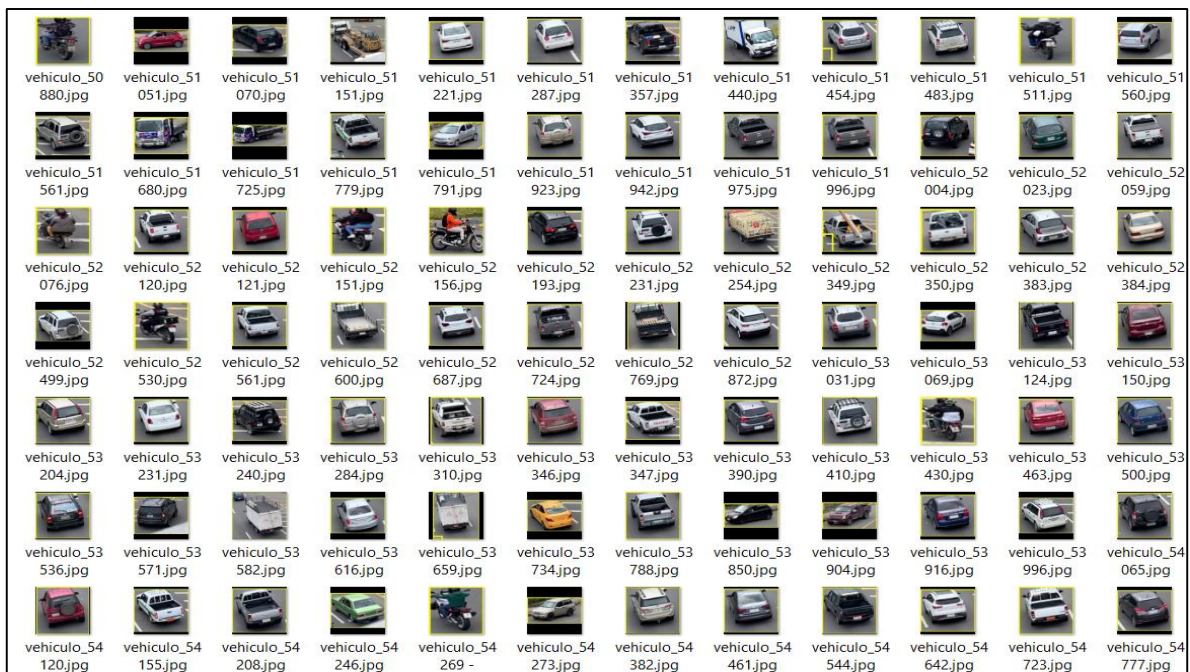
Así mismo, se configuro una tasa de muestreo de 1 frame cada 10 fotogramas mediante el parámetro “frame_skip” lo que permitió reducir la cantidad de imágenes extraídas y enfocarse en las más representativas, mejorando el procesamiento y almacenamiento de datos evitando redundancias en cuadros consecutivos, asegurando que las imágenes capturadas reflejen cambios significativos en la escena, como variaciones en las posiciones de los vehículos y las condiciones del tráfico. Además, esta tasa de muestreo facilita una anotación más eficiente en la base de datos sin comprometer la precisión del modelo de recuento y clasificación de vehículos, todos estos fotogramas se guardan y se muestran mediante la función “cv2.imshow” para mostrar la sustracción de fondo y fotogramas con las detecciones en tiempo real.

Finalmente, mediante el script programado en Python, se recolectaron 5809 imágenes o fotogramas evidenciados en la figura 22, los cuales son organizados y clasificados para crear una carpeta automáticamente con las diferentes categorías como vehículos livianos, vehículos pesados y motos, lo cual se evidencia en la figura 23.

Figura 22

Fotogramas recolectados

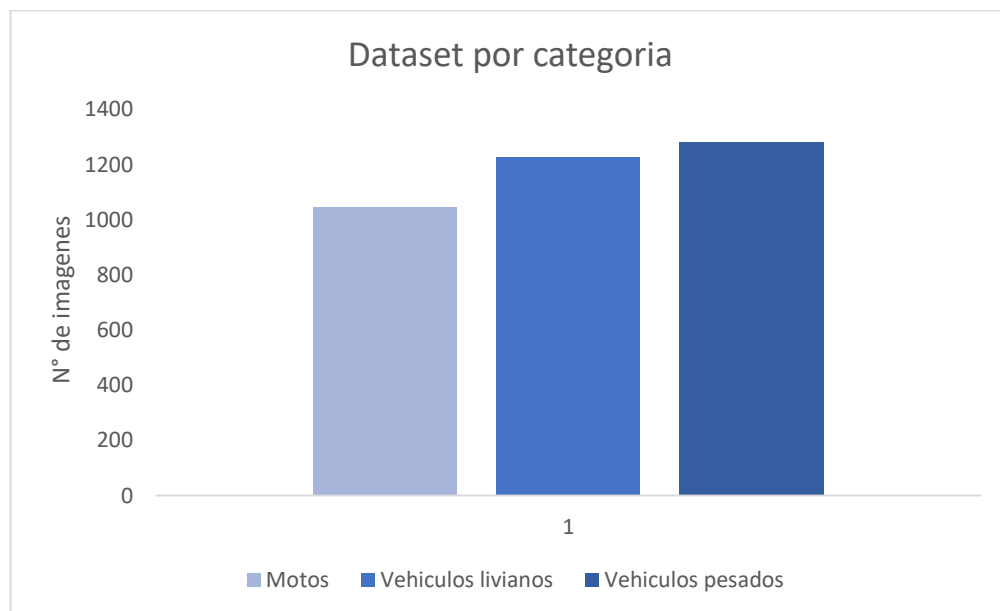




Nota. Fuente: Elaboración propia

Figura 23

Numero de imágenes por categoría



Nota. Fuente: Elaboración propia

3.6 Desarrollo de la red neuronal

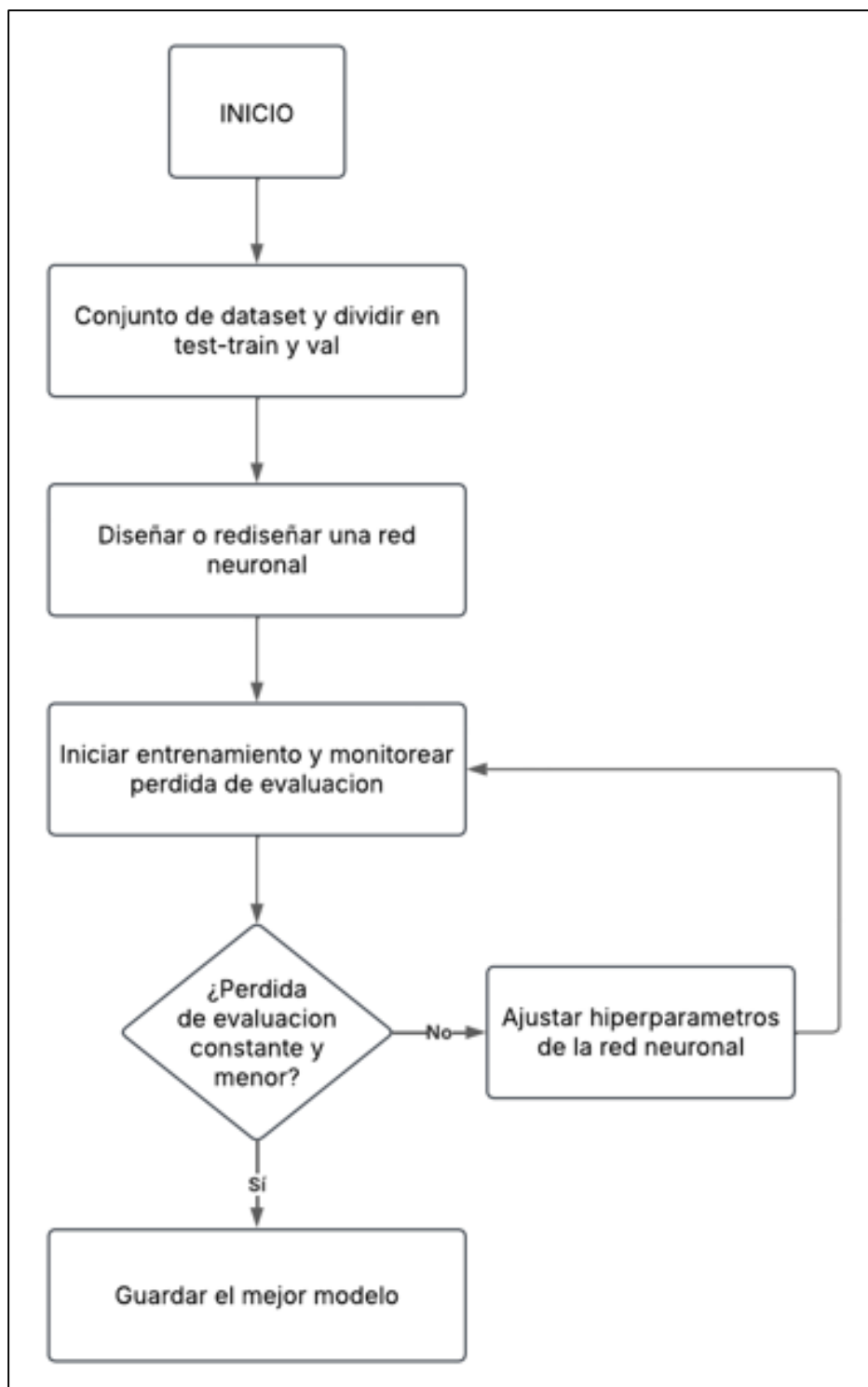
El desarrollo de la red neuronal ha revolucionado el campo de la inteligencia artificial permitiendo a las maquinas aprender patrones complejos de acuerdo a la base de datos, estas redes consisten en varias capas de neuronas artificiales que procesan información de manera jerárquica.

El modelo inicial para llevar a cabo este procedimiento fue AlexNet el cual revoluciono el aprendizaje profundo tras su victoria en el desafío ImageNet en 2012, siendo una red neuronal profunda compuesta por capas convolucionales y completamente conectadas. Su estructura se basa en cinco capas convolucionales, seguidas de capas de normalización, funciones de activación ReLU, capas de agrupamiento y tres capas conectadas, finalizando con una capa de salida para clasificar (Maeda, *et al.*, 2018). AlexNet se implementará en Tensor Flow, con el objetivo de clasificar vehículos en diferentes categorías a partir de imágenes extraídas de un video. Finalmente, para la implementación esta red es fundamental garantizar que las imágenes de entrada tengan un tamaño específico, esta red fue diseñada para procesar imágenes con una dimensión de 227 x 227 píxeles, lo que requiere un preprocesamiento adecuado de los datos antes de su uso en la red. Sin embargo, tras realizar pruebas iniciales, se identificaron limitaciones de AlexNet para la detección de vehículos en tiempo real, mostro dificultades en las ubicaciones de la imagen y entrenamiento del modelo.

Para superar estas limitaciones, se optó por la arquitectura YOLOv8n (You Only Look Once), diseñada para la detección de objetos en tiempo real de múltiples objetos en movimiento. Para llevar a cabo este procedimiento, se empleará Pytorch 2.6.0, una de las bibliotecas más utilizadas en la actualidad debido a que permite implementar modelos de inteligencia artificial, incorpora optimizaciones en el entrenamiento de redes neuronales y una mejor integración con GPU y TPU, así como una mayor compatibilidad con Python 3.11.11, el cual permite una ejecución más rápida y eficiente de los modelos de detección de vehículos. A continuación, se detalla en la figura 24 el diagrama de flujo de la arquitectura de la red neuronal.

Figura 24

Diagrama de flujo de la arquitectura de red neuronal



Nota. Fuente: Elaboración propia

3.6.1 Preparación del Dataset

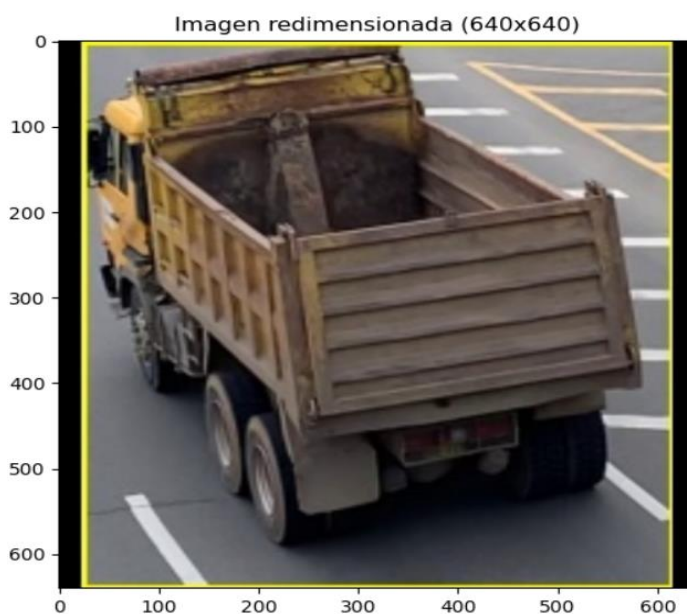
Las librerías empleadas indican lo siguiente:

- ❖ **CV2:** Pertenece a OpenCV y se utiliza para el procesamiento de imágenes y videos en Python.
- ❖ **Os:** Es un módulo de Python que permite interactuar con el sistema operativo para manipular archivos y directorios.

En este sentido todas las imágenes extraídas del video mediante OpenCV serán preprocesadas para asegurar la compatibilidad con la arquitectura de YOLO y para mantener la coherencia en la entrada de datos y mejorar la eficiencia del entrenamiento del modelo en Pytorch. A continuación, en la figura 25 se evidencia el preprocesamiento de las imágenes del dataset, es decir el redimensionamiento de los fotogramas a 640 x 640 pixeles debido a la arquitectura de YOLO. Además, en las imágenes preprocesadas se realiza el entrenamiento, validación y prueba del modelo, el dataset fue dividido en un 70% entrenamiento debido a que necesita de grandes volúmenes de datos para ajustar sus parámetros, 20% validación debido a que no se debe sobre ajustar el modelo y 10% de test.

Figura 25

Redimensión 640 x 640

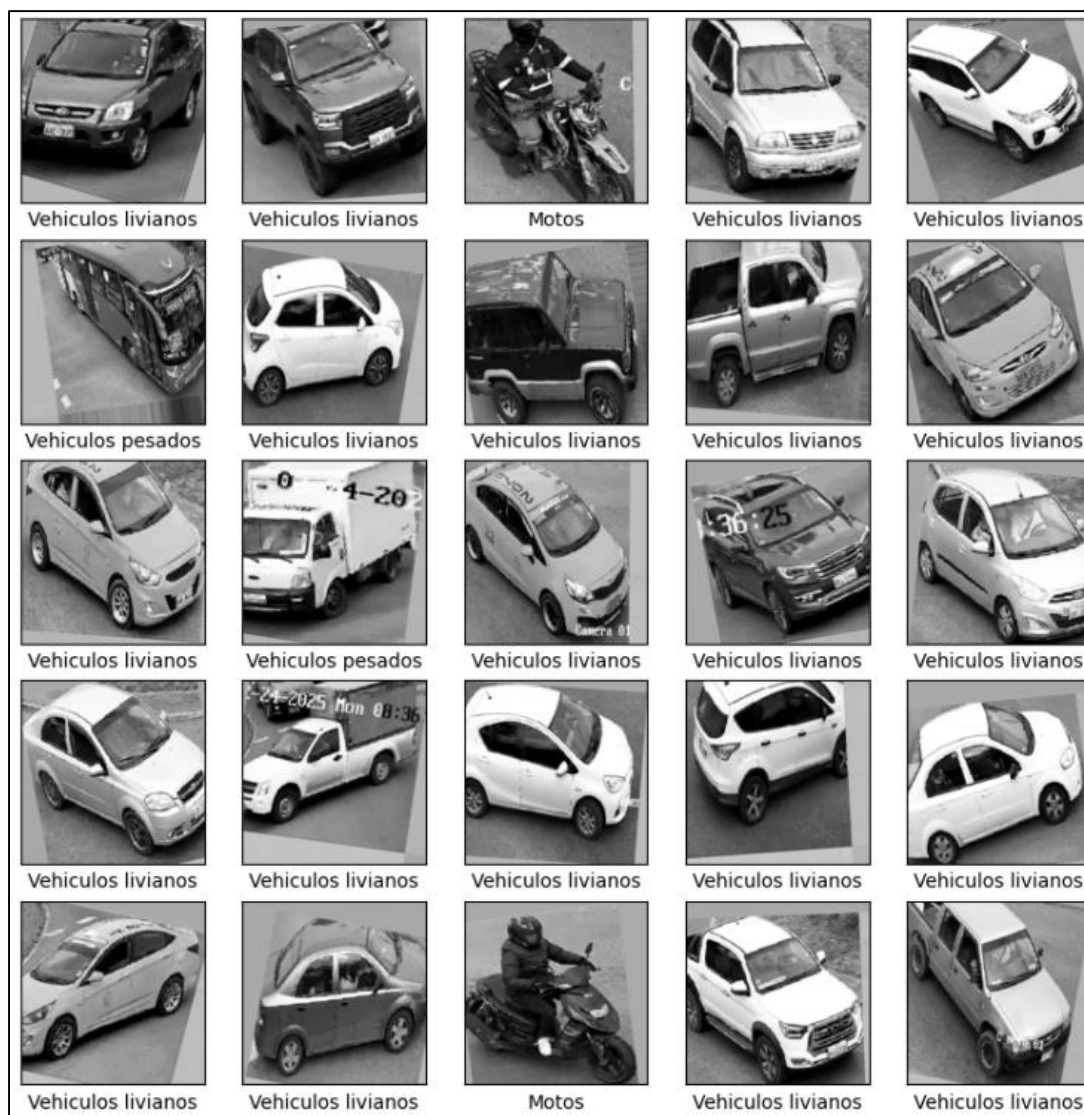


Nota. Fuente: Elaboración propia

A continuación, para validar el preprocesamiento se determinó la visualización de algunos fotogramas de la base de datos, normalizadas a una escala de grises y con su correcta etiqueta. En la figura 26 se muestra la etiqueta de las imágenes tomadas aleatoriamente del subconjunto de entrenamiento.

Figura 26

Etiqueta de imágenes



Nota. Fuente: Elaboración propia

Para el modelo YOLO es importante mantener el dataset en anotaciones, cada archivo de anotación debe estar en txt con el mismo nombre de la imagen correspondiente, este archivo

txt se encuentra en una carpeta denominada labels, la cual contiene una línea por cada objeto presente en la imagen para aprender a contar y clasificar vehículos.

En donde:

- ❖ Class_id = número que representa la clase de objeto, en este caso tendrá 0 para motos, 1 para vehículos livianos y 2 para vehículos pesados.
- ❖ X_center=coordenadas del centro del cuadro delimitador
- ❖ Y_center= coordenadas del centro del cuadro delimitador
- ❖ Width= dimensiones del cuadro delimitador
- ❖ Height= dimensiones del cuadro delimitador

3.6.2 Implementación de YOLOv8n

En el entrenamiento del modelo se utilizó el entorno Google Colab debido a que es de acceso gratuito a GPU T4 que es la unidad de procesamiento gráfico, ya que CPU es de procesamiento lento. Además, con GPU el procesamiento requiere de menos tiempo, menos almacenamiento y es compatible con una amplia variedad de librerías.

En el entorno de Google Colab se realiza un archivo dataset. yaml, el cual es un factor importante en el entrenamiento del modelo ya que está conformado con las carpetas images de dataset dividido en clases: motos, vehículos livianos y vehículos pesados, cada clase debe estar etiquetado adecuadamente para que se pueda acceder de manera correcta a la base de datos de entrenamiento, validación y prueba. El archivo dataset. yaml garantiza que no se cometa errores de lectura y está conformado de los siguientes parámetros:

- ❖ Path = indica la ruta en donde se encuentra la carpeta de dataset, por lo general suele empezar con /content/ debido a Google Colab.
- ❖ Train, test, val = carpetas que contiene las imágenes para el entrenamiento, validación y prueba.
- ❖ nc= indica la cantidad de clases
- ❖ names= lista los tipos de clases los cuales deben estar ordenados.

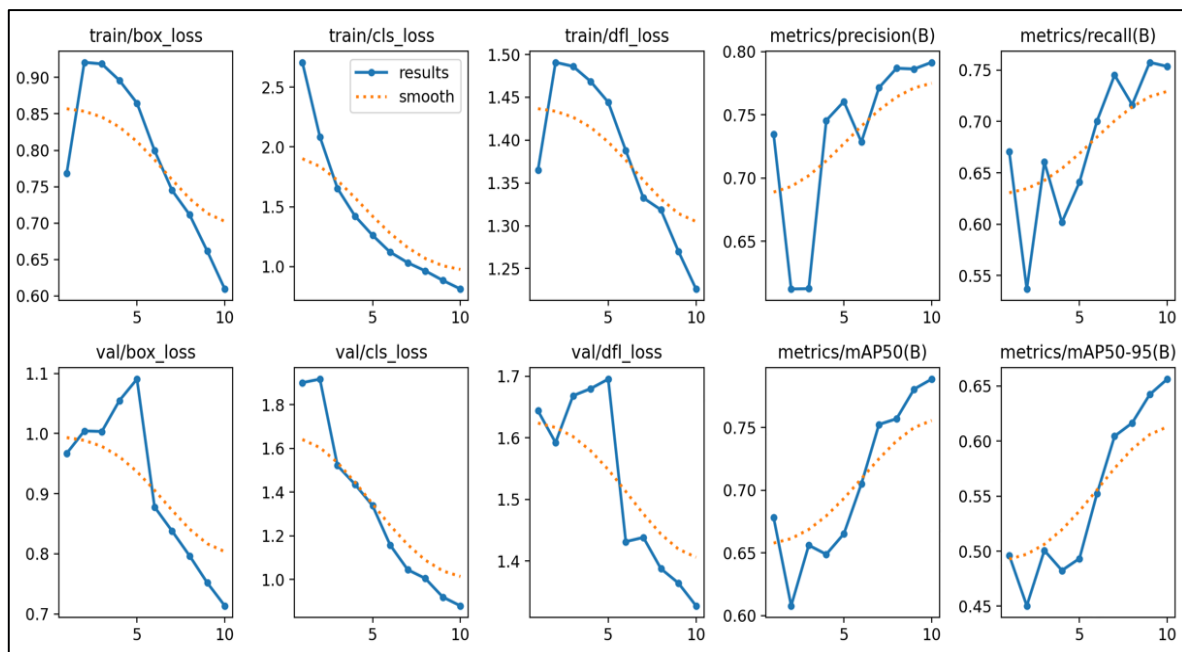
3.6.3 Entrenamiento y evaluación de la arquitectura

El modelo de YOLOv8n fue entrenado utilizando la base de datos en un 70% de entrenamiento, 20% validación y 10% de prueba, la configuración del modelo incluyó el

número de clases como motos, vehículos livianos y pesados, las épocas de entrenamiento la cual fuimos aumentando para obtener mayor eficiencia y precisión, el tamaño de lote, archivo yaml, utilización del GPU y algunos hiperparametros en el caso de ser necesarios. Además, se implementó una métrica de evaluación que permite evaluar el correcto desarrollo del modelo en cada iteración, lo que determina si el modelo Yolov8n está clasificando de manera correcta las diferentes clases. También es importante realizar las predicciones ya que permiten observar las predicciones correctas en función de las predicciones realizadas. Posteriormente se hace una evaluación de 4 iteraciones para un mejor entrenamiento del modelo probando de esta manera un ensayo-error, para analizar si se está sobre ajustando o no el modelo. En la figura 27 se muestra la iteración con 10 épocas, en la figura 28 con 20 épocas, figura 29 con 30 épocas y finalmente en la figura 30 se evidencia que luego de 50 épocas se obtiene una perdida tolerable el cual es adecuado para equilibrar la capacidad y tiempo del entrenamiento del modelo, ayudando a que se actualicen la base de datos en tiempo real.

Figura 27

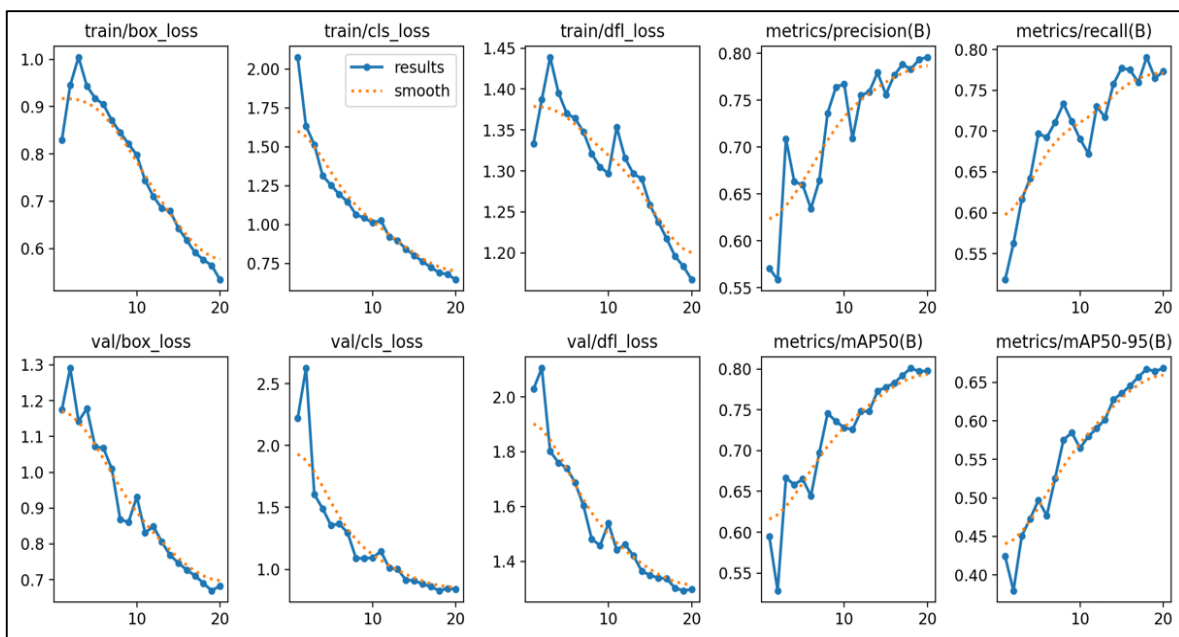
Iteración con 10 épocas de entrenamiento



Nota. Fuente: Elaboración propia

Figura 28

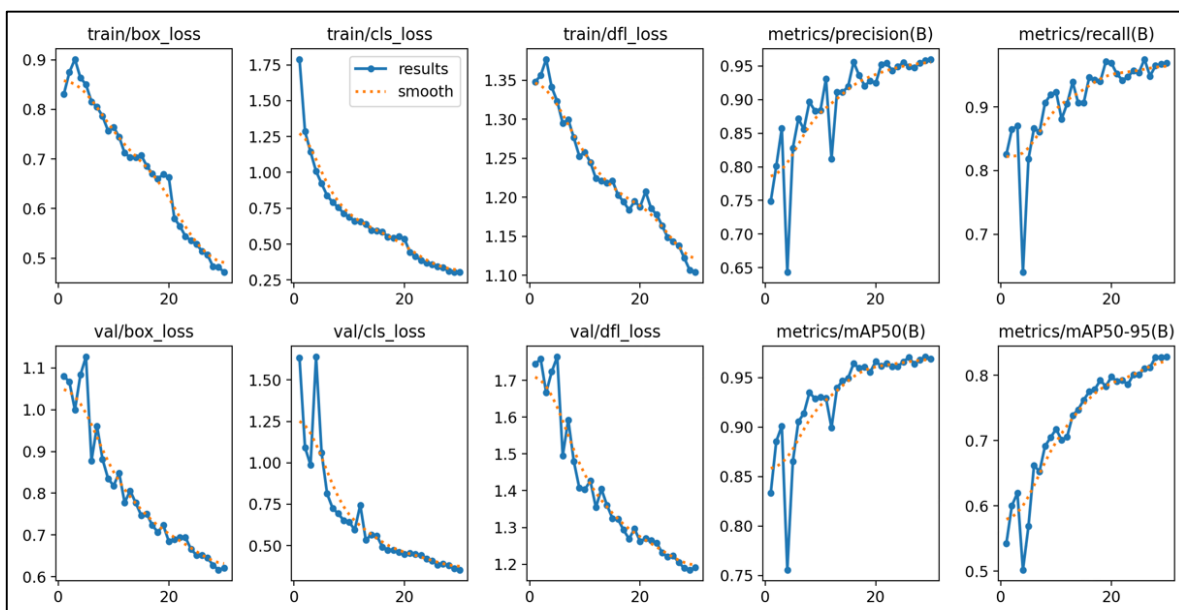
Iteración con 20 épocas de entrenamiento



Nota. Fuente: Elaboración propia

Figura 29

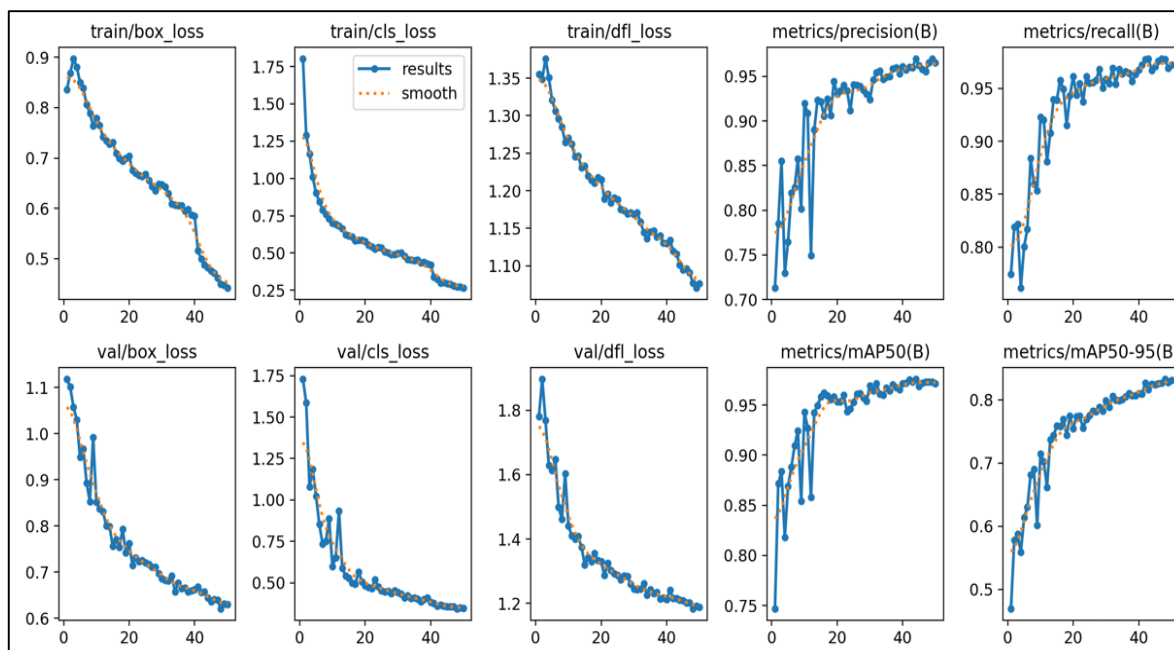
Iteración con 30 épocas de entrenamiento



Nota. Fuente: Elaboración propia

Figura 30

Iteración con 50 épocas de entrenamiento



Nota. Fuente: Elaboración propia

Una vez completado el entrenamiento del modelo, se procede a la evaluación utilizando los dataset de validación y prueba. Se utilizaron métricas de evaluación para el modelo evidenciados en la figura 31, la precisión mide cuantas detecciones fueron verdaderas, mPA@0,5 mide el promedio de precisión de todas las clases y mPA@0,95 evalúa el modelo con distintos umbrales desde 0,5 hasta 0,95. Durante el entrenamiento se alcanzó un mPA@0,5 superior al 90% para todas las clases de vehículos.

Figura 31

Métricas de evaluación

Class	Images	Instances	Box(P	R	mAP50	mAP50-95)
all	607	609	0.963	0.98	0.978	0.834
Motos	195	196	0.903	0.959	0.945	0.657
Vehiculos livianos	228	228	0.998	0.996	0.995	0.971
Vehiculos pesados	184	185	0.989	0.985	0.993	0.874

Precisión de validación: 0.9655

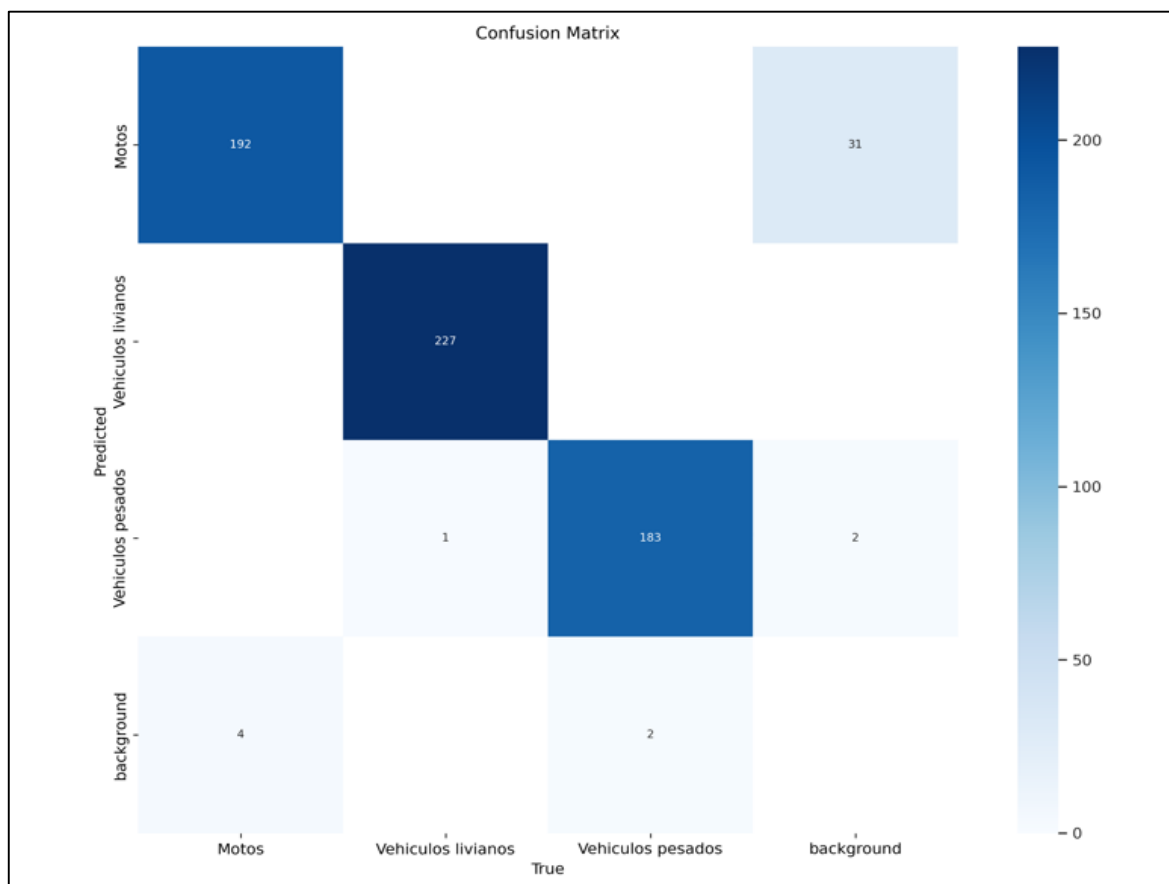
mAP@0.5 en validación: 0.97587

Nota. Fuente: Elaboración propia

Además, se utiliza una matriz de confusión para analizar la precisión y rendimiento del modelo en la clasificación de cada categoría. La Figura 32 y 33 muestran la matriz de confusión normal y la matriz de confusión normalizada obtenidas al comparar las predicciones del modelo con las etiquetas reales en el conjunto de validación. Es una tabla en donde se evidencia lo que el modelo predice en función de lo que es real, cada celda de la matriz representa el número de ejemplos clasificados verdadero o negativo en cada categoría. Es decir, la fila de la matriz representa la clase real mientras que la columna representa la clase predicha. A continuación, se observa que el modelo logra una alta precisión en la clasificación de vehículos, con un buen desempeño en todas las clases de vehículos, es decir, en las clases de vehículos livianos, vehículos pesados y motos logra obtener más del 97% de precisión.

Figura 32

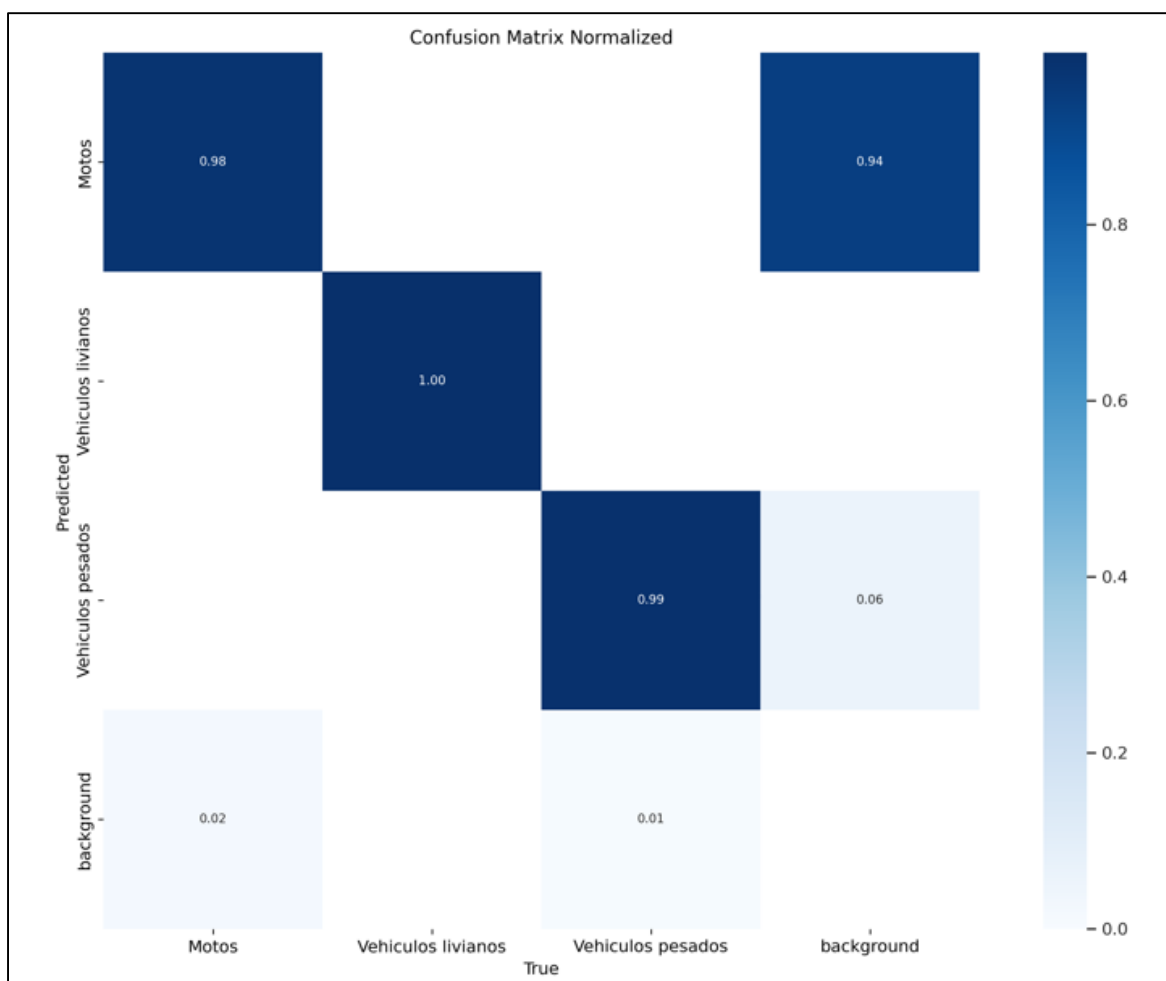
Matriz de confusión



Nota. Fuente: Elaboración propia

Figura 33

Matriz de confusión normalizada



Nota. Fuente: Elaboración propia

Para finalizar el análisis del modelo, se generó un informe de clasificación que proporciona las métricas como precisión, recuperación y puntuación F1 para cada clase de vehículo. Estos valores determinan el desempeño del modelo en la precisión y sensibilidad, lo que permite evaluar qué tan bien el modelo está diferenciando entre categorías. A continuación, se detalla cada termino descrito:

Precisión: Proporción de predicciones verdaderas para una clase sobre el total de predicciones realizadas. Se evidencia en la ecuación 5.

Ecuación 5: Precisión

$$Precision = \frac{TP}{TP + FP}$$

Donde:

TP=Verdaderos positivos

FP=Falsos positivos

Recall: Instancias de una clase que el modelo designo verdadero. Se evidencia en la ecuación 6.

Ecuación 6: Recall

$$Recall = \frac{TP}{TP + FN}$$

Donde:

TP=Verdaderos positivos

FN=Falsos negativos

F1-score: Toma en cuenta la precisión y la recuperación cuando existe un desbalance en las clases de vehículos. Se evidencia en la ecuación 7.

Ecuación 7: F1-score

$$F1 - score = 2 * \frac{Precision \times recall}{Precision + recall}$$

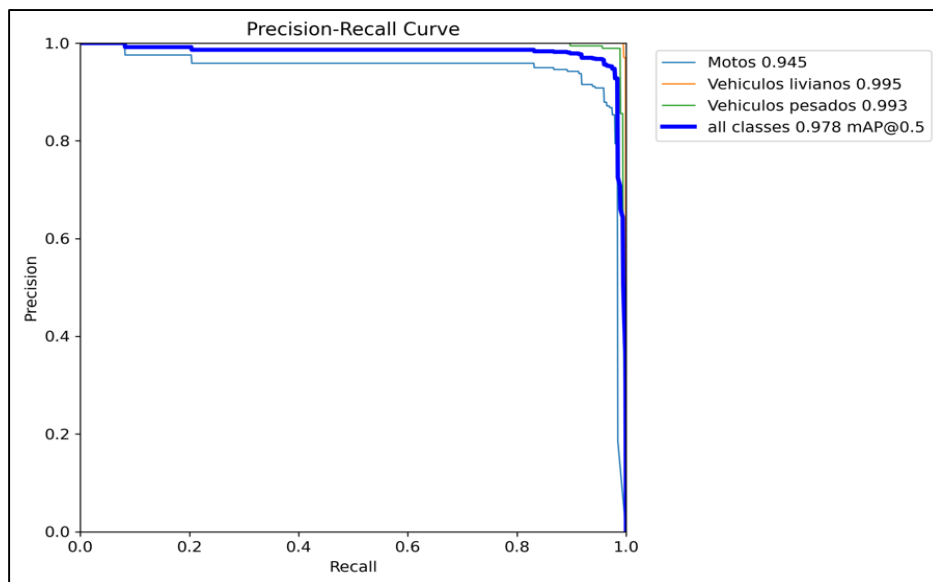
Support: Muestras verdaderas en cada clase.

Posteriormente en la figura 34 se evidencia la curva precisión vs Recall, la cual permite detallar y observar el equilibrio entre ambas métricas en distintos umbrales de decisión. Por otro lado, en la figura 35 se muestra la curva precisión vs confianza, que permite observar cómo varia la precisión del modelo, en la figura 36 se muestra la curva F1 la cual toma en cuenta la precisión y la recuperación cuando existe un desbalance en las clases de vehículos,

y finalmente en la figura 37, se presenta la curva recall vs confianza, la cual evalúa la sensibilidad del modelo en distintos niveles de confianza.

Figura 34

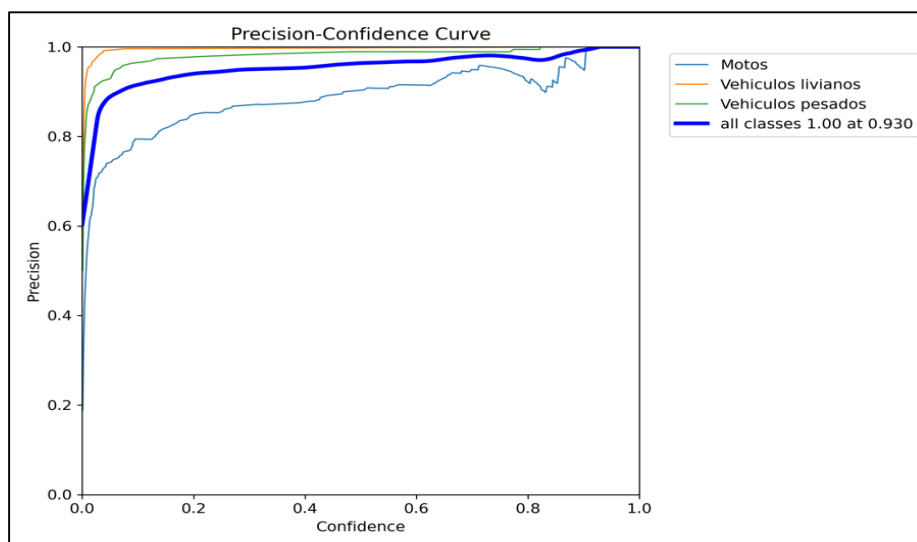
Curva precisión vs recall



Nota. Fuente: Elaboración propia

Figura 35

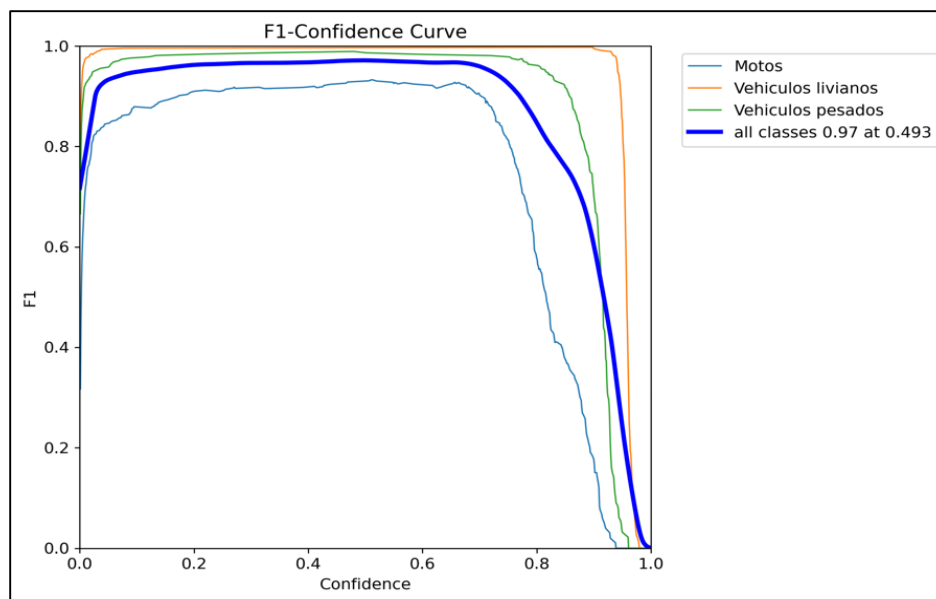
Curva precisión vs confianza



Nota. Fuente: Elaboración propia

Figura 36

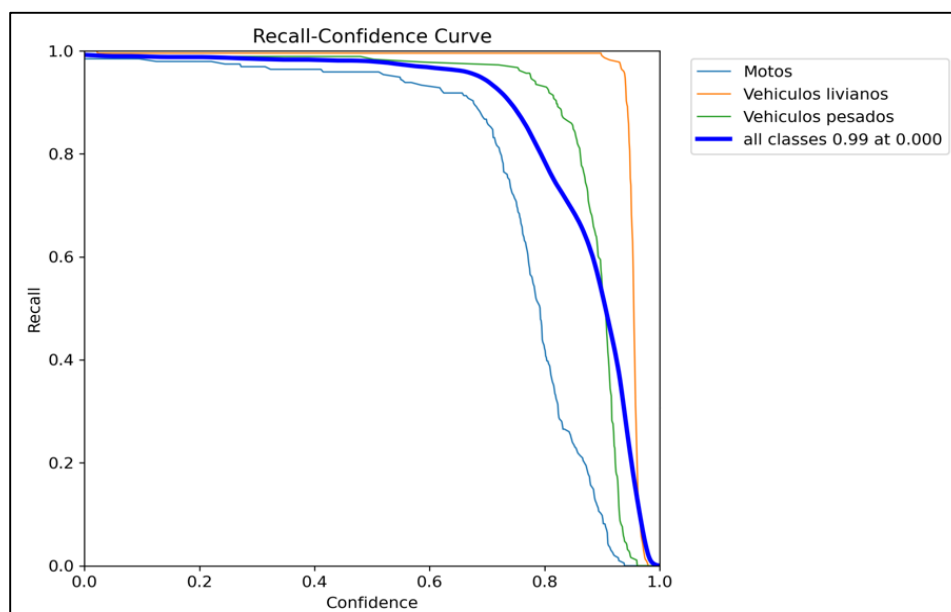
Curva F1 vs confianza



Nota. Fuente: Elaboración propia

Figura 37

Curva recall vs confianza

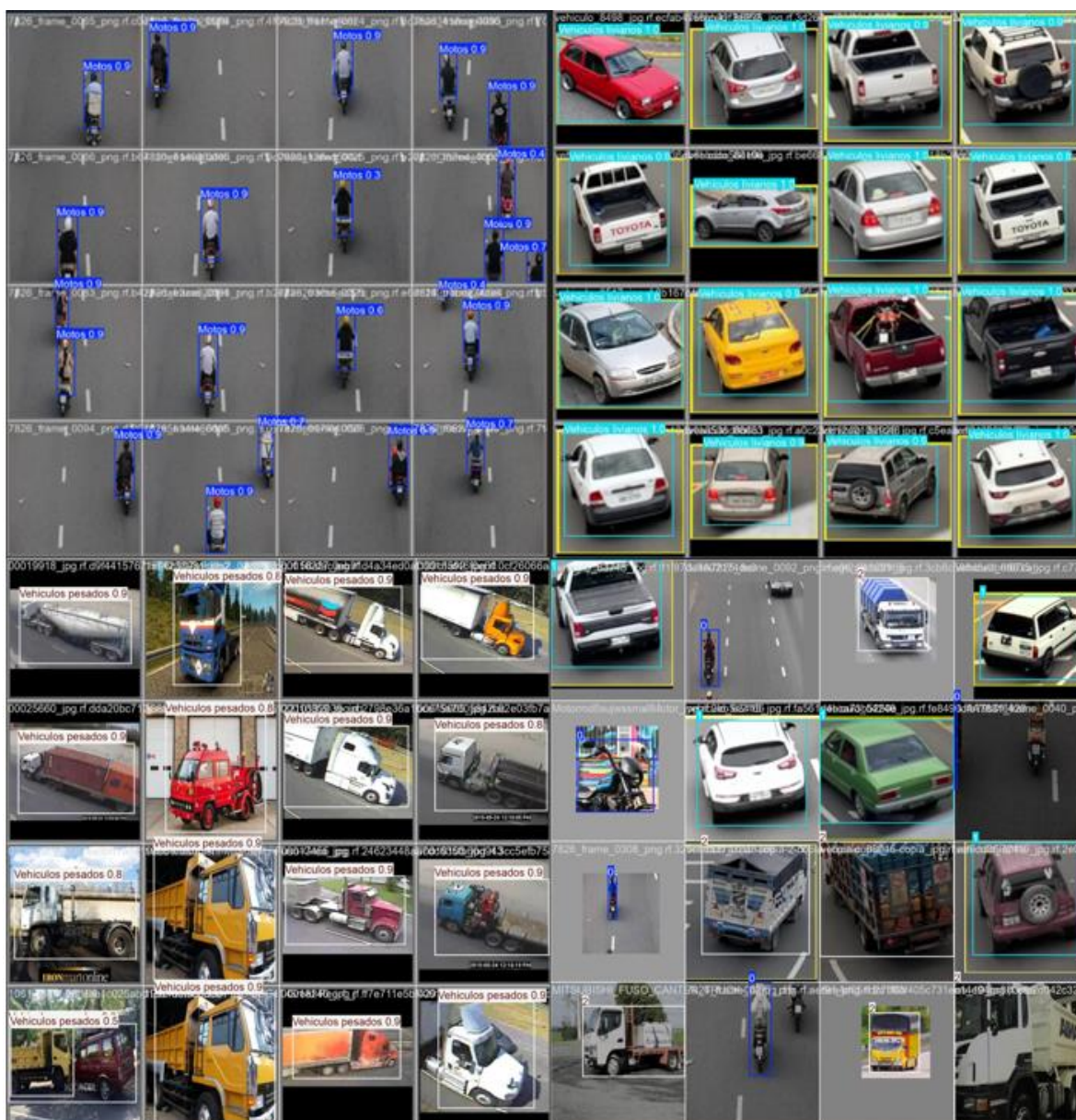


Nota. Fuente: Elaboración propia

En la visualización de predicciones se muestran los resultados en las imágenes del dataset, las visualizaciones muestran cuadros delimitadores denominados bounding boxes con la clase correspondiente, lo cual se evidencia en la figura 38.

Figura 38

Visualización de predicciones

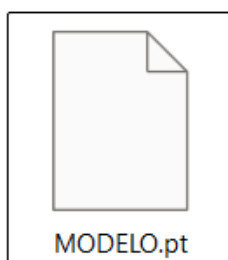


Nota. Fuente: Elaboración propia

El modelo final tiene un buen desempeño y rendimiento llegando a un 98% de precisión, el cual finalmente después de un adecuado entrenamiento y evaluación fue guardado en el formato.pt evidenciado en la figura 39, lo que va permitir que se pueda utilizar en futuras predicciones, facilitando su integración en un sistema de detección, conteo y clasificación vehicular.

Figura 39

Modelo guardado en formato pt



Nota. Fuente: Elaboración propia

3.7 Procesamiento

En la realización del algoritmo mediante inteligencia artificial se importaron librerías utilizando el comando `import` seguido del nombre de la librería como se detalla a continuación. Las librerías empleadas indican lo siguiente:

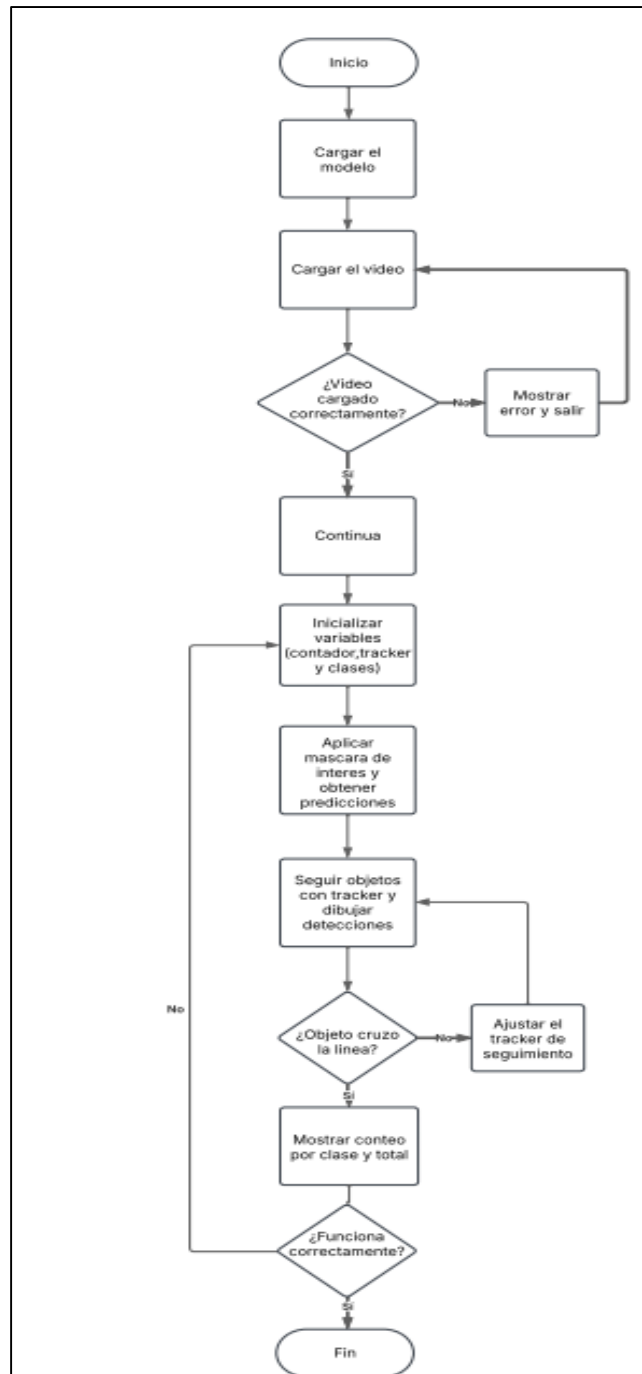
- ❖ **CV2:** Pertenece a OpenCV y se utiliza para el procesamiento de imágenes y videos en Python.
- ❖ **Numpy as np:** Realiza operaciones matemáticas y matrices.
- ❖ **Math:** Realiza operaciones matemáticas avanzadas
- ❖ **Ultralytics:** Sirve para cargar el modelo YOLOv8
- ❖ **Seguimiento de objetos:** Modulo para el seguimiento de vehículos
- ❖ **Torch:** Sirve para optimizar el rendimiento de YOLOv8

En esta fase se realiza el procesamiento del video, realización de una zona de mascara es decir para eliminar los ruidos que pueden causar los objetos en movimiento externos. Además, mediante YOLOv8n se realizará la detección de los vehículos y mediante un algoritmo de seguimiento se realizará el conteo vehicular. A continuación, se detalla en la

figura 40 el diagrama de flujo de la arquitectura de la detección, conteo y clasificación vehicular.

Figura 40

Diagrama de flujo de la arquitectura del procesamiento.



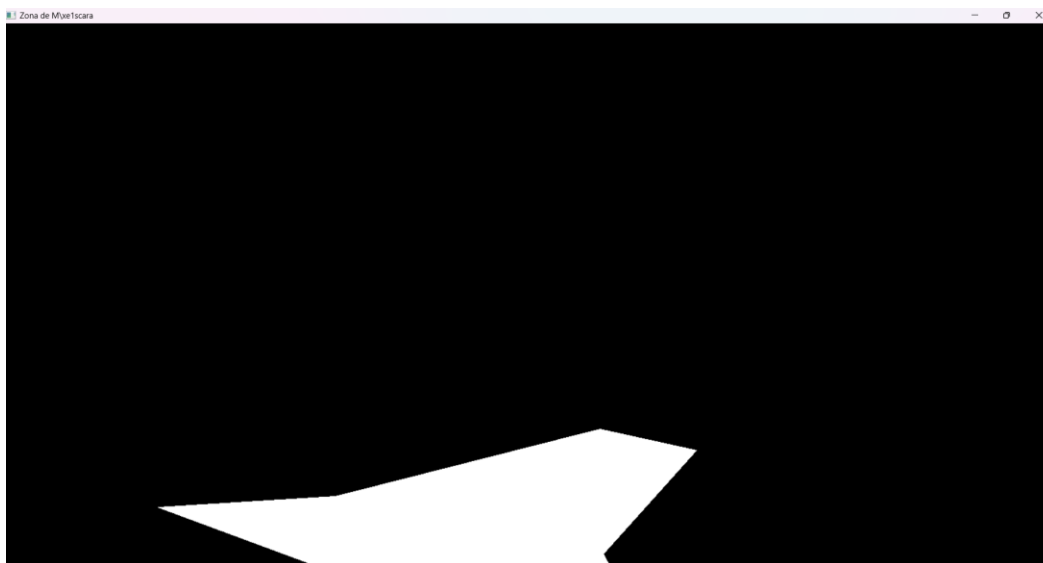
Nota. Fuente: Elaboración propia

3.7.1 Detección, conteo y clasificación vehicular

Una vez definidas las librerías a emplearse en el algoritmo mediante inteligencia artificial se realiza la carga del modelo entrenado y del video de la intersección Hernán Malo y Panamericana. Posteriormente se creará una zona de mascarar o de interés dentro del video que sirve para mejorar la eficiencia y eliminar el ruido, es decir el movimiento de árboles, personas o de cualquier otro objeto en movimiento que no sea de interés para el análisis del flujo vehicular. Esta zona de interés se realizó mediante un polígono con las siguientes coordenadas tanto en X como en Y [(869, 591), (1010, 622), (874, 773), (918, 854), (546,824), (224, 705), (483, 689)]. Además, se realizará la creación de una línea de conteo de color marrón la cual servirá de referencia para realizar el conteo vehicular, es decir cuando el vehículo cruce la línea se incrementa el conteo. En la creación de la línea de conteo se utilizó un color Brown definido por (19, 69, 139) con un grosor de 2 pixeles y coordenadas en X y Y, para definir el punto inicial y final de la línea, la coordenada inicial será (949,677) y la final será (732,629). En la figura 41 se puede evidenciar la zona de interés y en la figura 42 la línea de conteo.

Figura 41

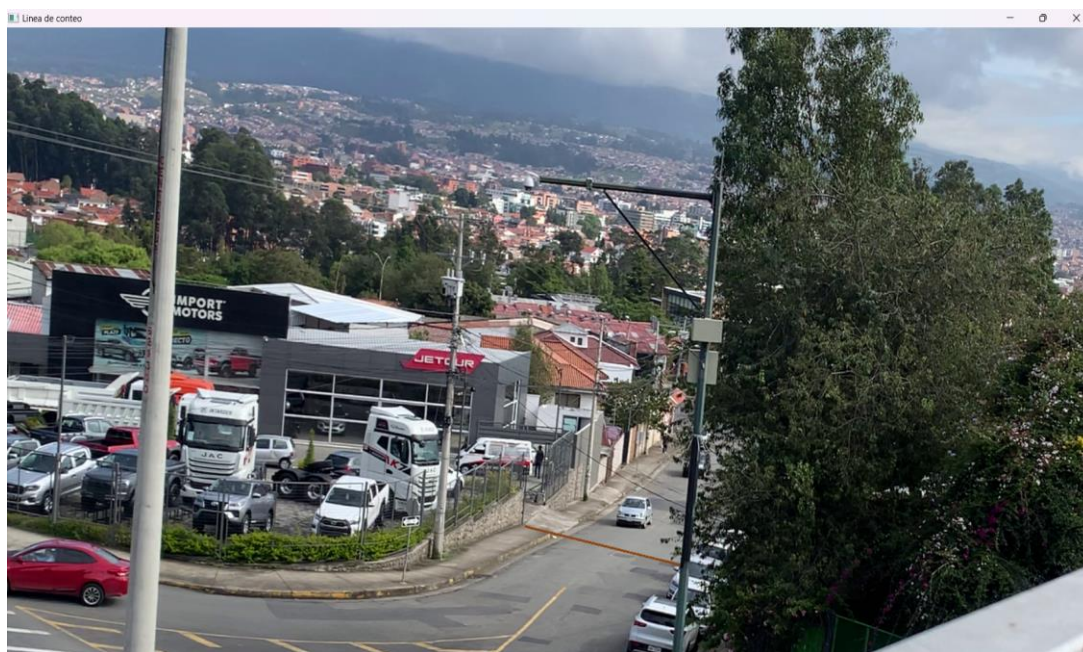
Zona de interés en el video (Hernán Malo y Panamericana)



Nota. Fuente: Elaboración propia

Figura 42

Línea de conteo en el video (Hernán Malo y Panamericana)



Nota. Fuente: Elaboración propia

Con el objetivo de evaluar el rendimiento y la precisión del modelo desde otra perspectiva, se replicó el procedimiento descrito en otro video grabado en la calle Panamericana en donde se puede observar a los vehículos de frente, buscando de esta manera analizar si la orientación de los vehículos influye en la precisión del modelo de conteo y clasificación vehicular. Para este segundo caso de análisis, se creó también una zona de mascara, con las coordenadas [(420, 411), (573, 367), (713, 404), (881, 492), (1163, 859), (300, 879), (477, 593), (487, 525), (475, 457), (453, 428)] que delimitan el área de interés, y se creó una nueva línea de conteo ajustada a la orientación frontal con las siguientes coordenadas [(1008, 661), (419, 693)] de color Brown definido por (19, 69, 139) con un grosor de 2 pixeles. Finalmente, en la figura 43 se evidencia la zona de interés utilizada para el análisis de la calle Panamericana y en la figura 44 se evidencia la línea de conteo.

Figura 43

Zona de interés en el video (Panamericana)



Nota. Fuente: Elaboración propia

Figura 44

Línea de conteo en el video (Panamericana)



Nota. Fuente: Elaboración propia

El procesamiento de cada fotograma se lo realiza tomando las dimensiones reales del video para que se adapten a su tamaño real. Además, mediante la función “model. predict ()” se identifican únicamente los objetos en movimiento dentro de la zona de interés. Es importante mencionar que imágenes debe ser de 640 pixeles ya que es el tamaño requerido para YOLOv8n y la confianza deberá ser entre un 0,25 lo que significa que solamente se tomaran en cuenta objetos en movimiento con una mayor probabilidad a este valor.

Dado que en la detección de vehículos el modelo YOLOv8n detecta millones de objetos en movimiento es importante realizar un filtrado de vehículos. Además, para impedir que se cuente muchas veces el mismo vehículo, se realiza un registro de las posiciones previas a los objetos detectados y se verifica si ha cruzado o no la línea de conteo.

Además, para facilitar la visualización de resultados, el algoritmo realiza una visualización de los vehículos detectados en tiempo real y el conteo total por cada clase de vehículo. Se dibujan rectángulos delimitadores alrededor de los vehículos en color naranja con el siguiente código de color (0, 165, 255), en el texto del conteo total de vehículos se muestra en la pantalla en color verde (0, 255,0) y en el texto de vehículos livianos, vehículos pesados, motos se muestra en la pantalla en color blanco (255, 255, 255).

Finalmente, para complementar el análisis del flujo vehicular observado en las grabaciones simultáneas en los dos horarios, se planteó un método de balance de flujos vehiculares que permiten identificar el flujo de vehículos que, por condiciones de la intersección, no es directamente visible en las grabaciones de video, para ello se definieron cuatro flujos vehiculares principales.

Q1: Flujo registrado en la autopista Panamericana.

Q2: Flujo de vehículos descendiendo desde la Panamericana hacia la Hernán Malo.

Q3: Flujo de vehículos ascendiendo desde la Hernán Malo hacia la Panamericana.

Q4: Flujo que continua su trayecto por la autopista Panamericana.

La estimación de este flujo vehicular establece el número de vehículos que ingresan y salen de una sección, los cuales deben mantenerse en equilibrio. El flujo indirecto denominado Q4, se calculó con la siguiente ecuación 8.

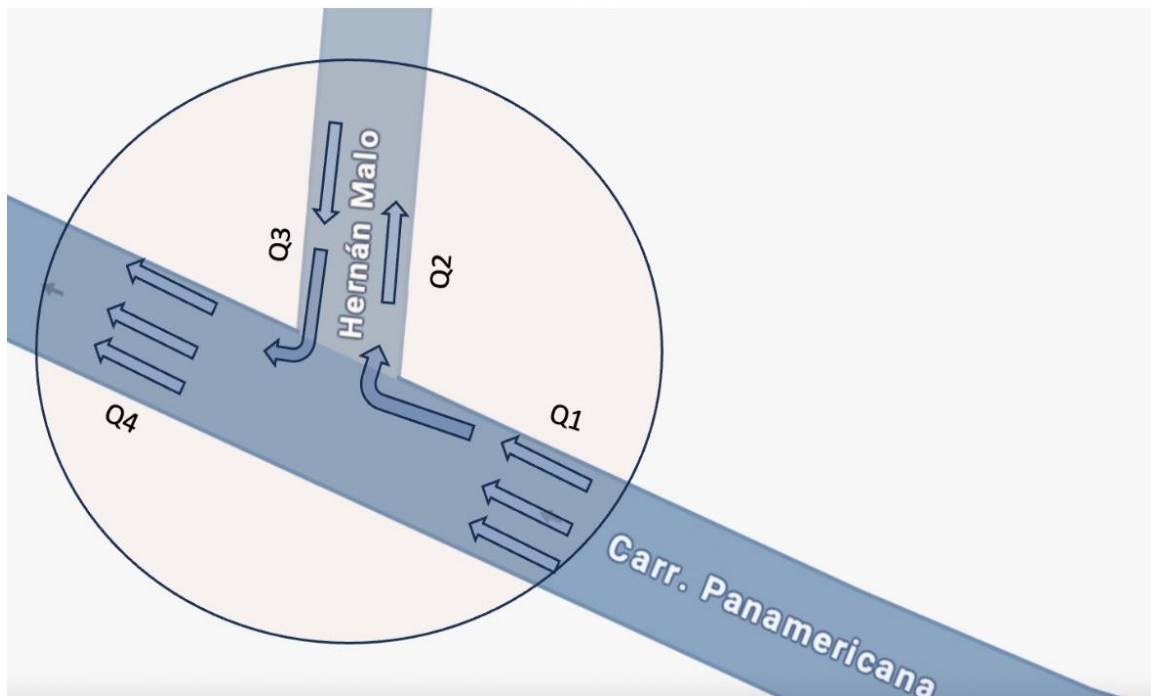
Ecuación 8: Flujo vehicular

$$Q4 = Q1 + Q3 - Q2$$

Al sumar el flujo que se mantiene en la Panamericana (Q1) los vehículos que ascienden desde la Hernán Malo (Q3), y restar aquellos que se desvían hacia la Hernán Malo (Q2), se obtiene el volumen neto de vehículos que permanecen en el sistema principal que sería la autopista Panamericana. Toda esta estimación permite completar el análisis de todos los movimientos vehiculares en la intersección, corregir posibles pérdidas de datos por limitaciones de visibilidad y obtener una estimación real del comportamiento del flujo vehicular. Se evidencia en la figura 45 la elaboración de un croquis esquemático de la intersección bajo estudio. En este croquis se representa gráficamente los flujos vehiculares Q1, Q2, Q3 y el flujo calculado Q4, facilitando la comprensión del movimiento de los vehículos entre la calle Panamericana y la avenida Hernán Malo.

Figura 45

Esquema de flujos en la intersección



Nota. Fuente: Elaboración propia

CAPITULO IV

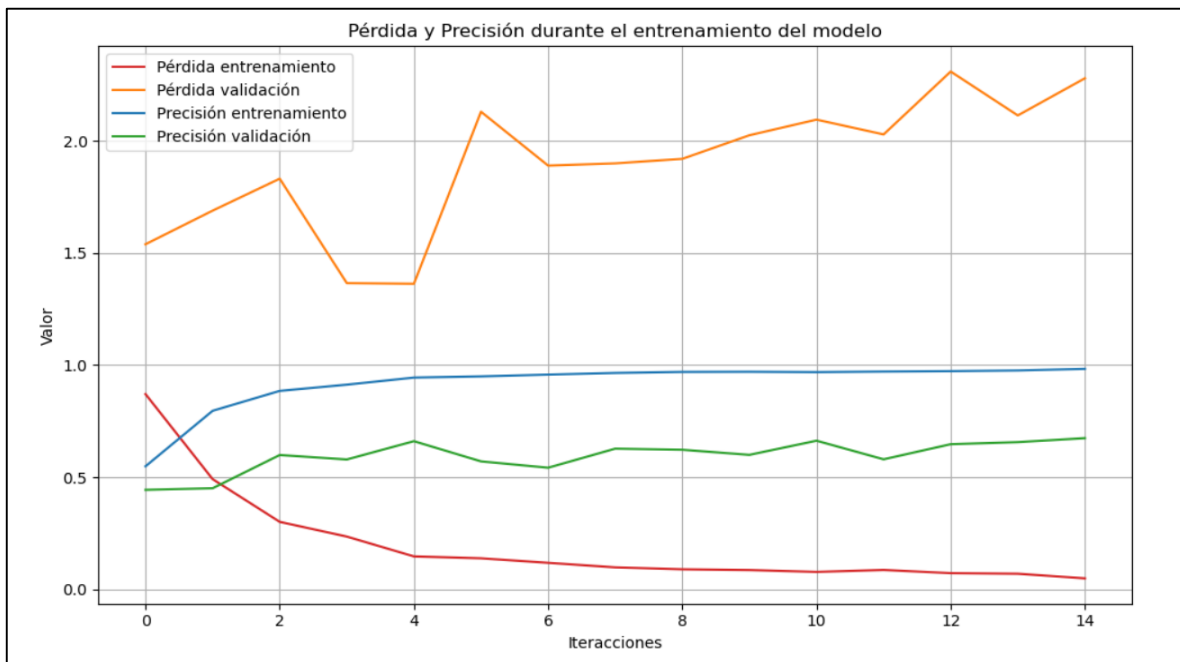
Capítulo 4: Resultados

4.1 Comparación de AlexNet y Yolov8n

Durante la fase inicial de pruebas, se entrenó un modelo basado en la arquitectura AlexNet con el objetivo de realizar el conteo y clasificación vehicular, el entrenamiento del modelo se realizó utilizando una base de datos dividido en entrenamiento, validación y prueba. Sin embargo, al realizar ajustes de hiperparámetros los resultados obtenidos fueron ineficientes, la precisión máxima alcanzada fue del 60% y diversos aspectos importantes que resultaron ineficientes. A continuación, se detallan los resultados obtenidos en el entrenamiento de los modelos. En la figura 46 se muestra la gráfica de pérdida y precisión del modelo AlexNet, en la figura 47 del modelo Yolov8n y en la tabla 1 se muestra los resultados comparativos de los dos modelos.

Figura 46

Resultados de grafica de precisión en el entrenamiento del modelo AlexNet



Nota. Fuente: Elaboración propia

Figura 47

Resultados de grafica de precisión en el entrenamiento del modelo Yolov8n



Nota. Fuente: Elaboración propia

Tabla 1

Comparación de la red neuronal AlexNet y YOLOv8n

Métrica de evaluación	AlexNet	YOLOv8n
Precisión de evaluación	60%	97%
Tiempo de entrenamiento	2 h	20m
Velocidad de detección	5fps	30fps
Recuperación	34%	96%
Falsos positivos	10%	2%

Nota: Datos obtenidos en el proyecto facilitan la comparación entre las diferentes redes neuronales.

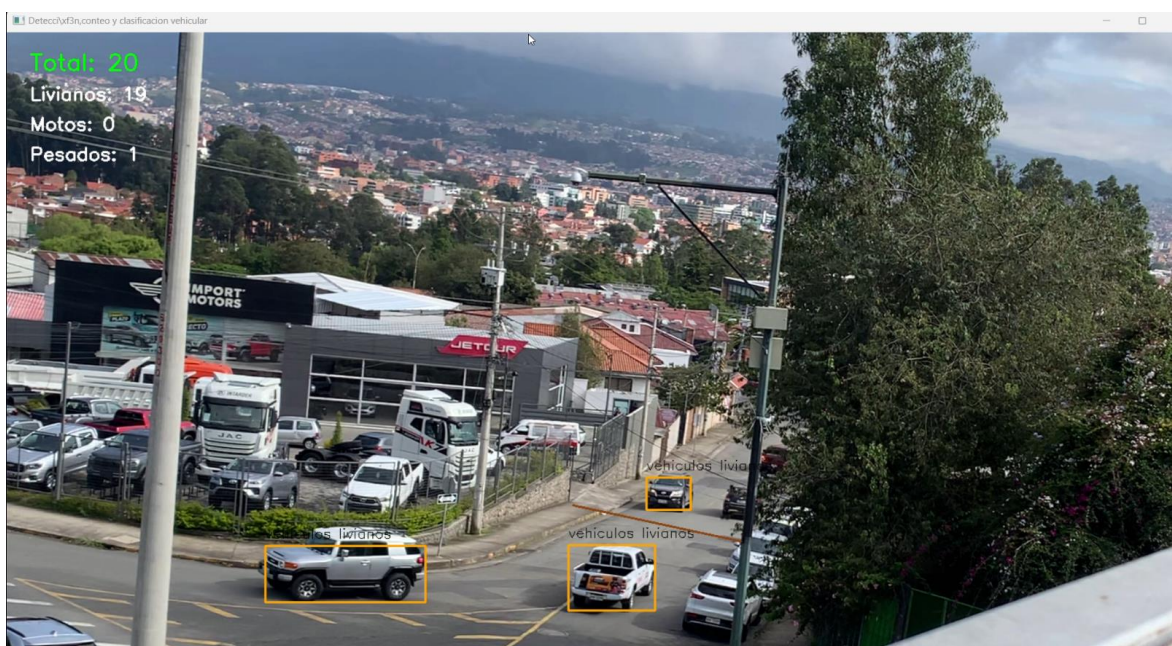
Los resultados indican que YOLOv8n presente una mayor precisión y eficiencia que AlexNet, ya que esta al ser una red neuronal orientada a la clasificación de imágenes, presente limitaciones al trabajar con secuencia de videos en movimiento, mientras que YOLOv8n logro detectar los objetos en movimiento en tiempo real con una precisión mAP del 97%, lo

cual resulta ser la arquitectura de red neuronal más factible para la detección en tiempo real del tráfico vehicular.

4.2 Resultados de clasificación de vehículos

Mediante un algoritmo de inteligencia artificial para la detección, conteo y clasificación vehicular en la intersección Hernán Malo y Panamericana se obtuvieron resultados con una precisión alta, el modelo fue evaluado en grabaciones simultaneas , con el objetivo de registrar de manera precisa el número de vehículos que descienden desde la Panamericana hacia la Hernán Malo y aquellos que ascienden desde la Hernán Malo hacia la Panamericana, evitando de esta manera perdidas de flujo vehicular en el conteo. Las grabaciones se realizaron en dos horarios representativos de alta demanda vehicular: 07:30 a 8:00 horas en la mañana y de 17:30 a 18:00 horas en la tarde, estos horarios fueron seleccionados para capturar los patrones de tráfico en hora pico. A continuación, en la figura 48 y 49 se muestra la detección y clasificación en la intersección Hernán Malo y Panamericana con sus respectivas etiquetas. Además, en la figura 50 y 51 se muestra la detección y clasificación en la calle Panamericana.

Figura 48 Detección con etiquetas de clasificación vehicular (Hernán Malo y Panamericana)



Nota. Fuente: Elaboración propia

Figura 49 Detección con etiquetas de clasificación vehicular (Hernán Malo y Panamericana)



Nota. Fuente: Elaboración propia

Figura 50 Detección con etiquetas de clasificación vehicular (Panamericana)



Nota. Fuente: Elaboración propia

Figura 51

Detección con etiquetas de clasificación vehicular (Panamericana)



Nota. Fuente: Elaboración propia

4.2.1 Conteo de vehículos en la intersección seleccionada

El conteo de vehículos fue realizado en la intersección seleccionada mediante un seguimiento individual de vehículos utilizando una línea de conteo virtual, en donde cada vehículo fue seguido a través de la trayectoria que realizó hasta cruzar la línea de conteo, evitando de esta manera que se cuenten muchas veces el mismo vehículo, lo cual puede afectar significativamente a la precisión del conteo vehicular ya que se puede presentar errores de duplicidad. El control de duplicidad es importante para garantizar la calidad de los datos, debido a que un sobre conteo puede alterar el análisis del flujo vehicular, afectando los balances del tráfico. A continuación, en la tabla 2 se presenta la cantidad de vehículos detectados en un lapso de tiempo de 30 min de 07:30 a 8:00 am en la intersección Hernán Malo y Panamericana tanto en sentido de ascenso como descenso. Así mismo en la tabla 3 se presenta la cantidad de vehículos detectados en un lapso de tiempo de 30 min de 17:30 a 18:00 pm.

Tabla 2

Conteo vehicular por clase en la Intersección Hernán Malo y Panamericana horario de 07:30 a 8:00 am.

Clase de vehículos	Conteo IA (Descenso Panamericana-Hernán Malo)	Conteo IA (Ascenso Hernán Malo- Panamericana)
Vehículos pesados	5	1
Vehículos livianos	159	83
Motos	1	1
Total	165	85

Nota: Datos obtenidos del conteo vehicular en las tres tipologías consideradas (livianos, pesados y motocicletas), incluyendo el total acumulado.

Tabla 3

Conteo vehicular por clase en la Intersección Hernán Malo y Panamericana horario de 17:30 a 18:00 pm.

Clase de vehículos	Conteo IA (Descenso Panamericana-Hernán Malo)	Conteo IA (Ascenso Hernán Malo-Panamericana)
Vehículos pesados	1	0
Vehículos livianos	45	38
Motos	3	1
Total	49	39

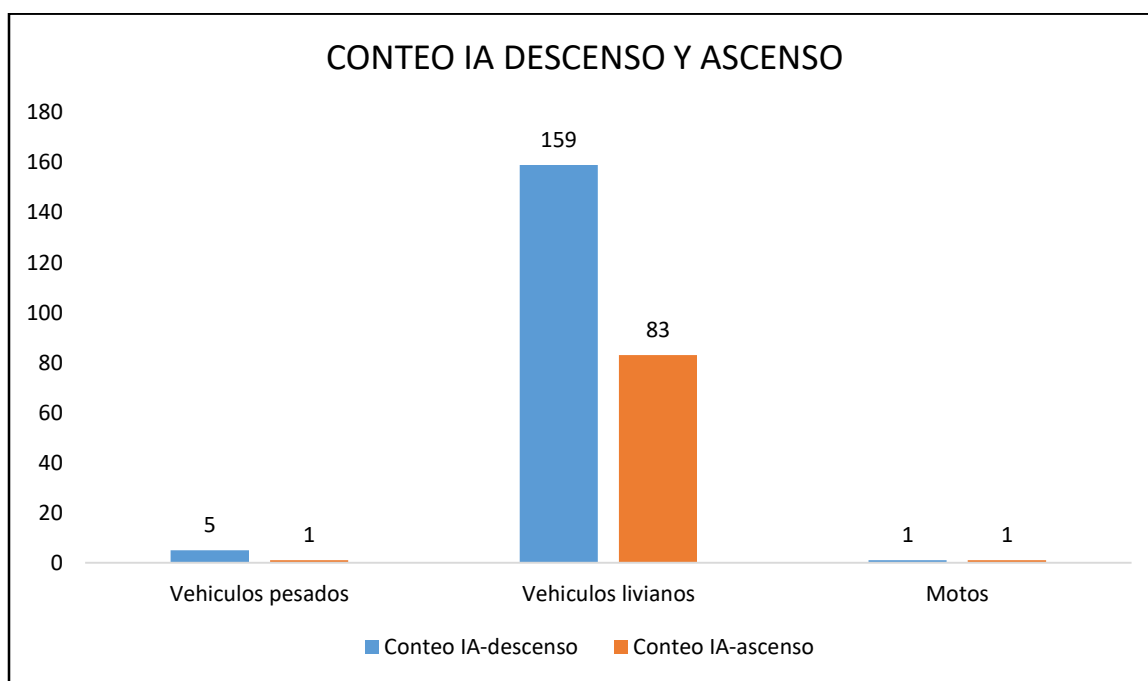
Nota: Datos obtenidos del conteo vehicular en las tres tipologías consideradas (livianos, pesados y motocicletas), incluyendo el total acumulado

La figura 52 muestra una gráfica del flujo vehicular en la intersección Hernán Malo y Panamericana tanto de descenso como ascenso en el horario de 07:30-8:00 am correspondiente a un periodo de alta demanda vehicular por la coincidencia con el ingreso a instituciones educativas y laborales, también permite observar el comportamiento direccional

del tránsito en un intervalo crítico del día. Por otro lado, la figura 53 evidencia el comportamiento del flujo vehicular en el mismo punto durante el horario de 17:30-18:00 pm, el cual coincide con el periodo de salida de actividades laborales y académicas. La comparación entre ambas figuras permite identificar patrones de congestión y posibles diferencias en la carga vehicular dependiendo del sentido de circulación y del momento del día.

Figura 52

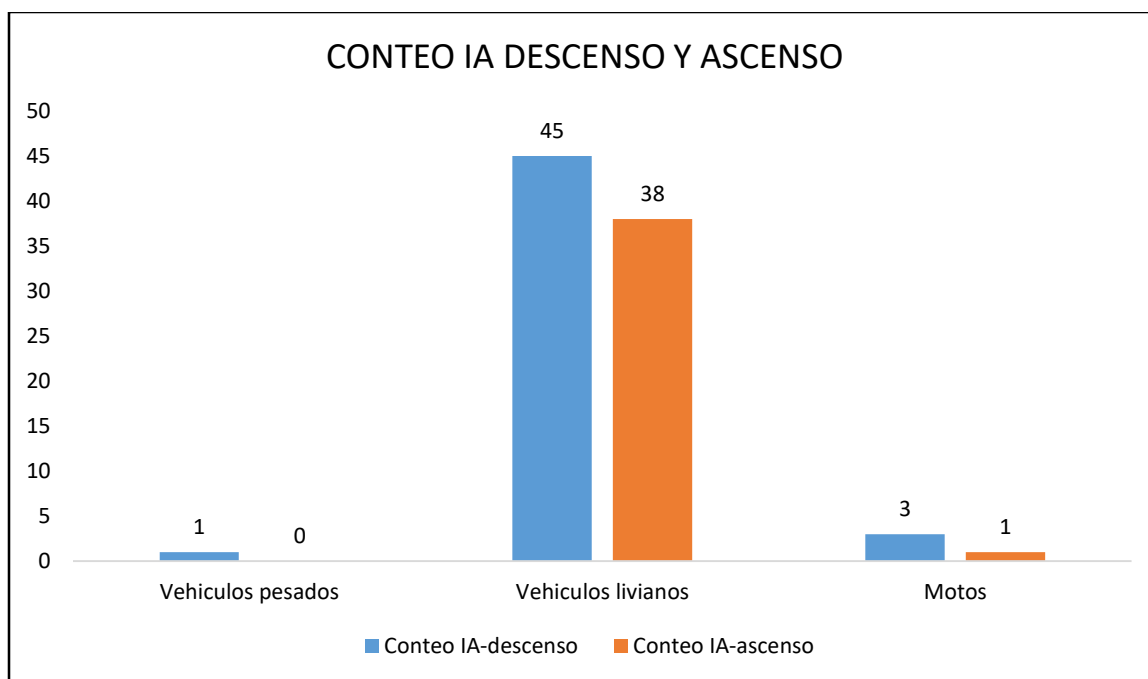
Grafica de flujo en la intersección Hernán M y Panamericana (07:30-8:00am)



Nota: La figura muestra el conteo vehicular mediante IA en ambos sentidos (subida y bajada) de la intersección entre la Hernán Malo y Panamericana en el horario de 7:30-8:00 am

Figura 53

Grafica de flujo en la intersección Hernán M y Panamericana (17:30-18:00pm)

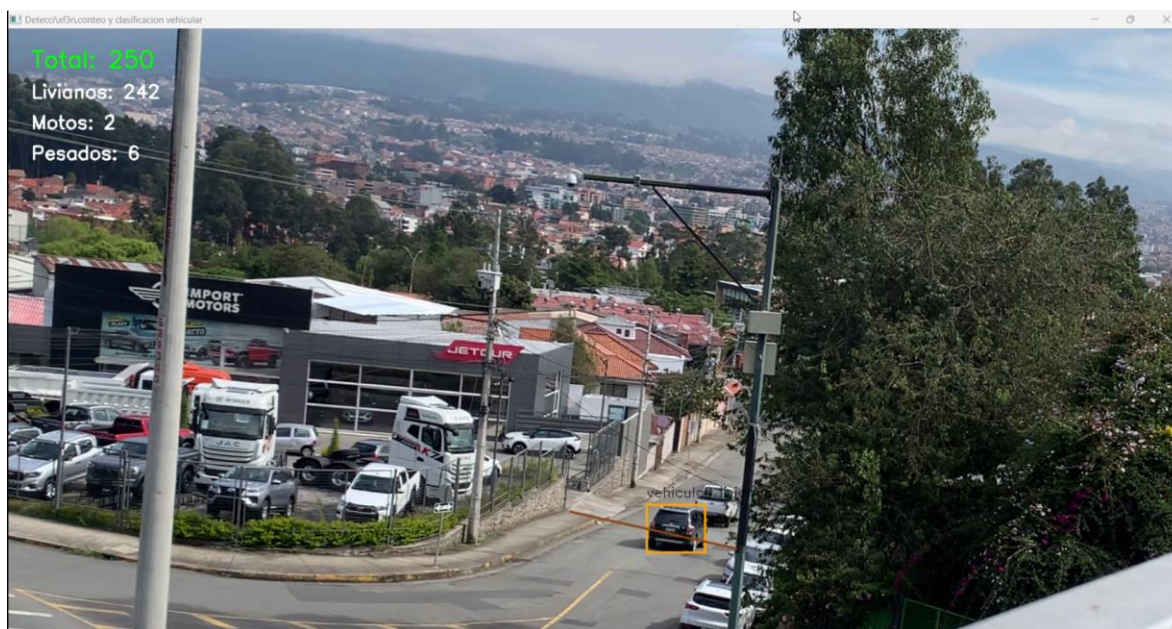


Nota: La figura muestra el conteo vehicular mediante IA en ambos sentidos (subida y bajada) de la intersección entre la Hernán Malo y Panamericana en el horario de 17:30-18:00 am

La detección de vehículos en la intersección seleccionada en el horario de 07:30 a 8:00 am presenta mayor afluencia de vehículos livianos con una cantidad total de 242 unidades, seguida de vehículos pesados con 6 unidades y finalmente las motos con 2 unidades. Mientras que en el horario de 17:30 a 18:00 pm presenta mayor afluencia de vehículos livianos con una cantidad total de 83 unidades, seguida de motos con 4 unidades y finalmente los vehículos pesados con 1 unidad. Lo mencionado anteriormente se evidencia en la figura 54 y 55, la cual es una captura de pantalla del video al finalizar el conteo IA en los horarios de 07:30-08:00 am y de 17:30 a 18:00 pm respectivamente.

Figura 54

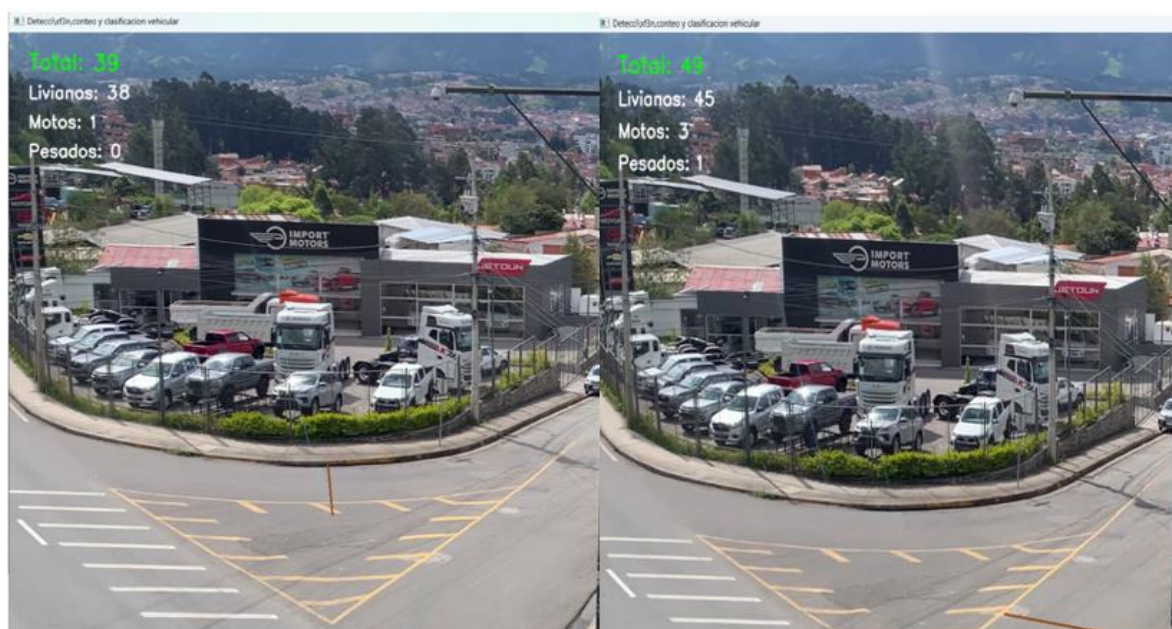
Pantalla final de video Hernán M y Panamericana, conteo IA-07:30 a 8:00 am.



Nota. Fuente: Elaboración propia

Figura 55

Pantalla final de video Hernán M y Panamericana, conteo IA-17:30 a 18:00 pm.



Nota. Fuente: Elaboración propia

Además, fue analizada la calle Panamericana al mismo tiempo para una mejor evaluación del algoritmo, para poder cuantificar de manera precisa el intercambio vehicular entre los dos puntos, es por eso que en la tabla 4 se muestra la cantidad de vehículos detectados en esa área en el horario de 07:30 a 8:00 am y en la tabla 5 la cantidad de vehículos detectados de 17:30 a 18:00 pm.

Tabla 4

Conteo de vehículos por clase en la calle Panamericana de 07:30 a 8:00 am.

Clase de vehículos	Conteo automático por algoritmo
Vehículos pesados	99
Vehículos livianos	844
Motos	50
Total	993

Nota: Conteo automático de las diferentes tipologías vehiculares (livianos, pesados y motocicletas) con su total, realizado mediante el algoritmo del proyecto en el horario 07:30 a 8:00 am.

Tabla 5

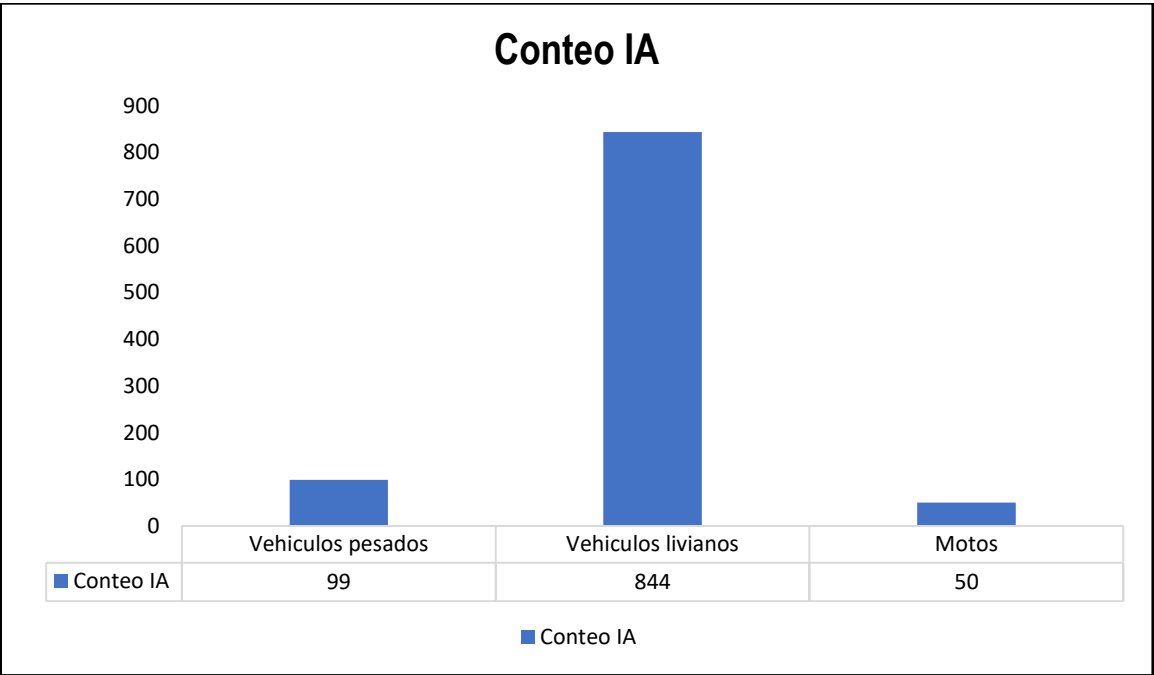
Conteo de vehículos por clase en la calle Panamericana de 17:30 a 18:00 am.

Clase de vehículos	Conteo automático por algoritmo
Vehículos pesados	20
Vehículos livianos	829
Motos	38
Total	887

Nota: Conteo automático de las diferentes tipologías vehiculares (livianos, pesados y motocicletas) con su total, realizado mediante el algoritmo del proyecto en el horario 17:30 a 18:00 pm.

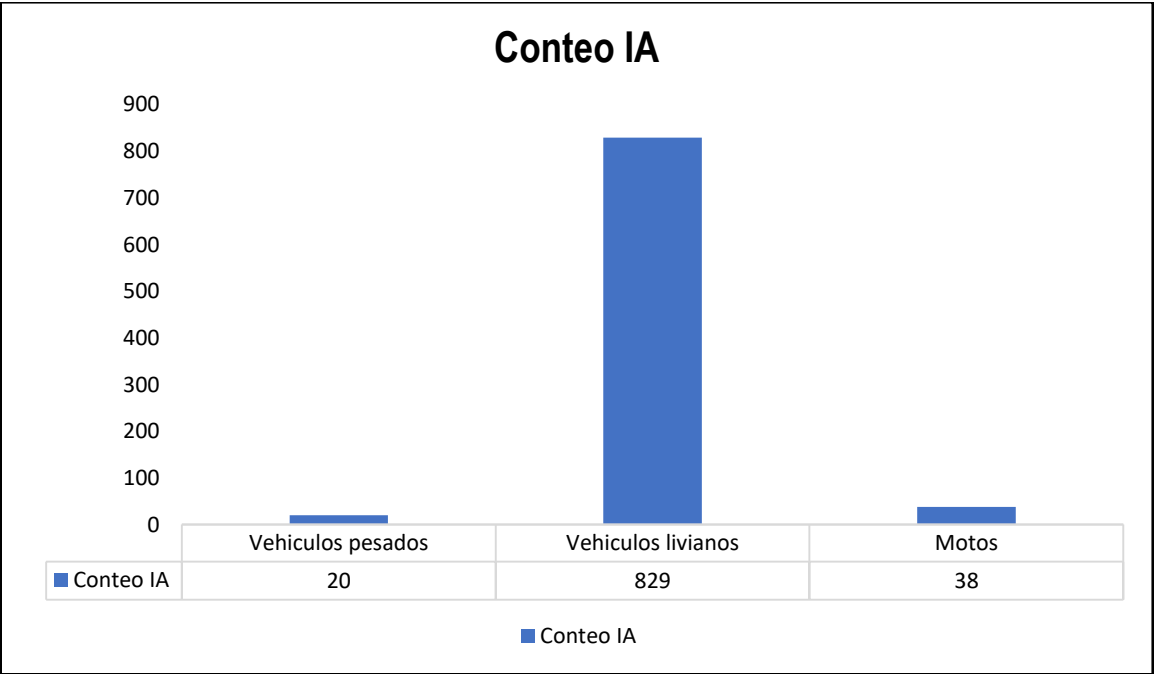
La figura 56 y 57 muestran el flujo vehicular durante el tiempo analizado.

Figura 56 Grafica de flujo vehicular en la calle Panamericana 07:30 a 8:00 am



Nota: La figura muestra el flujo vehicular registrado entre las 07:30 y las 08:00 am

Figura 57 Grafica de flujo vehicular en la calle Panamericana 17:30 a 18:00 pm



Nota: La figura muestra el flujo vehicular registrado entre las 17:30 y las 18:00 pm

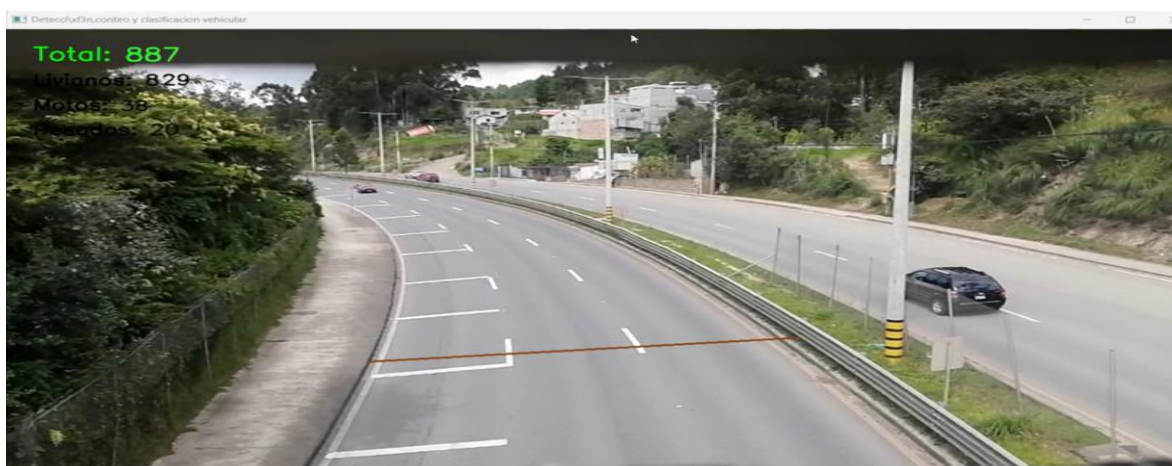
Finalmente, en la calle Panamericana la detección de vehículos en la intersección seleccionada en el horario de 07:30 a 8:00 am presenta mayor afluencia de vehículos livianos con una cantidad total de 844 unidades, seguida de vehículos pesados con 99 unidades y finalmente las motos con 50 unidades. Mientras que en el horario de 17:30 a 18:00 pm presenta mayor afluencia de vehículos livianos con una cantidad total de 829 unidades, seguida de motos con 38 unidades y finalmente los vehículos pesados con 20 unidades. Lo mencionado anteriormente se evidencia en la figura 58 y 59, la cual es una captura de pantalla del video al finalizar el conteo IA en los horarios de 07:30-08:00 am y de 17:30 a 18:00 pm respectivamente.

Figura 58 Pantalla final de video Panamericana con conteo IA de 07:30 a 08:00 am



Nota. Fuente: Elaboración propia

Figura 59 Pantalla final de video Panamericana con conteo IA de 17:30 a 18:00 pm



Nota. Fuente: Elaboración propia

4.3 Evaluación de la precisión del algoritmo

En la evaluación de la precisión del algoritmo se han realizado pruebas de conteo manual para obtener una referencia confiable para medir el rendimiento del algoritmo, ya que puede presentar errores debido a diversos factores. Al contar manualmente los vehículos mediante una grabación de video en la intersección Hernán Malo y Panamericana, se obtiene un valor real del conteo. Finalmente es importante mencionar que al comparar el conteo manual con el conteo automático es posible ajustar parámetros como la confianza para mejorar la precisión. A continuación, los datos reales del video en la intersección Hernán Malo-Panamericana y calle Panamericana tanto de descenso como de ascenso en los horarios de 07:30 a 08:00 am y de 17:30 a 18:00 pm se evidencian en la tabla 6 y 7. Así también, se evidencia en la tabla 8 y 9 el conteo real del video en la calle Panamericana en los horarios de 07:30 a 08:00 am y de 17:30 a 18:00 pm.

Tabla 6

Resultados de conteo real de vehículos (video) en la intersección Hernán Malo y

Panamericana de 07:30 a 08:00 am.

Clase	Conteo real de vehículos en el video-descenso	Conteo real de vehículos en el video-ascenso
Vehículos pesados	4	1
Vehículos livianos	158	80
Motos	1	1
Total, conteo	163	82

Nota: La tabla presenta los resultados del conteo vehicular manual en ambos sentidos (ascenso y descenso), realizados en intervalo de 07:30 y las 08:00 am.

Tabla 7

Resultados de conteo real de vehículos (video) en la intersección Hernán Malo y Panamericana de 17:30 a 18:00 am.

Clase	Conteo real de vehículos en el video-descenso	Conteo real de vehículos en el video-ascenso
Vehículos pesados	1	0
Vehículos livianos	45	39
Motos	3	1
Total, Conteo	49	40

Nota: La tabla presenta los resultados del conteo vehicular manual en ambos sentidos (ascenso y descenso), realizados en intervalo de 17:30 y las 18:00 pm.

Tabla 8

Resultados de conteo real de vehículos (video) en la calle Panamericana de 07:30-8:00 am.

Clase	Conteo real de vehículos en el video
Vehículos pesados	103
Vehículos livianos	848
Motos	49
Total, conteo	1000

Nota: La tabla presenta los resultados del conteo vehicular manual, realizados en intervalo de 07:30 y las 08:00 am.

Tabla 9

Resultados de conteo real de vehículos (video) en la calle Panamericana de 17:30-18:00 pm

Clase	Conteo real de vehículos en el video
Vehículos pesados	20
Vehículos livianos	830
Motos	40
Total, conteo	890

Nota: La tabla presenta los resultados del conteo vehicular manual, realizados en intervalo de 17:30 y las 18:00 pm.

Así como se realizó el conteo real de los vehículos en el video, también es importante realizar el conteo automático en base al algoritmo creado y presentar una comparación para obtener el margen de error. Los registros presentados mediante el conteo manual real total en el horario de 07:30 a 08:00 am para la intersección Hernán Malo y Panamericana son 245 vehículos y para la calle Panamericana son 1000 vehículos, mientras que con el conteo automático por algoritmo se registraron en la primera intersección 250 vehículos, es decir hay 5 vehículos faltantes y en la calle Panamericana se evidencian 993 vehículos, es decir hay 7 vehículos faltantes. Mediante la tabla 10 y 11 se evidencia que el algoritmo presenta más fallas en los vehículos pesados.

Tabla 10

Comparación de resultados obtenidos en la intersección Hernán Malo y Panamericana 07:30 a 08:00 am.

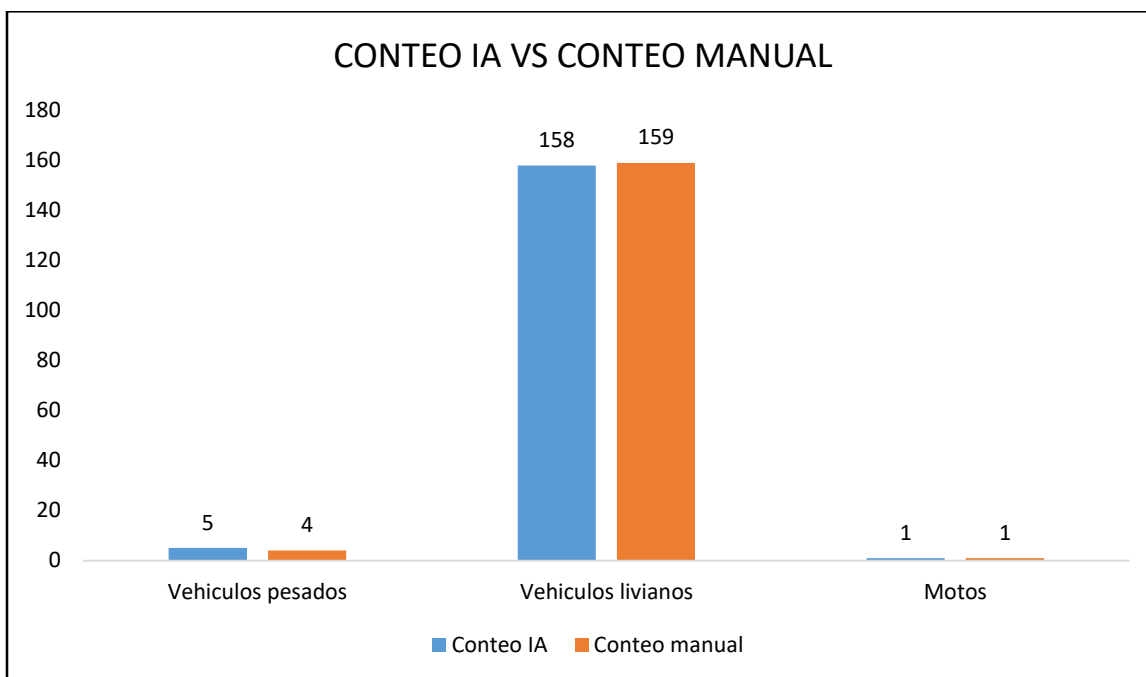
Clase	Conteo real	Conteo IA Descenso	Conteo real	Conteo IA Ascenso
Vehículos pesados	4	5	1	1
Vehículos livianos	158	159	80	83
Motos	1	1	1	1
Total	163	165	82	85

Nota: Comparación de los resultados del conteo vehicular en ambos sentidos de circulación (ascenso y descenso) en la intersección analizada, utilizando métodos manuales y el sistema del proyecto basado en IA.

La figura 60 y 61 muestran una gráfica de comparación de resultados entre conteo IA y conteo manual de la intersección Hernán Malo y Panamericana tanto de descenso como de ascenso.

Figura 60

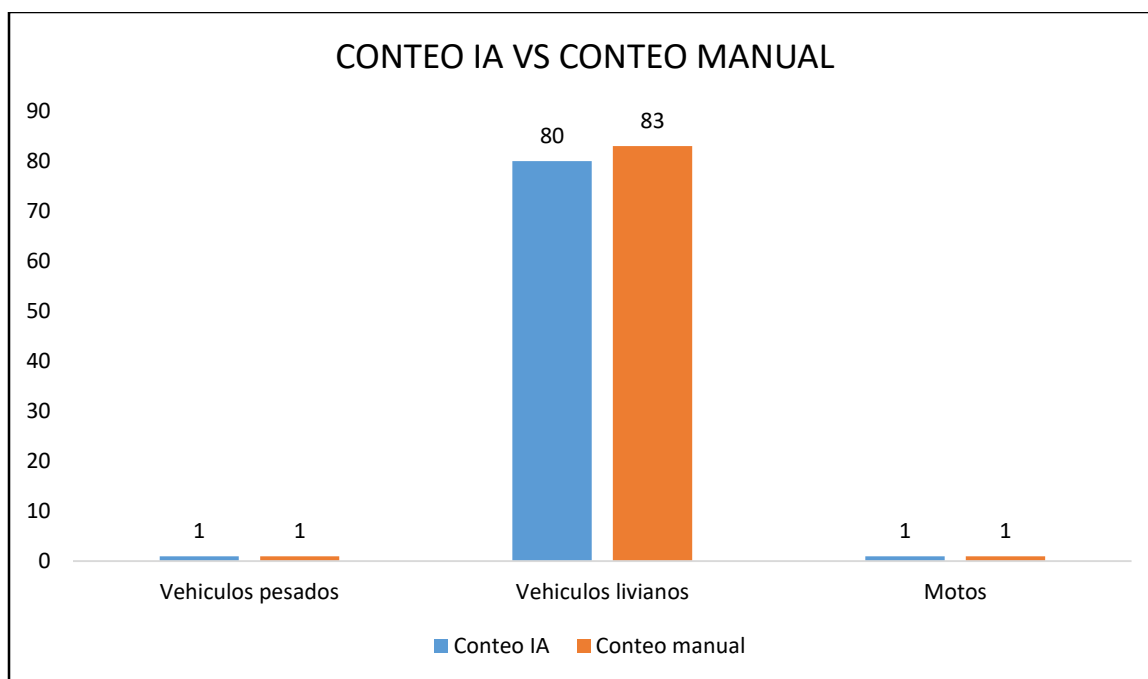
Comparación entre conteo IA vs conteo manual Descenso Panamericana- Hernán Malo



Nota: La figura presenta la comparación de los resultados del conteo vehicular manual y el de nuestro proyecto en sentido de descenso.

Figura 61

Comparación entre conteo IA vs conteo manual Ascenso Hernán Malo-Panamericana



Nota: La figura presenta la comparación de los resultados del conteo vehicular manual y el de nuestro proyecto en sentido de ascenso.

Tabla 11

Comparación de resultados obtenidos en la calle Panamericana

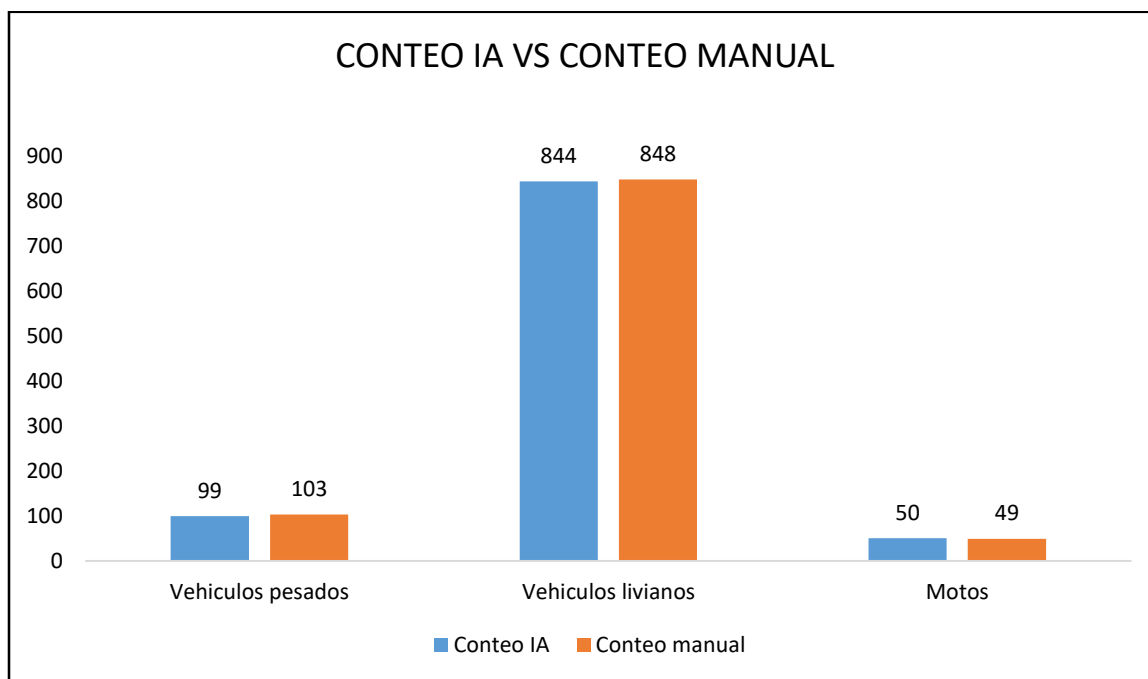
Clase	Conteo real	Conteo IA
Vehículos pesados	103	99
Vehículos livianos	848	844
Motos	49	50
Total	1000	993

Nota: comparación de los resultados del conteo vehicular en la panamericana, realizados tanto de forma manual y con IA.

La figura 62 muestra una gráfica de comparación de resultados entre conteo IA y conteo manual de la calle panamericana.

Figura 62

Comparación entre conteo IA vs conteo manual Panamericana



Nota: La figura presenta la comparación de los resultados del conteo vehicular manual y el de nuestro proyecto en la Panamericana sentido Norte-Sur.

Así mismo se presenta una comparación para obtener el margen de error. Los registros presentados mediante el conteo manual real total en el horario de 17:30 a 18:00 pm para la intersección Hernán Malo y Panamericana son 89 vehículos y para la calle Panamericana son 890 vehículos, mientras que con el conteo automático por algoritmo se registraron en la primera intersección 88 vehículos, es decir hay 1 vehículo faltante y en la calle Panamericana se evidencian 887 vehículos, es decir hay 3 vehículos faltantes. Mediante la tabla 12 y 13 se evidencia que el algoritmo presenta más fallas en las motos.

Tabla 12

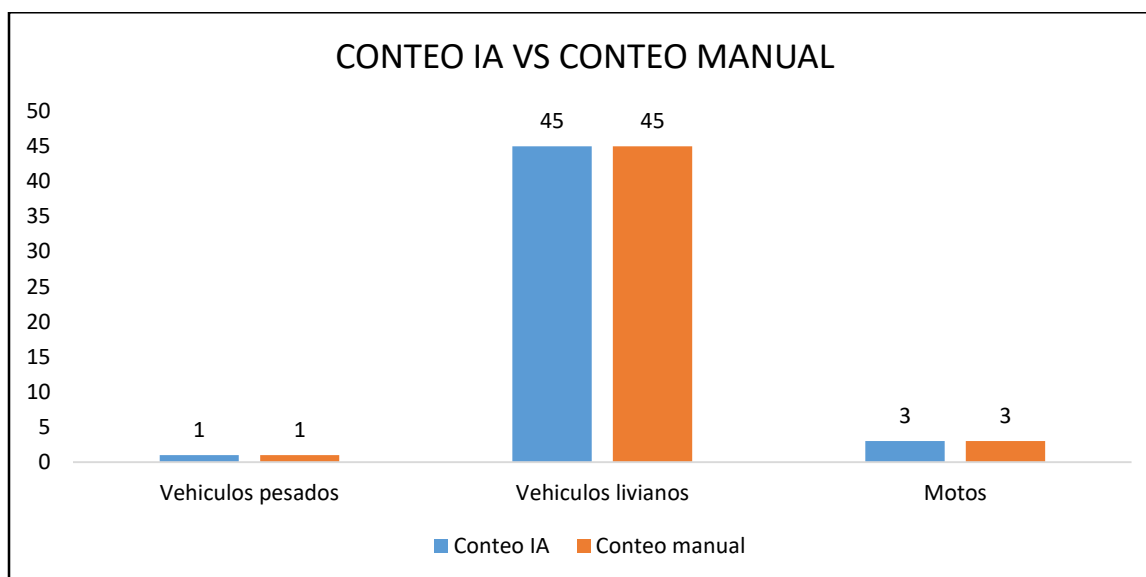
Comparación de resultados obtenidos en la intersección Hernán Malo y Panamericana

Clase	Conteo real	Conteo IA	Conteo real	Conteo IA
		Descenso		Ascenso
Vehículos pesados	1	1	0	0
Vehículos livianos	45	45	39	38
Motos	3	3	1	1
Total	49	49	40	39

Nota: Se presenta la distribución de las diferentes tipologías vehiculares, junto con la comparación entre el conteo manual y el realizado mediante IA, en ambos sentidos de circulación (ascenso y descenso).

La figura 63 y 64 muestran una gráfica de comparación de resultados entre conteo IA y conteo manual de la intersección Hernán Malo y Panamericana tanto de descenso como de ascenso.

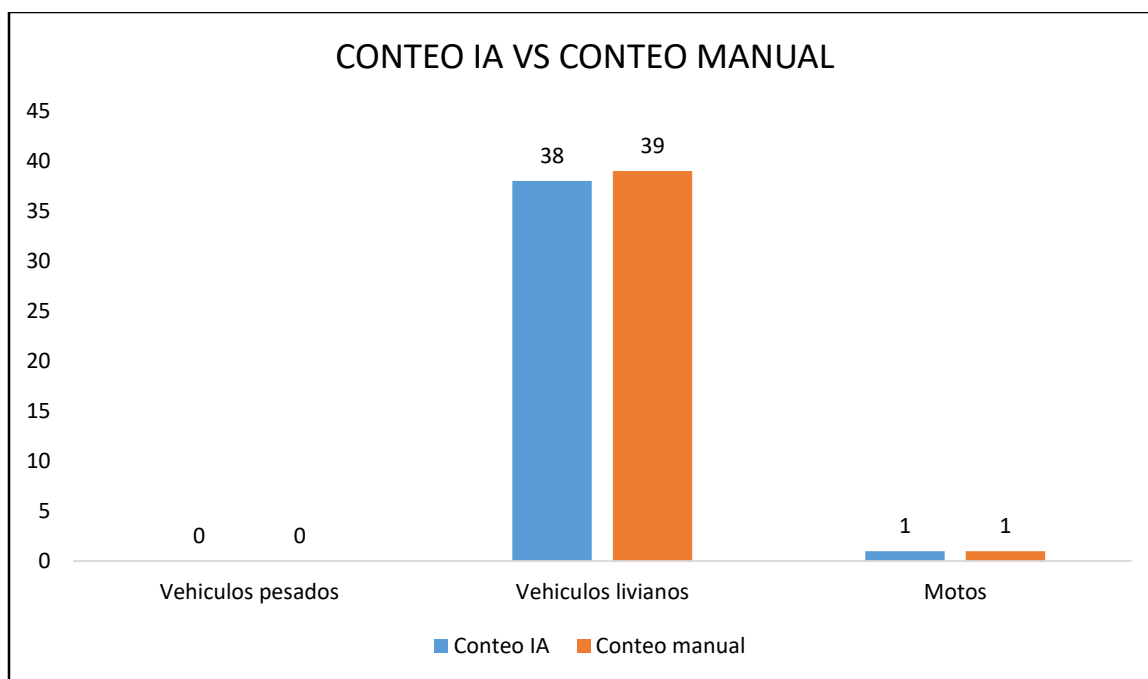
Figura 63 *Comparación entre conteo IA vs conteo manual Descenso Panamericana-Hernán Malo*



Nota: La figura presenta los resultados del conteo manual y con IA en sentido de descenso.

Figura 64

Comparación entre conteo IA vs conteo manual Ascenso Panamericana- Hernán Malo



Nota: La figura presenta la comparación de los resultados del conteo vehicular manual y el de nuestro proyecto en sentido de ascenso.

Tabla 13

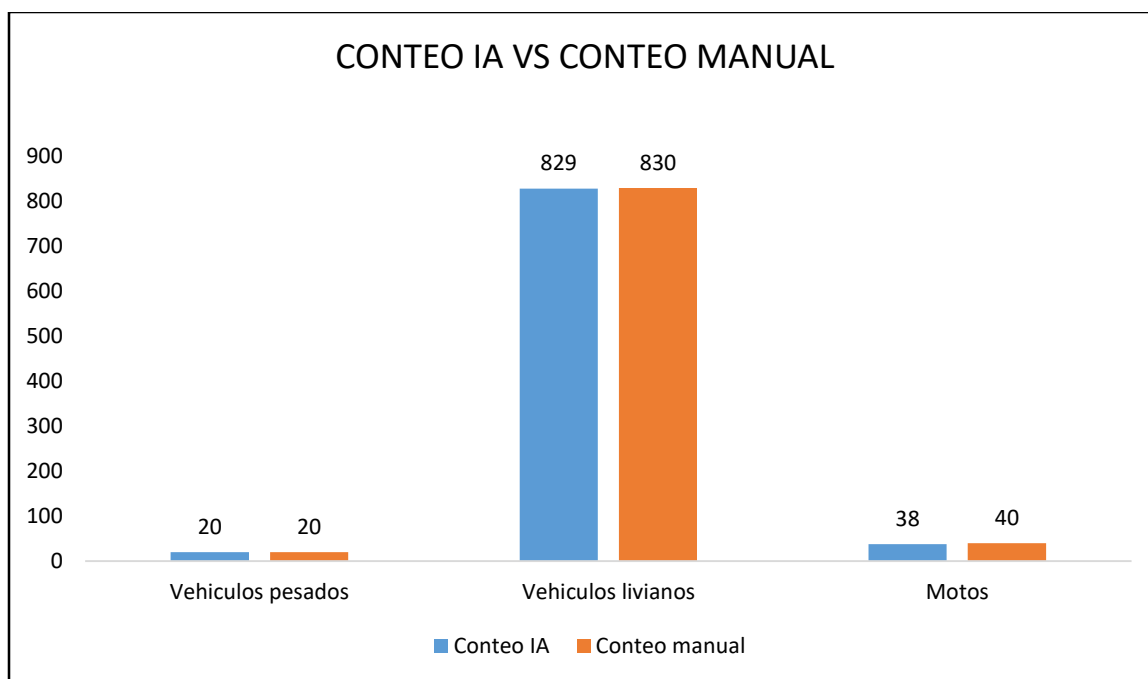
Comparación de resultados obtenidos en la calle Panamericana

Clase	Conteo real	Conteo IA
Vehículos pesados	20	20
Vehículos livianos	830	829
Motos	40	38
Total	890	887

Nota: Se presenta la distribución de las diferentes tipologías vehiculares, junto con la comparación entre el conteo manual y el realizado mediante IA en la calle Panamericana. La figura 65 muestra una gráfica de comparación de resultados entre conteo IA y conteo manual de la calle panamericana.

Figura 65

Comparación entre conteo IA vs conteo manual Panamericana



Nota: La figura presenta la distribución de las diferentes tipologías vehiculares, junto con la comparación entre el conteo manual y el realizado mediante IA en la calle Panamericana.

Además, en la tabla 14, 15, 16 y 17, se evidencia la efectividad del algoritmo el cual fue calculado con la ecuación 9 en los horarios de 07:30 a 08:00 am y de 17:30 a 18:00 pm. Realizando un análisis en el horario de 07:30 a 08:00 am se obtiene una mayor precisión en los vehículos livianos con el 98,3%, y en la clase de motos de igual manera con una precisión de más del 98%, mientras que el algoritmo presenta menor efectividad en vehículos pesados debido a diversos factores como el tamaño y forma del vehículo, ya que tienen una estructura compleja, o las condiciones de iluminación y superposición con otros objetos lo que dificulta su detección y clasificación precisa. Además, en el horario de 17:30 a 18:00 pm se presenta mayor precisión en vehículos pesados debido a que en ese horario se presentó menor flujo vehicular por lo tanto no existió la superposición con otros objetos. Finalmente, la efectividad total del algoritmo para la detección, conteo y clasificación vehicular en donde circularon 1000 vehículos fue del 99,30%, mientras que donde circularon 890 fue del 99,66%, del cual haciendo un promedio general se obtuvo un error del 0,52% para ambos análisis.

Ecuación 9: Efectividad

$$\% \text{ Efectividad} = \frac{N^{\circ} \text{ de contador automatico}}{N^{\circ} \text{ real de vehiculos (video)}} * 100$$

Tabla 14

Resultados de efectividad del conteo automático en la intersección Hernán Malo y Panamericano tanto en descenso como ascenso de 07:30 a 08:00 am.

Clase	Conteo real de vehículos en el video	Conteo automático por algoritmo	% Efectividad
Vehículos pesados	5	6	83,33%
Vehículos livianos	238	242	98,35%
Motos	2	2	100%
Total	245	250	98%

Nota: Se presenta los resultados del nivel de efectividad del conteo automático realizado mediante IA y forma manual en la intersección analizada en horario de 07:30 a 08:00, expresados en porcentaje de efectividad.

Tabla 15

Resultados de efectividad del conteo automático en la intersección Hernán Malo y Panamericano tanto en descenso como ascenso de 17:30 a 18:00 pm.

Clase	Conteo real de vehículos en el video	Conteo automático por algoritmo	% Efectividad
Vehículos pesados	1	1	100%
Vehículos livianos	84	83	98,8%
Motos	4	4	100%
Total	89	88	98,87%

Nota: Se presenta los resultados del nivel de efectividad del conteo automático realizado mediante IA y forma manual en la intersección analizada en horario de 17:30 a 18:00, expresados en porcentaje de efectividad.

Tabla 16

Resultados de efectividad del conteo automático en la calle Panamericana de 07:30 a 08:00 am.

Clase	Conteo real de vehículos en el video	Conteo automático por algoritmo	% Efectividad
Vehículos pesados	103	99	96,11%
Vehículos livianos	848	844	99,5%
Motos	49	50	98%
Total	1000	993	99,30%

Nota: Se presenta los resultados del nivel de efectividad del conteo automático realizado mediante IA y forma manual en la calle Panamericana en horario de 07:30 a 08:00, expresados en porcentaje de efectividad.

Tabla 17

Resultados de efectividad del conteo automático en la calle Panamericana de 17:30 a 18:00 pm.

Clase	Conteo real de vehículos en el video	Conteo automático por algoritmo	% Efectividad
Vehículos pesados	20	20	100%
Vehículos livianos	830	829	99,88%
Motos	40	38	95%
Total	890	887	99,66%

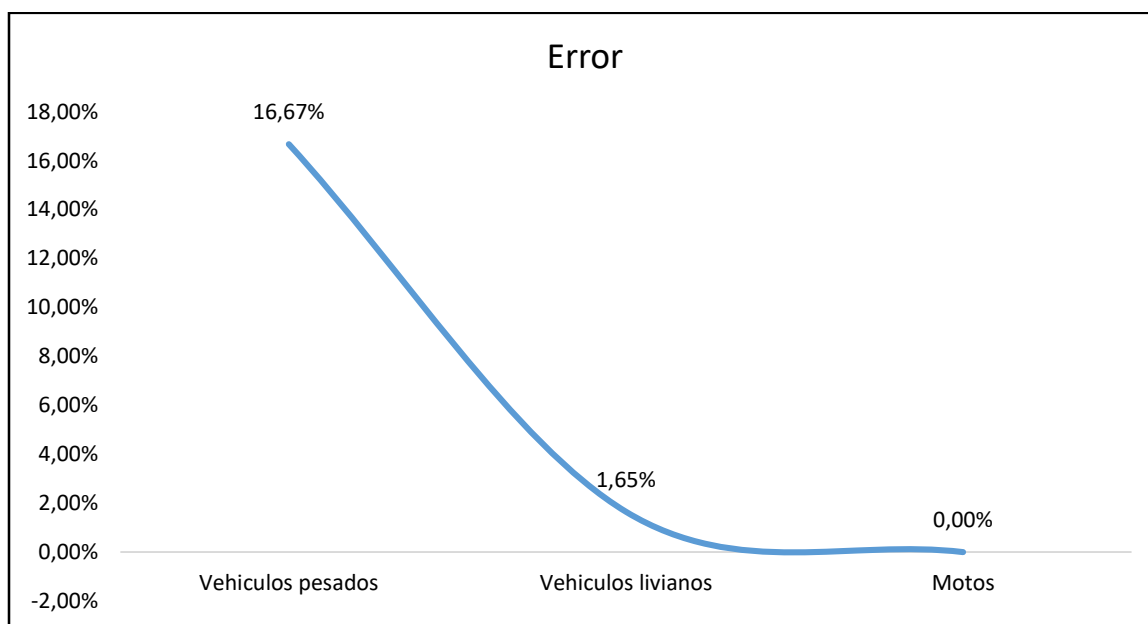
Nota: Se presenta los resultados del nivel de efectividad del conteo automático realizado mediante IA y forma manual en la calle Panamericana en horario de 17:30 a 18:00, expresados en porcentaje de efectividad.

Como se mencionó anteriormente, el contador de vehículos presentaba un error del 0,52%, sin embargo, también se mencionó la efectividad por clases, lo cual se evidencia en la figura

66,67,68 y 69 en los horarios de 07:30-08:00 am y 17:30 a 18:00 pm, donde se observa que en la intersección Hernán Malo y Panamericana tanto de descenso como ascenso el mayor error se presenta en la clase de vehículos pesados, mientras que en la calle Panamericana se observa el mayor error en la clase de motos.

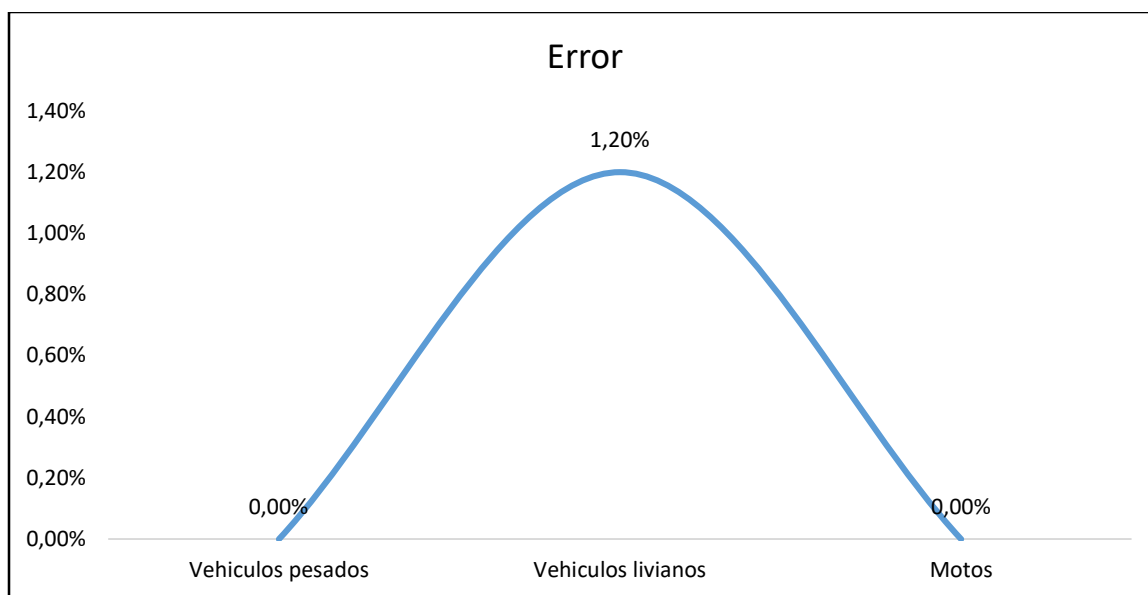
Figura 66

Error presentado por el contador IA en la Hernán Malo y Panamericana de 07:30 a 08:00 am.



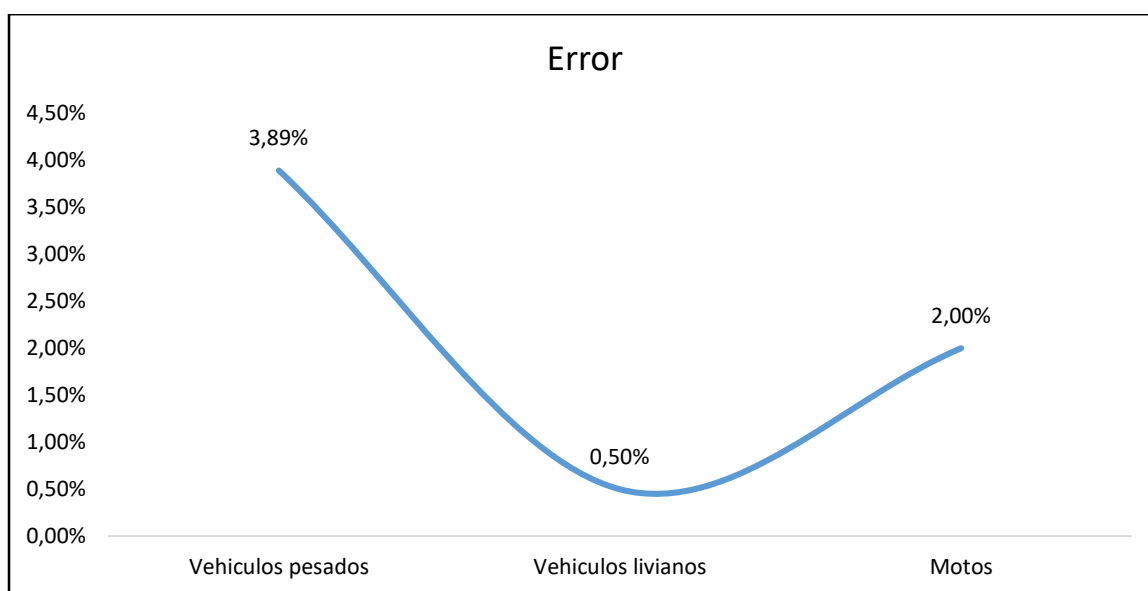
Nota: La figura muestra el porcentaje de error en la clasificación automática de vehículos por tipología (pesados, livianos y motocicletas) obtenido mediante el sistema de IA en horario de 07:30 a 08:00 en la intersección analizada.

Figura 67 Error presentado por el contador IA en la Hernán Malo y Panamericana de 17:30 a 18:00 pm



Nota: La figura muestra el porcentaje de error en la clasificación automática de vehículos por tipología (pesados, livianos y motocicletas).

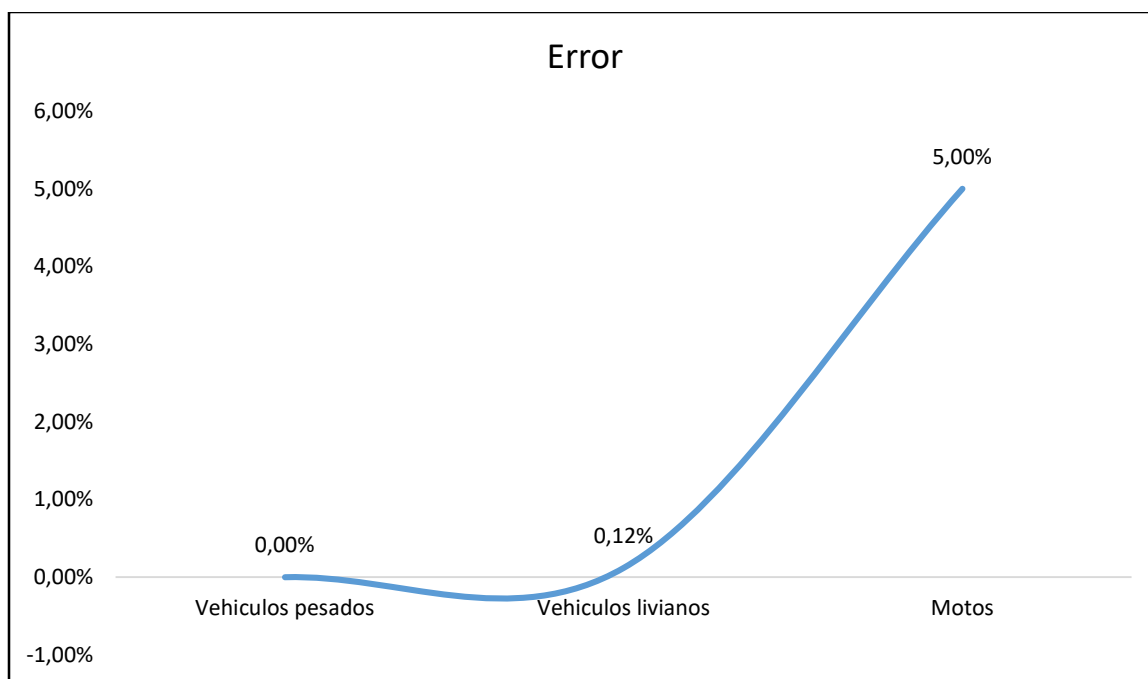
Figura 68 Error presentado por el contador IA en la Panamericana de 07:30 a 08:00 am



Nota: La figura muestra el porcentaje de error en la clasificación automática de vehículos por tipología (pesados, livianos y motocicletas).

Figura 69

Error presentado por el contador IA en la Panamericana de 17:30 a 18:00 pm



Nota: La figura muestra el porcentaje de error en la clasificación automática de vehículos por tipología (pesados, livianos y motocicletas) obtenido mediante el sistema de IA en horario de 17:30 a 18:00 en la calle Panamericana.

4.3.1 Análisis de la efectividad del modelo de clasificación

En el análisis del flujo vehicular se calculó el flujo vehicular denominado Q4 correspondiente a los vehículos que continúan su trayecto por la autopista Panamericana. Este flujo se obtuvo mediante la ecuación 8.

Este enfoque no solo permite estimar flujos no captados visualmente, sino que también facilita identificar giros, que en este caso se encuentran representados por el flujo Q2 (giros a la derecha). Por tanto, este flujo brinda una medición directa de cuantos vehículos abandonan la Panamericana en dirección a la Hernán Malo.

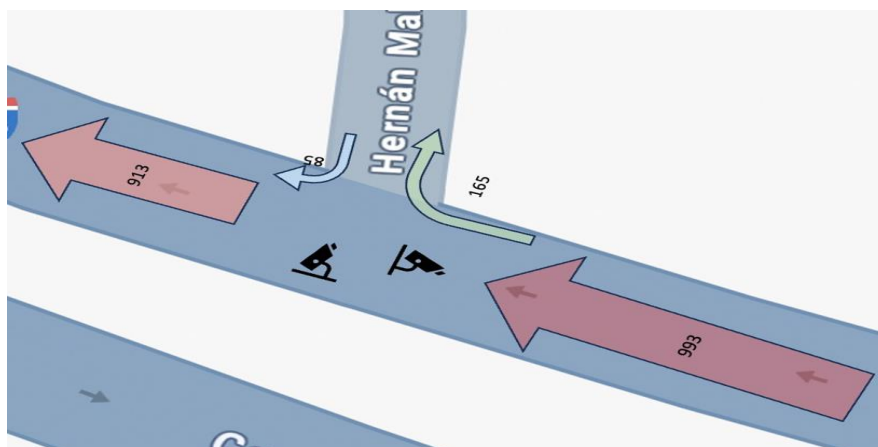
Para llevar a cabo este análisis, se utilizaron los datos obtenidos en dos horarios representativos del día: 07:30 a 08:00 am y 17:30 a 18:00 pm. A continuación, se evidencia la tabla 18 en los dos horarios con los resultados obtenidos:

Tabla 18*Resultados del análisis de flujos vehiculares en dos horarios*

Horario	Q1 (Panamericana)	Q2 (Giro derecho a Hernán Malo)	Q3 (Hernán Malo- Panamericana)	Q4 (Estimado)
07:30 a 08:00	993	165	85	913
17:30 a 18:00	887	49	39	877

Nota: Resultados del análisis comparativo del flujo vehicular en 4 cuadrantes de la intersección que son los flujos dependiendo su direccionalidad y el flujo total en los diferentes horarios.

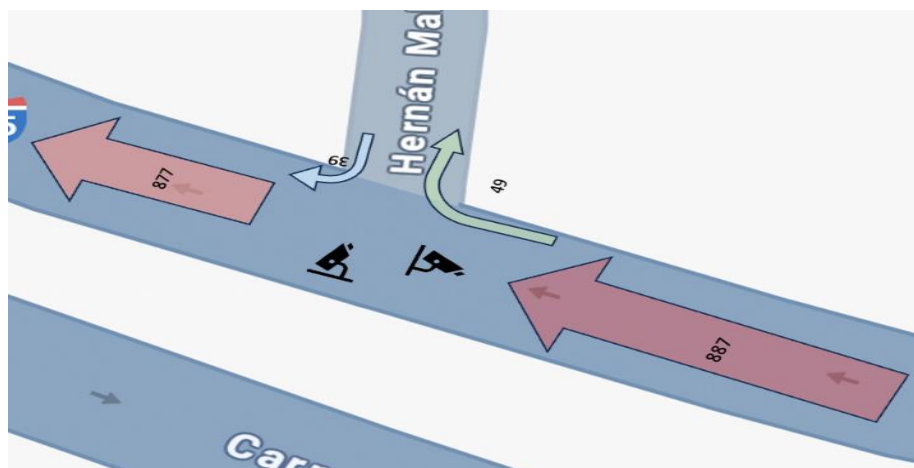
Además, en la figura 70 y 71 se evidencian dichos resultados. Como se puede observar, el flujo Q2 permite identificar con precisión el número de vehículos que realizan el giro hacia la avenida Hernán Malo en cada horario. Además, el flujo estimado Q4 representa una porción significativa del tráfico que permanece en la Panamericana, siendo mayor en la mañana, lo que sugiere una posible tendencia de retorno vehicular en ese periodo. Este valor resulta clave para completar el análisis del comportamiento vehicular en la intersección, ya que permite compensar las limitaciones de visibilidad del sistema de grabación y obtener un panorama más preciso del volumen total de tránsito en la zona.

Figura 70 *Resultados del análisis de flujos vehiculares en dos horarios 07:30 a 08:00 am*

Nota. Fuente: Elaboración propia

Figura 71

Resultados del análisis de flujos vehiculares en dos horarios 17:30 a 18:00 pm



Nota. Fuente: Elaboración propia

Una vez entrenado el modelo de clasificación, se evaluó su rendimiento aplicándolo a un conjunto de datos de prueba representativo de las condiciones de tráfico reales. El modelo se entrenó para diferenciar entre tres categorías: motocicletas, vehículos livianos y vehículos pesados. Los resultados obtenidos reflejan un rendimiento general positivo, con una precisión del 97,4 %, lo que indica que el modelo puede clasificar eficazmente los casos propuestos.

En cuanto a las métricas específicas, el modelo obtuvo los siguientes resultados: precisión del 97,4 % y precisión por clase del 98,0 % para motocicletas, del 100 % para vehículos ligeros y del 99 % para vehículos pesados, lo que resulta en una tasa de error del 2,0 % para los vehículos pesados y del 1,0 % para las motocicletas. Este tipo de confusión es comprensible debido a las similitudes visuales en ciertos ángulos de cámara. Estos resultados demuestran un equilibrio adecuado entre la capacidad del modelo para detectar y clasificar correctamente los vehículos sin sesgo excesivo hacia ninguna categoría.

Las curvas de evaluación del modelo (véanse las Figuras 34 a 37) muestran varias relaciones clave para analizar el rendimiento del clasificador. En particular, se presentan las curvas de Precisión vs. *Recall*, Precisión vs. Confianza, F1 vs. Confianza y *Recall* vs. Confianza, que permiten observar cómo varía el rendimiento del modelo en función de diferentes umbrales de decisión. El área bajo la curva (AUC) supera 0,96 para todas las clases, lo que demuestra

la buena capacidad discriminatoria del modelo incluso en diversas condiciones. Cabe destacar que se realizaron pruebas con diferentes valores de umbral de decisión, lo que confirma que el modelo mantiene una consistencia aceptable en su rendimiento.

A nivel operativo, estos resultados son ideales para implementar el sistema en escenarios reales de monitorización de vehículos. Al no depender de sensores físicos, el sistema basado en visión artificial se adapta fácilmente a diferentes ubicaciones y condiciones ambientales. Esto lo convierte en una herramienta valiosa para el análisis automatizado del flujo de tráfico en entornos urbanos como la ciudad de Cuenca, donde la congestión y la diversidad vehicular representan un desafío constante.

CAPITULO V

Capítulo 5: Análisis

5.1 Interpretación de los resultados

En esta sección, buscamos interpretar los resultados obtenidos del análisis del flujo de tráfico, centrándonos en la comparación entre el conteo manual y el conteo obtenido mediante el algoritmo. Esto nos permite evaluar la precisión del sistema automatizado de detección, conteo y clasificación, así como identificar los factores que podrían haber influido en los márgenes de error observados.

5.1.1 Factores que podrían afectar los resultados

El sistema de conteo automático mostró un rendimiento excepcional, con un error promedio del 0,62 % en ambas intersecciones analizadas (Hernán Malo y Panamericana, y Calle Panamericana). Este bajo margen de error sugiere que el algoritmo de detección y clasificación tiene una robusta capacidad para realizar conteos precisos en situaciones de tráfico reales.

La efectividad general del sistema fue del 98,95%, con una ligera variación en los porcentajes de precisión según la clase de vehículo. La clase con menor precisión fue la de vehículos pesados (94,86%), seguida de las motocicletas (98,25%) y, finalmente, la de vehículos livianos, que mostró la mayor efectividad, con un 99,14%.

En la vía Panamericana, la precisión por clase fue similar a la observada en la intersección, con un rendimiento superior para los vehículos livianos (99,39 %) y un error más pronunciado para las motocicletas (95,74 %).

Los vehículos pesados presentaron el mayor margen de error, con una diferencia de 1 vehículo en la calle Panamericana (1000 vehículos reales frente a los 993 detectados por el algoritmo en el horario matutino y 890 vehículos reales frente a los 887 vehículos capturados con el algoritmo). Este tipo de vehículo presenta una estructura compleja y grandes dimensiones, lo que puede dificultar una clasificación precisa, especialmente a la circular junto a otros vehículos o en situaciones de tráfico congestionado. Además, la superposición con otros vehículos o la obstrucción parcial de la visión (por ejemplo, al ser parcialmente bloqueado por vehículos más pequeños) pueden afectar el rendimiento del sistema. Las condiciones de iluminación también pueden ser cruciales.

Los vehículos livianos, por otro lado, mostraron una alta efectividad de detección, con una diferencia de uno a cuatro vehículos en el conteo total. En la calle Panamericana, se detectaron 848 vehículos, en comparación con los 844 reales en el primer horario y en el segundo 830 vehículos y 829 vehículos reales, lo que representa una tasa de precisión del 99,39 %. Este alto nivel de precisión se puede atribuir a varias características que facilitan la detección:

- Los vehículos livianos son más visibles en las imágenes debido a su tamaño relativamente grande en comparación con las motocicletas.
- La presencia de un mayor número de vehículos livianos en las intersecciones facilita la clasificación por parte del algoritmo, ya que identifica patrones repetitivos en el flujo vehicular.

Sin embargo, en escenarios con tráfico muy denso o con vehículos muy juntos, incluso los vehículos ligeros pueden presentar ligeros errores, aunque estos sean muy pequeños en comparación con otras clases de vehículos.

Motocicletas: Las motocicletas presentaron un mayor margen de error, especialmente en la intersección de la calle Panamericana, con un error de 1 a 2 vehículos en los diferentes horarios (49 motocicletas reales frente a las 50 detectadas automáticamente y 70 y 38

motocicletas respectivamente). Esto representa una efectividad del 99,34%, que, aunque sigue siendo alta, demuestra que la detección de motocicletas sigue siendo uno de los aspectos más desafiantes para los sistemas automatizados. Varias razones pueden explicar este fenómeno:

- Las motocicletas tienen una huella visual reducida, lo que dificulta su detección, especialmente en condiciones de tráfico congestionado o cuando se encuentran cerca de otros vehículos más grandes que podrían obstruirlas o bloquearlas parcialmente.
- Las motocicletas, al ser más bajas y difíciles de identificar visualmente en el tráfico, pueden verse fácilmente obstruidas por la presencia de vehículos más grandes o por variaciones en las condiciones de iluminación.

CAPITULO VI

Capítulo 6: Conclusiones y Recomendaciones

6.1 Conclusiones

El presente estudio desarrolló una solución basada en inteligencia artificial para la detección, el conteo, la clasificación y el análisis integral del flujo vehicular, la cual demostró ser efectiva. El sistema logro una precisión global superior al 98% en el conteo de vehículos, validando su fiabilidad como herramienta inteligente para el análisis vehicular.

Se implementó un sistema de algoritmos avanzados de visión artificial e inteligencia artificial ha mejorado significativamente la gestión del tráfico, optimizando la toma de decisiones en tiempo real y proporcionando herramientas útiles para el análisis detallado del comportamiento del tráfico.

Vehículos livianos tuvo la precisión de 98,35% y 99,88% de mañana y tarde respectivamente siendo la categoría más precisa gracias a su frecuencia de aparición, tamaño y silueta.

La precisión de vehículos pesados un margen de 83,33% y 96,11% en los dos horarios establecidos la variabilidad que se nota es por lo poca circulación de esta tipología de vehículos en los diferentes horarios.

Las motocicletas tuvieron una precisión entre 95% y 100%estas fluctuaciones fueran específicamente por la superposición con vehículos de mayor tamaño

Los resultados obtenidos demuestran que la solución propuesta es eficaz para abordar problemas complejos de tráfico urbano y ofrece una mejora tangible en la eficiencia de la gestión del tráfico.

La ubicación de la cámara resultó ser un factor determinante en el rendimiento del sistema. Se observó que la superposición de vehículos y los ángulos de visión deficientes pueden provocar conteos duplicados o sub-detecciones, principalmente en vehículos pesados y motocicletas. El algoritmo de inteligencia artificial diseñado para clasificar vehículos en tiempo real, se logró una alta precisión en la clasificación de vehículos livianos, pesados y motocicletas. El uso de redes neuronales profundas (Kernels) permitió una clasificación eficiente y rápida, incluso en condiciones de tráfico muy denso. Esta capacidad de clasificar vehículos en tiempo real no solo mejora la comprensión del flujo vehicular, sino que también facilita la toma de decisiones para optimizar las rutas, lo que demuestra la eficacia del algoritmo desarrollado.

El algoritmo de clasificación, basado en redes neuronales profundas con núcleos convolucionales, demostró su capacidad para realizar clasificaciones en tiempo real con alta precisión, incluso en condiciones de tráfico intenso. Esta capacidad facilita una toma de decisiones más eficiente, lo que permite la priorización de rutas, el rediseño de infraestructuras y una mejor gestión de la movilidad urbana.

En conclusión, el desarrollo e implementación de una solución basada en inteligencia artificial para la detección, conteo, clasificación y análisis del flujo vehicular ha demostrado ser una gran herramienta para realizar análisis de tráfico en tiempo real. Los resultados obtenidos validan la eficacia de la solución propuesta, no solo en términos de precisión y eficiencia en tiempo real, sino también en su capacidad para contribuir a la optimización de los procesos de toma de decisiones en entornos urbanos. Esta investigación sienta las bases para futuras mejoras y aplicaciones, como la integración con sistemas de transporte inteligentes o el análisis de datos más complejos, que podrían ofrecer beneficios aún mayores en la planificación y gestión del tráfico urbano. Por lo tanto, este trabajo abre la puerta a nuevas oportunidades de investigación y desarrollo que pueden mejorar la eficiencia de la infraestructura vial y la movilidad en las ciudades del futuro.

6.2 Recomendaciones

Con base en los resultados obtenidos y un análisis crítico de las condiciones que afectan al sistema, se proponen las siguientes recomendaciones estratégicas para fortalecer y ampliar el alcance de este proyecto:

Para mejorar la precisión de la detección de motocicletas y vehículos pesados, se recomienda aumentar el tamaño y la diversidad de la base de datos utilizada durante el entrenamiento del modelo. Incluir más imágenes que representen este tipo de vehículos en diferentes contextos (como ángulos variados, condiciones climáticas diversas, iluminación nocturna y escenarios de alta densidad de vehículos) permitirá que el sistema aprenda patrones más robustos y generalizables, reduciendo así los errores de clasificación en entornos urbanos complejos implementando de cámaras con especificaciones técnicas mejoradas.

Para asegurar que el sistema funcione correctamente en todo momento y en diferentes lugares, se recomienda incluir una forma de comparar, de vez en cuando, los resultados del conteo automático con algunos conteos realizados manualmente. Esta comparación ayudará a detectar posibles errores y, si es necesario, ajustar automáticamente el nivel de sensibilidad del sistema. Así, se podría mantener una mayor precisión y corregir errores sin necesidad de una revisión completa del sistema.

Sería beneficioso integrar el sistema de detección y análisis del flujo vehicular con otras tecnologías de gestión del tráfico, como sistemas de semáforos inteligentes o plataformas de gestión del transporte en tiempo real. Esta integración permitiría una mayor optimización del flujo vehicular, ajustando dinámicamente los semáforos o recomendando rutas alternativas a los conductores, contribuyendo así de forma más eficaz a la reducción de la congestión y a la mejora de la movilidad urbana.

En resumen, si bien el sistema desarrollado en esta investigación ha demostrado su eficacia para la detección, el conteo y la clasificación de vehículos, existen diversas áreas de mejora y expansión que podrían fortalecer aún más su impacto en la gestión del tráfico urbano. Las recomendaciones propuestas buscan optimizar la precisión, ampliar la integración con otras tecnologías de transporte y explorar nuevas aplicaciones predictivas que permitan una gestión del tráfico más eficiente. La implementación de estas mejoras no solo optimizaría el rendimiento del sistema, sino que también abriría la puerta a nuevas oportunidades de

investigación y desarrollo en el campo de la inteligencia artificial aplicada al transporte, contribuyendo significativamente a mejorar la movilidad y la sostenibilidad en las ciudades del futuro.

Bibliografía

- Bilbao, C. (2022). *Modelización de una red neuronal capaz de detectar emoticonos en imágenes*. Madrid.
- Agencia Europea de Medio Ambiente. (2019). *EEA Report*. Obtenido de <https://www.eea.europa.eu/es>
- Anaconda.com. (s.f.). *Anaconda*. Obtenido de <https://www.anaconda.com/download>
- Arevalo, V., Gonzáles, J., & Ambrosio, G. (2004). La librería de visión artificial opencv. aplicación a la docencia e investigación.
- Caceres, P. (2021). *Implementación de un sistema inteligente para la identificación vehicular*. Ambato-Ecuador.
- Caiza, L. J. (2019). *DESARROLLO E IMPLEMENTACIÓN DE UN ANALIZADOR DE TRÁFICO VEHICULAR BASADO EN LA OBTENCIÓN Y TRATAMIENTO DE LOS DATOS CONTENIDOS EN MAPAS GEOGRÁFICOS CON INFORMACIÓN DE FLUJO VEHICULAR GEORREFERENCIADOS*. Proyecto de titulación , Escuela Politécnica Nacional, Quito.
- Campos, J. (2011). *Una metodología para el análisis de la congestión del tráfico metropolitano*. Obtenido de Dialnet: <https://dialnet.unirioja.es/servlet/tesis?codigo=237950>
- Cascajares Ruperez, M. (2021). La aportación científica a la innovación tecnológica: análisis de estrategias de investigación en base de datos de referencia. *Dialnet*, 180. Obtenido de <http://hdl.handle.net/10835/11511>
- Challenger , I., Díaz, Y., & Becerra, R. (2014). El lenguaje de programación Python. *Ciencias Holguín*. Obtenido de <https://www.redalyc.org/pdf/1815/181531232001.pdf>
- Coanqui, F., Estofanero, R., & Mamani, H. (2022). *Aplicación de inteligencia y visión artificial para la obtención del aforo vehicular*. Juliaca.
- Coello , A. (2024). Prototipo de un sistema de semaforización inteligente para la optimización del tráfico vehicular empleando inteligencia artificial. *Universidad*

- Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera de Telecomunicaciones. Obtenido de <https://repositorio.uta.edu.ec/items/f53071d6-ee1c-4220-bf36-df90b53ee5a4>*
- Cueva, W., & Muñoz, F. (2017). *Analisis, monitoreo e implementacion de un algoritmo para el conteo del trafico vehicular aplicando vision artificial*. Cuenca.
- Cuzco Cualchi, A. J. (2020). *SISTEMA DE GESTIÓN Y AUTOMATIZACIÓN DEL ACCESO A UN PARQUEADERO*. Proyecto de titulacion, Universidad tecnica del Norte, Ibarra.
- De los Santos, P. (2016). *Numython*. Obtenido de <https://numython.github.io/posts/2016/01/introduccion-scikit-image-procesamiento/>
- Diaz, L. (2024). Propuesta de monitoreo en tiempo real para la gestión del tráfico vehicular en ciudades inteligentes con un enfoque experimental y exploratorio. *Universidad Tecnológica de Pereira*. Obtenido de <https://hdl.handle.net/11059/15521>.
- Falconí Punguil, D. G. (2021). *Algoritmos de Deep Learning utilizando Tensor Flow para el tratamiento de datos de produccion cientifica en la Universidad Tecnica de Cotopaxi*. Tesis de Maestria, Universidad Tecnica de Cotopaxi, Latacunga.
- García, J. (2023). Aplicaciones de los métodos de optimización cuasi-newtoniana en el aprendizaje profundo. *Instituto Tecnológico de Santo Domingo*.
- Guaman Tisalema, D. P. (2015). *Analisis, evaluacion y caracterizacion de imagenes termograficas utilizando redes neuronales artificiales y tecnicas de extraccion de caracteristicas de textura de haralick*. Universidad Tecnica de Ambato, Ambato.
- Guevara, R., Diaz, M., & Torres, M. (2024). Modelo Basado en YOLOv8 para la Detección de Residuos Sólidos. *Innovación y Software*. Obtenido de <https://revistas.ulasalle.edu.pe/innosoft/article/view/160/252>
- Instituto Nacional de Estadísticas y Censos. (2024). *Estadísticas de transporte (ESTRA)*. Obtenido de https://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Economicas/Estadistica%20de%20Transporte/veh_matriculados/2023/2023_Resultados_Vehiculos.pdf

- Jarrin, D., & Vinueza, D. (2020). *Desarrollo de un sistema prototipo de pago de frutas a traves del entrenamiento de redes neuronales artificiales convolucionales*. Escuela Politecnica Nacional.
- Llerena, J. P. (2019). *Deteccion de vehiculos y flujo de trafico con redes neuronales convolucionales en vision artificial*. Tesis de Maestria.
- Loaiza Quintana, A. F., Manzano Herrera, D. A., & Munera Salazar, L. E. (2012). Sistema de vision artificial para conteo de objetos en movimiento. *El Hombre y la Maquina*(40), 87-101.
- Maeda, V., Guerrero, C., Olvera, C., Araiza, M., Espinoza, G., & Bordón, R. (2018). Redes neuronales convolucionales para la detección y clasificación de enfermedades de plantas basadas en imágenes digitales. *Universidad Autonoma de Zacatecas*.
- Matich, D. J. (2001). *Redes neuronales: Conceptos Basicos y Aplicaciones*.
- Mazzola, M. (2021). Aplicación de la Inteligencia Artificial Geoespacial a la epidemiología medio ambiental. Obtenido de <https://hdl.handle.net/10630/20719>
- Ministerio de Transporte y Obras Públicas. (2023). *Politica nacional de movilidad urbana sostenible del Ecuador*. Obtenido de <https://www.obraspublicas.gob.ec/wp-content/uploads/downloads/2023/10/Politica-Nacional-de-Movilidad-Urbana-Sostenible-del-Ecuador-2023.pdf>
- Norma tecnica ecuatoriana INEN 2656. (09 de 2016). *Clasificacion vehicular*. Obtenido de file:///D:/Downloads/toaz.info-clasificacion-vehicular-ecuador-nte-inen-2656-1-pr_10164352686ddcb70b65fd8f2afd1da8.pdf
- NTE INEN 2207. (2016). *Gestion Ambiental.Aire.Vehiculos Automotores .Limites permitidos de emisiones producidas por fuentes móviles terrestres de Diesel*. Obtenido de <https://www.aeade.net/wp-content/uploads/2016/12/2207-1.pdf>
- OpenCV.org. (2016). *OpenCV*. Obtenido de <https://opencv.org/>

- Ortiz, A., & Flores, M. (2024). Evaluación y aplicación de diferentes estructuras de redes neuronales para grafos : implementación de ensambles de modelos GNN. Obtenido de <http://bibdigital.epn.edu.ec/handle/15000/26018>
- Picazo Montoya, O. (2018). *Redes neuronales convolucionales profundas para el reconocimiento de emociones en imagenes*. Tesis de Maestria, Universidad politecnica de Madrid.
- Pino, R., Gómez, A., & de abajo Martínez, N. (2001). Introducción a la inteligencia artificial: sistemas expertos, redes neuronales artificiales y computación evolutiva.
- python.org. (2016). *Python*. Obtenido de <https://www.python.org/>
- Pytorch.org. (2015). *Pytorch*. Obtenido de https://pytorch.org/tutorials/beginner/introyt/introyt1_tutorial.html
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified,Real-Time Object Detection. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 779-788. doi:10.1109/CVPR.2016.91.
- Rivera, I., & Zambrano, D. (2022). *Implementacion de Reconocimiento Facial y Vision artificial en Robot Nao con Python y OpenCV*. Guayaquil.
- Rodríguez, M., & Pérez, E. (2024). Integración de la inteligencia artificial en el diagnóstico y pronóstico del cáncer de mama en México. *nstituto Tecnológico de Durango/TecNM*, 8(1), 3358-3377.
- Rodríguez, R. (2004). Actores sintéticos en tiempo real: Nuevas estructuras de datos y métodos para su integración en aplicaciones de simulación. En R. Rodríguez, *Actores sintéticos en tiempo real: Nuevas estructuras de datos y métodos para su integración en aplicaciones de simulación*.
- Rosero, A. (2019). Cuantificación automatizada de factores de emisión vehicular: desarrollo de una metodología híbrida de visión artificial y métodos tradicionales. Obtenido de <http://hdl.handle.net/1992/39342>
- TensorFlow.org. (2015). *TensorFlow*. Obtenido de <https://www.tensorflow.org/?hl=es-419>

- Terrones Rodriguez, A. L., & Rocha Bernardi, M. (2024). EL VALOR DE LA ÉTICA APLICADA EN LOS ESTUDIOS DE INGENIERÍA EN UN HORIZONTE DE INTELIGENCIA ARTIFICIAL CONFIABLE. *Sophia, Colección de Filosofía de la Educación*(36), 221-245. Obtenido de <https://doi.org/10.17163/soph.n36.2024.07>
- Urrego, G. E., Calderón, F. C., Forero, A., & Quiroga, J. A. (2009). Adquisición de variables de trafico vehicular usando visión por computador. *Revista de Ingeniería*(30), 07-15. Obtenido de http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0121-49932009000200002&lng=en&tlng=es.
- Zurita, L., Quintero, Y., & Cortijo, R. (2024). Diseño e implementación de un sistema de monitoreo de tráfico en tiempo real que utilice visión artificial y sensores IoT. Obtenido de <http://repositorio.uisrael.edu.ec/handle/47000/4257>

Anexos

Anexo A: Código en Python de la base de datos

```
import cv2 # Librería de OpenCV para procesamiento de imágenes
import os  # Librería para manejo de archivos y directorios

# Cargar el clasificador Haarcascade para detección de vehículos
vehiculos_classifier =
cv2.CascadeClassifier("C:/Users/ac607/Desktop/BASEDEDATOS/cars.xml")

# Ruta del video
Ruta_video = "C:/Users/ac607/Desktop/BASEDEDATOS/AUTOPISTA2.MOV"
Direccion_video = os.path.dirname(Ruta_video)

# Abrir el video
cap = cv2.VideoCapture(Ruta_video)

# Obtener los FPS originales del video
fps = cap.get(cv2.CAP_PROP_FPS)
if fps == 0: # Evitar división por cero en caso de error al obtener FPS
    fps = 30 # Valor por defecto

frame_time = int(1000 / fps) # Convertir FPS a milisegundos por frame

# Reducir resolución si el video es muy grande
frame_width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH) * 0.5) # Reducir al 50%
frame_height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT) * 0.5)

# Crear el sustractor de fondo (MOG2) con sombras desactivadas para mayor
velocidad
bg_subtractor = cv2.createBackgroundSubtractorMOG2(history=500,
varThreshold=50, detectShadows=False)

# Directorio para almacenar imágenes detectadas en el video Hernan Malo y
panamericana
directorio_deteccion = os.path.join(Direccion_video, 'Deteccion_vehiculos3')
if not os.path.exists(directorio_deteccion):
    os.makedirs(directorio_deteccion)

x_contador = 0 # Inicializar contador de imágenes
frame_skip = 2 # Procesar todos los fotogramas (puedes cambiarlo a 2 o más
si va muy lento)
frame_count = 0 # Contador de fotogramas

# Procesar el video frame a frame
```

```

while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break

    frame_count += 1
    if frame_count % frame_skip != 0:
        continue # Saltar este fotograma para mejorar velocidad

    # Reducir el tamaño del frame para más velocidad
    frame = cv2.resize(frame, (frame_width, frame_height))

    # Convertir a escala de grises
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Aplicar sustracción de fondo
    fg_mask = bg_subtractor.apply(frame)

    # Operaciones morfológicas
    fg_mask = cv2.morphologyEx(fg_mask, cv2.MORPH_OPEN,
cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3, 3)))

    # Encontrar contornos
    contours, _ = cv2.findContours(fg_mask, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)

    # Procesar cada contorno detectado
    for contour in contours:
        x, y, w, h = cv2.boundingRect(contour)

        # Filtrar las detecciones pequeñas
        if w > 100 and h > 100:
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 255),
2) # Se dibuja el rectángulo
            # Recortar la imagen del vehículo detectado
            Imagen_vehiculo = frame[y:y+h, x:x+w]
            # Guardar la imagen del vehículo
            cv2.imwrite(os.path.join(directorio_deteccion,
f"vehiculo_{x_contador}.jpg"), Imagen_vehiculo)
            x_contador += 1

    # Mostrar las imágenes procesadas
    cv2.imshow('Sustracción de fondo', fg_mask)
    cv2.imshow('Detección de vehículos', frame)

```

```

        # Sincronizar con la velocidad del video
        if cv2.waitKey(frame_time) & 0xFF == ord('q'): # Espera el tiempo
            exacto según FPS
            break

# Liberar recursos
cap.release()
cv2.destroyAllWindows()

```

Anexo B: Código en Python del archivo yaml

```

import yaml
dataset_yaml = {
    'path': '/content/sample_data/dataset7', # Carpeta de imágenes
    'train': 'images/train',                # Subcarpeta de entrenamiento
    'val': 'images/val',                    # Subcarpeta de validación
    'test': 'images/test',                  # Subcarpeta de prueba
    'names': ['Motos', 'Vehiculos livianos', 'Vehiculos pesados'] #clases de vehiculos
}
# Ruta donde se guardará el archivo dataset7.yaml
yaml_archivo = '/content/sample_data/dataset7/dataset8.yaml'
with open(yaml_archivo, 'w') as file:
    yaml.dump(dataset_yaml, file, default_flow_style=False)

print(f"Archivo YAML: {yaml_archivo}")

```

Anexo C: Código en Python del entrenamiento del modelo

```

from ultralytics import YOLO

model = YOLO('yolov8n.pt')

# Entrenar la red
model.train(
    data='/content/sample_data/data.yaml',
    epochs=30,
    imgsz=640,
    batch=16,
    device=0 # GPU
)

```

Anexo D: Código en Python de la detección de vehículos

```
import cv2 # Libreria para el procesamiento de imagenes ( Vision artificial)
import numpy as np #Libreria para operaciones matematicas
import math #Libreria para operaciones matematicas avanzadas
from ultralytics import YOLO # Importacion del modelo YOLO
from config import clases #Class de YOLO
import torch # Libreria para aprendizaje profundo
from seguimiento import Seguimientodeobjetos # Seguimiento de objetos

model = YOLO("C:/Users/ac607/Desktop/BASEDEDATOS/MODELO2vn.pt") # Modelo
entrenado con YOLOv8n
cap =
cv2.VideoCapture("C:/Users/ac607/Desktop/BASEDEDATOS/AUTOPISTA2.mov") #Ruta
del video Hernan Malo y Panamericana
if not cap.isOpened(): #Se verifica si la ruta del video es correcta
    print("Error al abrir el video.")
    exit()

frame_width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH)) #Dimensiones del video
original
frame_height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT)) #Dimensiones del
video original
mask_polygon = np.array([(869, 591), (1010, 622), (874, 773), (918, 854),
(546, 824), (224, 705), (483, 689)]), dtype=np.int32) # Zona de mascara en el
video con coordenadas dadas
brown_line = [(949, 677), (732, 629)] #Linea de conteo con coordenadas dadas
y1_line = brown_line[0][1] #Coordenadas Y de linea de conteo

tracker = Seguimientodeobjetos() #Inicializar el objeto de seguimiento
contador_total = 0 #Variable de conteo total de vehiculos
ids_contados = set() #Variable para impedir que se cuente un vehiculo varias
veces
contador_clases = {"vehiculos livianos": 0, "vehiculos pesados": 0, "motos":
0} #Variable con el conteo de cada clase de vehiculo

#Procesamiento para cada frame del video
while True:
    ret, frame = cap.read() #Se lee un frame del video
    if not ret:
        break #Si no existe mas frames , se sale del bucle

    mask = np.zeros_like(frame) # Creacion de mascara con el tamaño del
frame
```

```

        cv2.fillPoly(mask, mask_polygon, (255, 255, 255)) #Se dibuja el poligono
en la mascara
        masked_frame = cv2.bitwise_and(frame, mask) #Se aplica la mascara al
frame para que se oculten zonas no deseadas
        with torch.no_grad(): #Mejora el rendimiento del video
            results = next(model.predict(source=masked_frame, imsz=640,
conf=0.4, stream=True)) #Realiza predicciones con el modelo en base a nivel
de confianza.

        objetos = [] #Variable de objetos detectados con las coordenadas
        clases_detectadas = [] #Variable de las clases detectadas
        for box in results boxes: #Procesar cada deteccion del modelo
            cls_id = int(box.cls[0]) if box.cls is not None else -1 #Se obtiene
el ID por cada clase
            label_in = model.names[cls_id] if cls_id >= 0 and cls_id <
len(model.names) else "Desconocido"
            if label_in in ['car', 'truck', 'person', 'motorcycle']: #Solo se
detecta vehiculos liviano-pesado y motos
                x1, y1, x2, y2 = map(int, box.xyxy[0].tolist()) #Se obtiene
coordenadas de objeto detectado
                w, h = x2 - x1, y2 - y1 #Se calcula el ancho y alto del objeto
detectado
                objetos.append([x1, y1, w, h]) #Guarda las coordenadas de
objetos detectados
                clases_detectadas.append(cls_id)#Guarda el ID de clase
detectada

        objetos_rastreados = tracker.seguir_objetos(objetos) #Se realiza un
seguimiento de la clase detectada
        for i, obj in enumerate(objetos_rastreados): #Se dibuja los objetos
detectados y se realiza el conteo vehicular
            x, y, w, h, obj_id = obj #Se extraen características importantes del
objeto detectado
            cx = x + w // 2 #Se calcula el centro en X
            cy = y + h // 2 #Se calcula el centro en Y

            cls_id = clases_detectadas[i] if i < len(clases_detectadas) else -1
            label_in = model.names[cls_id] if cls_id >=0 and cls_id <
len(model.names) else "Desconocido"
            label = clases.get(label_in, "Desconocido") #Se obtiene el nombre de
la clase detectada

            if obj_id not in ids_contados: #Se hace una verificacion, es decir
si el objeto ya fue contado

```

```

        if obj_id in tracker.prev_positions: #Se verifica si el objeto
tiene una posicion anterior registrada
            prev_cx, prev_cy = tracker.prev_positions[obj_id] #Se
obtiene una posicion anterior
            if (prev_cy < y1_line and cy >= y1_line) or (prev_cy >
y1_line and cy <= y1_line):
                contador_total += 1 #Se aumenta el conteo total
                ids_contados.add(obj_id) #Se marca al objeto como
contado
            if label in contador_clases: #Se va aumentando el conteo
vehicular segun la clase
                contador_clases[label] += 1

        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 165, 255), 2) #Se
dibuja el rectangulo alrededor del objeto detectado en color naranja
        cv2.putText(frame, f'{label}', (x, y - 10), #Se dibuja la etiqueta
del objeto detectado
                    cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 0), 1)

        cv2.line(frame, brown_line[0], brown_line[1], (19, 69, 139), 2) #Se
dibuja la línea de conteo en color marron
        cv2.putText(frame, f'Total: {contador_total}', (30, 50), # Mostrar el
conteo total de vehiculos en la pantalla en color verde
                    cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
        cv2.putText(frame, f'Livianos: {contador_clases["vehiculos livianos"]}',
(30, 90), #Mostrar el conteo total de vehiculos livianos en color blanco
                    cv2.FONT_HERSHEY_SIMPLEX, 0.8, (255, 255, 255), 2)
        cv2.putText(frame, f'Motos: {contador_clases["motos"]}', (30,
130), #Mostrar el conteo total de motos en color blanco
                    cv2.FONT_HERSHEY_SIMPLEX, 0.8, (255, 255, 255), 2)
        cv2.putText(frame, f'Pesados: {contador_clases["vehiculos pesados"]}',
(30, 170), #Mostrar el conteo total de vehiculos pesados en color blanco
                    cv2.FONT_HERSHEY_SIMPLEX, 0.8, (255, 255, 255), 2)

        # Mostrar frame
        cv2.imshow("Detección, conteo y clasificacion vehicular", frame) #Se
muestra en pantalla la detecccion, conteo y clasificacion vehicular
        if cv2.waitKey(5) & 0xFF == ord('q'): #Se sale del bucle si se presiona
la letra "q"
            break

cap.release()
cv2.destroyAllWindows() #Cerrar ventanas

```