Escuela Superior Politécnica del Litoral

Facultad de Ingeniería en Electricidad y Computación

Sistema inteligente para la automatización del rastreo vehicular por medio de múltiples cámaras

TECH-421

Proyecto Integrador

Previo la obtención del Título de:

Ingeniero en Ciencias de la Computación

Presentado por:

FERNANDEZ BUSTAMANTE DANIEL FRANCISCO

JARAMILLO ARANA JOSÉ GABRIEL

Guayaquil - Ecuador

Año: 2025

Dedicatoria

El presente trabajo y obtención del título se lo dedico a mi padre celestial Jehová por brindarme siempre su espíritu santo a fin de poder cumplir con cada asignación propuesta y encomendada. A mis padres, hermanos y familia muy cercana por siempre estar pendientes del progreso universitario. A mi esposa por estar en todas las actividades presente y por no dejarme en esas madrugadas llenas de tareas y proyectos. A los compañeros de vivencia universitaria que sin duda alguna enriquecieron el aprendizaje.

José Gabriel Jaramillo Arana

Dedicatoria

La elaboración de este trabajo, así como el esfuerzo y la perseverancia que lo sustentan, se lo dedico en primer lugar a Dios, por su infinita gracia, bondad y misericordia, que han sido la luz y la fortaleza en cada paso de este camino. Dedico también este logro a mis padres, quienes, con su amor incondicional, sacrificio y ejemplo de vida han sido un pilar esencial en mi formación personal y profesional. A mis hermanas y a toda mi familia, cuyo apoyo constante me han impulsado a no rendirme frente a las adversidades. Finalmente, a mis amistades más cercanas, que, con su compañía sincera, consejos y ánimo en los momentos difíciles, han contribuido a que este proyecto se convierta en una meta alcanzada y en una nueva motivación para seguir adelante.

Daniel Francisco Fernández Bustamante

Agradecimientos

En primer lugar, le agradezco a Jehová nuestro Dios por brindarme el conocimiento, la paciencia y el aguante a lo largo de toda la carrera. A mis padres y hermanos les expreso una gratitud enorme por instruirme y guiarme para seguir progresando a nivel personal. A mi esposa y su familia por saber darme el ánimo necesario para no abandonar mis ideales. A mi compañero de tesis, Daniel y a todos los compañeros, profesores, consejeros y autoridades que hicieron de esta formación académica algo grato de recordar día a día.

José Gabriel Jaramillo Arana

Agradecimientos

Agradezco a Dios por concederme fortaleza y guía a mis pasos con su gracia y misericordia. A mis padres por su amor incondicional, paciencia y entrega. A mis hermanas, por su cariño y apoyo constante. Extiendo mi gratitud a mis amigos Isaac, Leonardo, Nicol, Aaron y Génesis, quienes, con su apoyo inquebrantable, me motivaron a dar siempre lo mejor de mí, recordándome que nunca debía rendirme. A mi compañero José, por su esfuerzo, compromiso y compañerismo y a todos aquellos colegas y compañeros con los que tuve la oportunidad de compartir a lo largo de este camino, ya que cada uno, con su aporte, dejó huellas que hicieron posible llegar a este día.

Daniel Francisco Fernández Bustamante

Declaración Expresa

Nosotros Daniel Francisco Fernández Bustamante y José Gabriel Jaramillo Arana acordamos y reconocemos que:

La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores.

La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por mí/nosotros durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que nos corresponda de los beneficios económicos que la ESPOL reciba por la explotación de nuestra innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique los autores que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Guayaquil, 30 de mayo del 2025.

Firmado electrónicamente por:
DANIEL FRANCISCO FERNANDEZ BUSTAMANTE
Validar únicamente con FirmaEC

Daniel Francisco Fernández Bustamante

Firmado electrónicamente por:
JOSE GABRIEL JARAMILLO ARANA
Validar únicamente con FirmaEC

José Gabriel Jaramillo Arana

Evaluadores

Firmado electrónicamente por:
FEDERICO XAVIER DOMINGUEZ BONINI
Validar únicamente con FirmaEC

Federico Xavier Domínguez Bonini

Profesor de Materia

Firmado electrónicamente por:
MIGUEL ANDRES REALPE ROBALINO
Validar únicamente con FirmaEC

Miguel Andrés Realpe Robalino

Tutor de proyecto

Resumen

Este proyecto implementa un sistema automatizado de rastreo vehicular mediante técnicas de Deep Learning y visión por computadora. El sistema utiliza YOLOv8 para detectar vehículos, ByteTrack para seguimiento intra-cámara y Vision Transformer (ViT) para generar embeddings robustos que permiten la Reidentificación multicámara. Se procesaron secuencias de video de múltiples cámaras, aplicando técnicas de matching por similitud coseno y re-ranking para mejorar la precisión. Los resultados demostraron un 94.7% de precisión Rank-1 y 77.96% mAP, validando la efectividad del enfoque propuesto. El sistema ofrece una solución escalable para el monitoreo vehicular automatizado en entornos urbanos complejos.

Palabras Clave: Rastreo vehicular, Visión artificial, Deep Learning, Reidentificación, Procesamiento multicámara.

Abstract

This project implements an automated vehicle tracking system using Deep Learning and computer vision techniques. The system employs YOLOv8 for vehicle detection, ByteTrack for intra-camera tracking, and Vision Transformer (ViT) to generate robust embeddings enabling multi-camera ReIDentification. Multiple camera video sequences were processed, applying cosine similarity matching and re-ranking techniques to improve accuracy. Results demonstrated 94.7% Rank-1 accuracy and 77.96% mAP, validating the effectiveness of the proposed approach. The system provides a scalable solution for automated vehicle monitoring in complex urban environments.

Keywords: *Vehicle tracking, Computer vision, Deep Learning, ReIDentification, Multi-camera processing.*

Índice general

Abreviatura	
ALPR	Automatic License Plate Recognition (Reconocimiento Automático de Matrículas)
BNNeck	Batch Normalization Neck (Cuello de Normalización por Lotes)
CNN	Convolutional Neural Network (Red Neuronal Convolucional)
COCO	Common Objects in Context (Objetos Comunes en Contexto)
ECU911	Servicio Integrado de Seguridad ECU 911
ESPOL	Escuela Superior Politécnica del Litoral
FFN	Feed-Forward Network (Red de Alimentación hacia Adelante)
GPU	Graphics Processing Unit (Unidad de Procesamiento Gráfico)
HA-CNN	Harmonious Attention CNN (Red de Atención Armónica)
ID	Identifier (Identificador)
IDF1	Identification F1 Score (Puntuación F1 de Identificación)
IoU	Intersection over Union (Intersección sobre Unión)
mAP	mean Average Precision (Precisión Promedio Media)
MGN	Multiple Granularity Network (Red de Múltiples Granularidades)
MHSA	Multi-Head Self-Attention (Auto-Atención Multi-Cabezal)

MOT	Multi-Object Tracking (Seguimiento Multi-Objeto)
OCR	Optical Character Recognition (Reconocimiento Óptico de Caracteres)
OC-SORT	Observation-Centric SORT (SORT Centrado en Observaciones)
OSNet	Omni-Scale Network (Red Omni-Escala)
OTRI	Oficina de Transferencia de Resultados de Investigación
PCB	Part-based Convolutional Baseline (Línea Base Convolucional Basada en Partes)
PK	P identidades, K instancias (muestreo por identidad)
ReID	Re-Identification (Reidentificación)
R-CNN	Region-based Convolutional Neural Network (Red Neuronal Convolucional Basada en Regiones)
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional
RNN	Recurrent Neural Network (Red Neuronal Recurrente)
ROI	Region of Interest (Región de Interés)
RPN	Region Proposal Network (Red de Propuesta de Regiones)

SORT	Simple Online and Realtime Tracking (Seguimiento Simple en Tiempo Real)
SSD	Single Shot Detector (Detector de Una Sola Pasada)
TTA	Test-Time Augmentation (Aumento en Tiempo de Prueba)
UMAP	Uniform Manifold Approximation and Projection (Aproximación y Proyección de Variedades Uniformes)
ViT	Vision Transformer (Transformador de Visión)
YOLO	You Only Look Once (Solo Miras Una Vez)

Índice de Figuras

Índice de Tablas

Capítulo 1

1.1 Introducción

En Ecuador, la implementación de sistemas tecnológicos para el monitoreo y control de tráfico vehicular ha cobrado relevancia en el marco de una gestión urbana más segura y eficiente. El crecimiento exponencial de la tecnología y la inteligencia artificial ofrecen la posibilidad de mantener un registro detallado de los vehículos que transitan o ingresan áreas específicas que sean monitoreadas por cámaras. De esta manera se mejora la capacidad de las entidades públicas y privadas para gestionar el flujo vehicular, optimizando la vigilancia en zonas de alto tránsito o de acceso restringido. Pese a que la implementación de cámaras de seguridad en las ciudades latinoamericanas aún es limitada en comparación con otras regiones, lo cual resalta la necesidad de aprovechar al máximo las tecnologías actuales para alcanzar un control vehicular.

Para el desarrollo de este proyecto, se utilizarán imágenes obtenidas de un dataset público, con el fin de simular un entorno realista en el que operar. Dado que no se cuenta con una entidad que proporcione imágenes en tiempo real o de manera controlada, el uso de datasets abiertos permitirá entrenar y evaluar el sistema de manera efectiva. El objetivo central de este trabajo es implementar un modelo capaz de identificar vehículos basados en la ubicación y Reidentificación del objetivo mediante diversas cámaras.

Esta investigación tiene como objetivo no solo desarrollar un sistema técnico, sino también fortalecer las capacidades de las autoridades competentes y mejorar la seguridad pública en el país. Busca proporcionar una herramienta que permita rastrear vehículos tanto en entornos privados como públicos, garantizando el uso adecuado de tecnologías emergentes, asegurando una correcta utilización de las nuevas tecnologías emergentes implementando inteligencia artificial en los ámbitos cotidianos para mejorar la calidad de esta.

1.2 Descripción del problema

En varios países de Latinoamérica la necesidad de mejorar el control y monitoreo vehicular plantea desafíos significativos para garantizar una gestión de tránsito más eficiente y segura en áreas urbanas y zonas privadas. En 2022 la empresa Telconet implementó cámaras de seguridad con el fin de contemplar diversas problemáticas locales como el robo de vehículos y la omisión de las señales de tránsito. Este proyecto ambicioso luego de 3 años aún enfrenta dificultades en su operatividad debido a sus falsos positivos en el reconocimiento y a la saturación del monitoreo de cámaras que ahora se encarga el ECU911 de las instalaciones de Samborondón en revisar, controlar y gestionar las alertas recibidas en Samborondón y Guayaquil [1]. Estas limitaciones son la oportunidad perfecta para implementar soluciones avanzadas que permitan automatizar el proceso de identificación y rastreo vehicular.

El objetivo del proyecto es diseñar un sistema inteligente de automatización capaz de rastrear vehículos utilizando múltiples cámaras, con la capacidad de identificar características específicas de los automóviles, tales como marca, modelo, color y de ser necesario, matrícula para un mayor control. Se requiere considerar:

- Automatizar el rastreo vehicular.
- Integrar múltiples cámaras de vigilancia.

1.3 Justificación del problema

La necesidad de contar con sistemas avanzados para el monitoreo vehicular en tiempo real es importante para la gestión del tránsito y la garantía de seguridad en zonas de acceso restringido y áreas urbanas de alta densidad vehicular. Un sistema de monitoreo automatizado no solo contribuiría a mejorar la eficiencia en la supervisión del tráfico, sino también a optimizar los procesos de control en sitios sensibles, como accesos a instalaciones privadas.

En [2] mediante un video se muestra que el monitoreo de las cámaras de vigilancia en el ECU911 se hace de forma manual lo cual incrementa sesgos humanos en cuanto a precisión y tiempo de respuesta. Un sistema automatizado reduciría considerablemente el tiempo de respuesta de autoridades aumentando la tasa de recuperación de autos robados. Este tipo de tecnología es especialmente relevante en el contexto latinoamericano, donde las redes de cámaras de vigilancia están en crecimiento, pero aún no se han optimizado en su utilización para aplicaciones de control vehicular automatizado.

1.4 Objetivos

1.4.1 Objetivo general

Diseñar e implementar un sistema inteligente de rastreo vehicular automatizado mediante el uso de técnicas Deep Learning de Reidentificación a partir de imágenes de múltiples cámaras de vigilancia.

1.4.2 Objetivos específicos

1. Desarrollar un algoritmo capaz de detectar automáticamente vehículos en imágenes de cámaras de seguridad para su posterior seguimiento visual.
2. Desarrollar un codificador de imágenes que extraiga las características principales de un vehículo para su Reidentificación.
3. Crear un sistema que permita identificar y agrupar vehículos con características similares a partir de imágenes.

1.5 Marco teórico

La automatización del rastreo vehicular ha evolucionado junto con la tecnología y la inteligencia artificial. Dichas herramientas son determinantes en contextos como la seguridad, gestión del tráfico, logística y transporte, y en casos extremos, la asistencia para emergencias.

Por lo que, en esta sección se presentarán conceptos relevantes para entender la solución propuesta.

1.5.1 Algoritmos de detección de objetos

La detección de objetos es una tecnología relacionada con la visión artificial y el procesamiento de imágenes que trata de detectar casos de objetos semánticos de una cierta clase (como humanos, edificios, o vehículos) en videos e imágenes digitales [3].Tal es el caso del algoritmo YOLO (You Only Look Once) el cuál usa una sola red neuronal convolucional para identificar y clasificar objetos. A diferencia de otros algoritmos que necesitan más etapas de procesamiento y por ende mayor tiempo de ejecución, YOLO ofrece una detección de objetos bastante eficiente en un corto tiempo lo que eleva su uso hasta las aplicaciones en tiempo real [4].

Otro de los algoritmos funcionales para detección de objetos se llama SSD (Single Shot Detector) que permite identificar objetos más pequeños en una agrupación, pues, el tamaño de sus cajas delimitadoras de sección varía con relación al objetivo. SSD utiliza una serie de convoluciones para que la extracción de características sea precisa y rápida [5].

Un algoritmo con una funcionalidad especializada en la detección de objetos de manera más precisa, pero más lenta es Faster R-CNN, que consiste en que el algoritmo propone regiones para hipotetizar la ubicación de los objetos [6]. Una implementación que incluye un bajo costo a la hora de proponer regiones es utilizar RPN que es una red totalmente convolucional encargada de predecir simultáneamente los límites de los objetos y las puntuaciones de objetividad en cada posición.

Por otro lado, RetinaNet toma un enfoque que se centra en la implementación de la pérdida focal, una técnica que mejora la detección de objetos pequeños y difíciles de identificar. Está diseñado para manejar datasets desbalanceados.

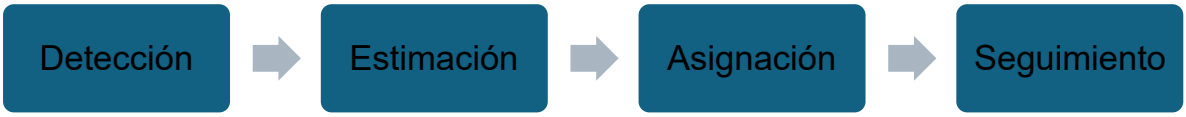
1.5.2 Algoritmos de seguimiento

SORT (Simple Online and Realtime tracking) es un algoritmo de seguimiento diseñado para predecir el movimiento de un objeto en tiempo real. Para su funcionamiento, utiliza el filtro de kalman que le permite predecir el siguiente cuadro en que se moverá el objeto según su trayectoria. Además, implementa el algoritmo húngaro el cual es una abstracción de la asignación de costo mínimo que sirve para sintetizar la distancia mínima [7].

Su estructura se basa en el análisis de 4 etapas:

Figura 1.1

Secuencia de Algoritmo Sort



En cuanto a la etapa de seguimiento (tracking), SORT crea y destruye las cajas delimitadoras de sección a medida que cambia la posición del objeto. Cuando el objetivo seleccionado cambia su apariencia debido a situaciones externas, se da origen al ReID, lo que permite reconocer el objeto en otro momento y en otras circunstancias.

DeepSORT es una extensión del algoritmo SORT que incorpora un módulo de apariencia basado en redes neuronales [8], además utiliza un filtro de Kalman, que es un algoritmo recursivo que predice la posición y velocidad de un objeto utilizando un modelo de movimiento y luego ajusta esa predicción con cada nueva medición. Esto es empleado para

suavizar trayectorias y mantener el tracking incluso cuando dos vehículos se cruzan o quedan parcialmente ocultos [9].

ByteTrack es un algoritmo de seguimiento que aprovecha todas las detecciones, incluyendo aquellas de baja confianza, para mejorar la asociación de tracks entre frames. Utiliza un mecanismo de asociación simple pero efectivo basado en similitud de movimiento y apariencia, logrando alto rendimiento en escenarios con oclusiones y movimientos rápidos.

Esta estrategia de "no descartar detecciones" permite a ByteTrack mantener una alta tasa de recuperación de identidades (ID) y reducir significativamente los fragmentos en las trayectorias. Su simplicidad y efectividad lo han posicionado como un referente en seguimiento multi-objeto [10].

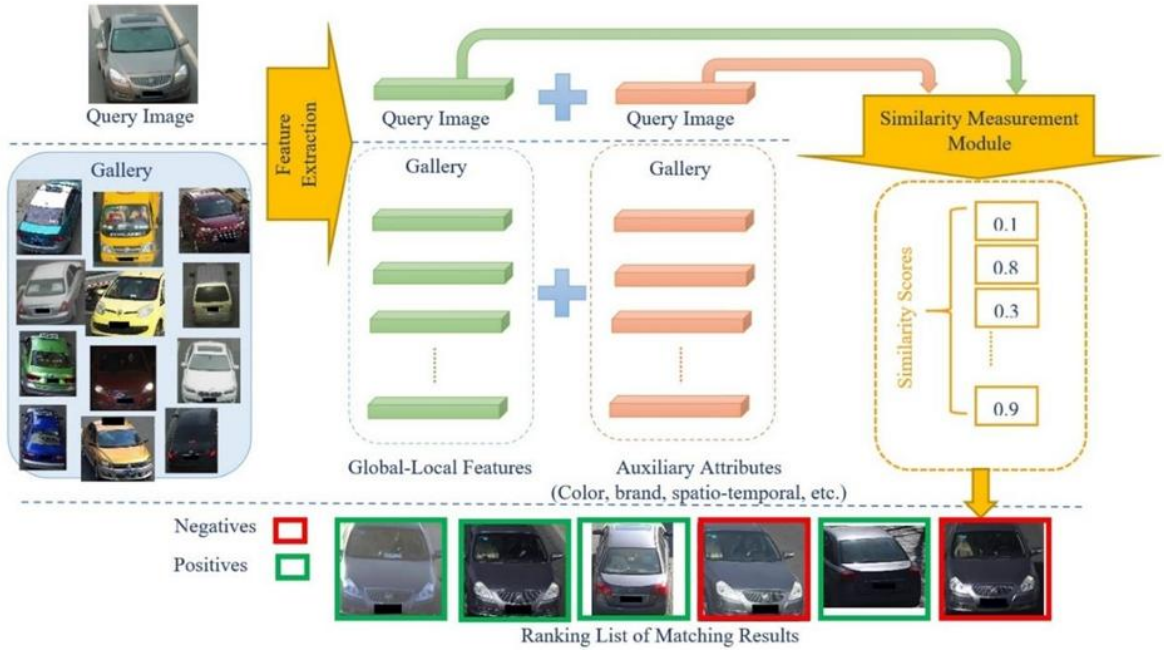
1.5.3 Reidentificación

Con el avance rápido en técnicas de aprendizaje profundo, muchos investigadores centran su atención en la Reidentificación de objetos también conocida como ReID, tiene como tarea reconocer y emparejar instancias del mismo objeto. Es decir, reubica un objeto cuando cambian las circunstancias de este. Tales son: posición, exposición a la luz y similitud. Estos factores son claves y determinantes a la hora de visualizar de forma automática un objeto [11].

Reconocer los atributos es una información importante para la tarea de ReID ya que dos imágenes de un vehículo pueden compartir el mismo modelo y color, pero sin necesidad de hacer referencia al mismo. Para cumplir con esta asignación, se usa la extracción de características globales (forma, tamaño) y auxiliares (color, posición espacio temporal) que dan lugar a un vector de características usado para comparar con las bases de datos de vehículos.

Figura 1.2

Ejemplo extracción de características



Nota: Tomada de [11]

Los modelos y algoritmos de ReID nacen como una necesidad técnica y práctica para reconocer objetos en entornos complejos, y se construyen sobre avances en redes neuronales, visión por computadora y aprendizaje de métricas [12].

Modelos basados en redes convolucionales (CNN)

Son modelos de aprendizaje profundo que procesan imágenes a través de filtros llamados convoluciones, capaces de detectar patrones como bordes, texturas y formas. Se usan para extraer características visuales (por ejemplo, color del vehículo, forma, matrícula, ropa de una persona) y convertir la imagen en un vector (embedding) que representa su identidad visual [13].

Tabla 1.1

Modelos basados en redes convolucionales

Modelo	Descripción
ResNet-50/101	Red de uso común. Usado como extractor de características.

PCB (Part-based Convolutional Baseline)	Divide la imagen en partes (por ejemplo, cabeza, torso, ruedas) para aprender características más detalladas.
MGN (Multiple Granularity Network)	Extrae características en diferentes escalas (global y local).
HA-CNN (Harmonious Attention CNN)	Introduce atención para enfocar regiones discriminativas.
OSNet	Compacto y eficaz, con convoluciones selectivas para extraer patrones omniescalares.

Modelos basados en Transformers

Reemplazan o complementan a las CNNs para entender la imagen como un conjunto de regiones interconectadas, lo que ayuda a captar detalles importantes, aunque estén lejos entre sí (como placa y forma del vehículo) [14].

Tabla 1.2

Modelos basados en Transformers

Modelo	Descripción
TransReID	Modelo Transformer con módulos de alineación de vista y atención para datos de vehículos/personas.
ViT (Vision Transformer)	Aplicado directamente a imágenes para aprender relaciones globales sin convoluciones.
TLP-ReID	Modelo Transformer con fusión de información local y global para mejorar el matching.

Algoritmos de aprendizaje de métricas

Estos algoritmos entrenan modelos para que aprendan una distancia adecuada entre objetos: los que son iguales deben estar cerca en el espacio de características, y los diferentes, lejos. Se aplican para que el sistema pueda comparar imágenes: si dos embeddings están cerca, se trata del mismo objeto (o persona/vehículo); si están lejos, no [15].

Tabla 1.3

Algoritmos de aprendizaje de métricas

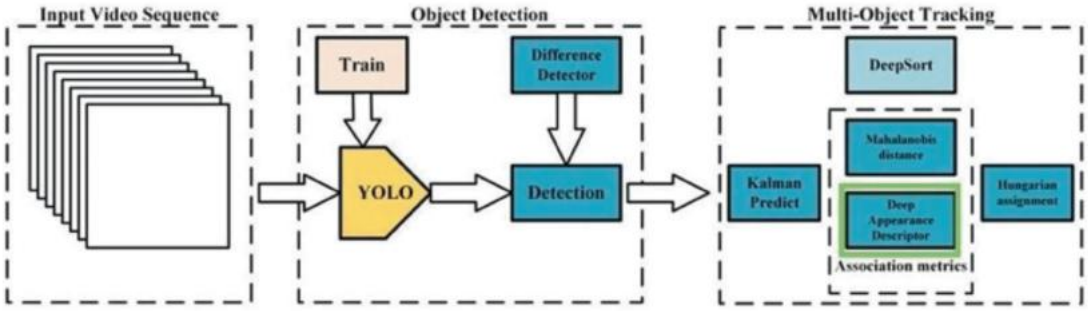
Algoritmo	Uso
Triplet Loss	Se entrena con tríos: ancla, positivo, negativo.
Contrastive Loss	Se entrena con pares de imágenes (similares/diferentes).
Center Loss	Reduce la variación intra-clase durante el entrenamiento.
Circle Loss / ArcFace	Técnicas avanzadas para mejorar la separación entre clases en el espacio de embedding.

1.5.4 Estado del arte

En [16] el autor del proyecto realizó un reconocimiento de vehículos mediante una cámara al aplicar una etapa de preprocesamiento mediante OpenCV con el fin de segmentar en frames el video, luego delimitaron cada vehículo en los diferentes frames para poder entrenar su modelo de predicción basado en YOLO para el reconocimiento y el algoritmo DeepSort para dar el seguimiento de los vehículos.

Figura 1.3

Secuencia de desarrollo para Reidentificación



Nota: Tomada de [6]

En [17] se aprecia un video de tracking de carros y personas mediante una cámara que se encuentra dentro de un carro en movimiento. De manera consistente con el caso anterior, se emplearon las mismas tecnologías, utilizando OpenCV para el preprocesamiento del video, YOLO tanto para el entrenamiento del modelo detector como para la inferencia de detecciones, y DeepSORT para el seguimiento temporal de las cajas delimitadoras a lo largo de la secuencia.

En [18] para el ReID de vehículos utilizan fastReID con el fin de codificar las características del objetivo y extraer un vector de 4096 características que se usan para volver a identificar el vehículo. En caso de haber similitudes en el vector de características, utilizan la distancia euclidiana entre sus vectores para saber sí es el mismo o no.

En [19] se emplea una cámara fija en entorno urbano para la detección, seguimiento y ReID de vehículos con varias etapas integradas. Primero, se aplica un detector (modelo YOLO) al flujo de video para localizar vehículos en cada fotograma. Tras esto, se utiliza un tracker múltiple, combinando un algoritmo de tracking (StrongSORT) para asignar identificadores únicos a cada vehículo y seguir su trayectoria a lo largo del video. Adicionalmente, se integran técnicas de reconocimiento automático de matrículas (ALPR) con OCR, que permite validar y mejorar la ReID correlacionando tracks con la lectura de placas. Finalmente, se compara la similitud de apariencias (mediante vectores de

características del vehículo o de la matrícula) y se corrigen posibles errores de tracking, lo que aumenta la precisión del sistema en escenarios reales de movilidad.

En [20] ByteTrack se presenta como un algoritmo de seguimiento multi-objeto (MOT) que se utiliza para asociar detecciones entre frames en un video, manteniendo la identidad de los objetos a lo largo del tiempo. Su principal innovación radica en cómo maneja las detecciones de baja confianza, que normalmente se descartan en otros métodos como SORT o DeepSORT.

ByteTrack evita descartar de entrada las detecciones con puntaje bajo y las reconsidera usando la historia del propio seguimiento: trabaja con dos umbrales y dos pasadas por cuadro (primero asocia con detecciones “altas” y después intenta recuperar los tracks que quedaron sueltos usando detecciones “bajas” si encajan en tiempo y espacio según Kalman. Con ello transforma observaciones débiles (oclusión, mala iluminación, imagen borrosa) en señales útiles y filtra fondo gracias a la coherencia del trayecto, logrando mejoras en IDF1 en MOT17/MOT20 sin perder tiempo real [21].

La novedad se complementa con avances recientes: OC-SORT mejora la dinámica ante oclusiones prolongados, Bot-SORT integra apariencia y compensación de cámara y StrongSORT moderniza DeepSORT; en conjunto ByteTrack ayuda a recuperar identidad y reducir fragmentaciones antes de la etapa de ReID o de la función multicámaras.

En el contexto de la ReID vehicular, Vision Transformer se ha utilizado para generar embeddings altamente discriminativos. Por ejemplo, en [22] se propone una variante de ViT entrenada con pérdida por triplet que permite modelar relaciones espaciales y semánticas entre diferentes partes del vehículo.

Para la evaluación del rendimiento del sistema de ReID, se utilizan las métricas estándar de la literatura: Mean Average Precision (mAP) y Rank-k accuracy [20][22]. La métrica mAP evalúa la precisión promedio a través de todos los niveles de recuperación,

proporcionando una medida integral de la efectividad del ranking. Por otro lado, la métrica Rank-k, particularmente Rank-1, indica el porcentaje de consultas en las que el vehículo correcto aparece en la primera posición del ranking resultante, lo cual es crítico para aplicaciones en tiempo real.

Capítulo 2

2.1 Metodología

Para abordar el problema del rastreo vehicular, se plantearon varias alternativas de solución. El objetivo principal era desarrollar un sistema automatizado que tuviese la capacidad de identificar y rastrear vehículos que cumplan con características específicas de manera eficiente. Las soluciones propuestas se dividieron en 3 enfoques principales:

1. Sistema basado en inteligencia artificial (IA) y aprendizaje profundo: Esta alternativa consistió en la utilización de modelos de aprendizaje automático y redes neuronales convolucionales (CNN, por sus siglas en inglés) entrenadas para reconocer vehículos en imágenes provenientes de múltiples cámaras. En este sistema, la IA sería capaz de identificar automáticamente el color, el modelo del vehículo, y en caso necesario, la matrícula, correlacionando esta información con un repositorio centralizado para identificar vehículos.

2. Sistema avanzado de ReID vehicular**:** Esta alternativa propone un enfoque más sofisticado basado en técnicas de aprendizaje profundo que combinan información visual y contextual para mejorar la ReID de vehículos. En primer lugar, se detectan los vehículos presentes en las cámaras mediante un modelo de visión por computadora. Luego, a través de un sistema de reconocimiento avanzado, se extraen características únicas de cada vehículo, como su forma general, color y otros patrones visuales. Toda esta información se analiza de forma conjunta para determinar si un mismo vehículo ha sido visto en diferentes lugares. Finalmente, se utiliza un sistema de seguimiento que combina esta información para rastrear al vehículo a lo largo de distintas cámaras en el tiempo.

3. Sistema basado en Deep Learning para Detección, Seguimiento y Reidentificación Robustos: Esta alternativa propone una arquitectura modular y de última generación. En la etapa de detección, se utilizaría un modelo de Deep Learning de una sola pasada (como YOLO en su versión más reciente) para localizar vehículos con alta precisión y velocidad. Para el seguimiento intra-cámara, se emplearía un algoritmo de asociación de detecciones que aprovecha tanto las detecciones de alta como de baja confianza para mantener la identidad de los vehículos incluso bajo oclusiones temporales, reduciendo significativamente los cambios de ID. La Reidentificación (ReID) multicámara se abordaría mediante un modelo basado en arquitecturas Transformer, que captura dependencias globales y contextos de largo alcance en las imágenes. Este modelo se entrenaría con una función de pérdida híbrida que combina pérdida de entropía cruzada para clasificación de identidades y pérdida por tripletes con minería de ejemplos difíciles para garantizar que los embeddings de un mismo vehículo estén más cercanos en el espacio de características que los de vehículos diferentes.

Después de evaluar las ventajas y limitaciones de cada enfoque, se seleccionó la **alternativa 3**. Esta solución ofrece una alta precisión en el reconocimiento de patrones y es capaz de aprender y adaptarse a nuevos datos, lo que la convierte en la opción más adecuada para el rastreo de vehículos en entornos urbanos complejos.

2.2 Requerimientos

2.2.1 Requerimientos Funcionales

Tabla 2.1

Requerimientos Funcionales

Código	Requerimiento Funcional	Descripción
RF1	Detección de vehículos	El sistema debe detectar vehículos en tiempo real o desde videos grabados mediante un modelo de visión por computadora, como YOLO.
RF2	Extracción de características visuales	El sistema debe extraer características distintivas de los vehículos, como forma, tamaño y detalles particulares de la carrocería.
RF3	Generación de vectores de características	El sistema debe generar un descriptor o vector representativo de cada vehículo detectado.
RF4	Comparación entre cámaras	El sistema debe comparar las características de los vehículos detectados en diferentes cámaras para determinar coincidencias.
RF5	Identificación consistente del vehículo	El sistema debe mantener una identidad única y coherente para cada vehículo detectado, incluso cuando reaparezca en diferentes momentos o cámaras, utilizando algoritmos de seguimiento como ByteTrack.
RF6	Seguimiento multicámara	El sistema debe realizar el seguimiento de un mismo vehículo entre diferentes cámaras, manteniendo su identidad.

RF7	Visualización de resultados	El sistema debe proporcionar una visualización de los resultados obtenidos, incluyendo los Ids de seguimiento asignados a cada vehículo.
RF8	Procesamiento de múltiples videos	El sistema debe permitir la carga y procesamiento de múltiples videos, ya sea en serie o simultáneamente.

2.2.2 Requerimientos no funcionales

Tabla 2.2

Requerimientos no Funcionales

Código	Requerimiento No Funcional	Descripción
RNF1	Tiempo de procesamiento aceptable	El sistema debe procesar los videos en un tiempo proporcional a la longitud de los videos y a los recursos computacionales.
RNF2	Precisión	El sistema debe mantener una exactitud ≥ 90% en la tarea de identificación vehicular, evaluada sin intervención del módulo de seguimiento, minimizando falsos positivos y negativos.
RNF3	Escalabilidad	El sistema debe ser capaz de funcionar eficientemente al trabajar con múltiples cámaras y grandes volúmenes de datos.
RNF4	Modularidad	El sistema debe estar diseñado de forma modular para facilitar la sustitución o mejora de componentes individuales.

RNF6	Facilidad de uso	La interfaz del sistema debe ser intuitiva para usuarios con o sin experiencia técnica.
RNF8	Documentación	El sistema debe incluir documentación clara que permita su instalación, configuración, ejecución y mantenimiento.

2.3 Alcance de la solución

1. **Detección de vehículos en video**: El sistema utilizará el detector de objetos YOLOv8 para localizar vehículos en fotogramas de video de manera eficiente y con alta precisión, generando cuadros delimitadores para cada instancia detectada.
2. **Seguimiento de vehículos en múltiples cámaras con ByteTrack:** Mediante el algoritmo ByteTrack, el sistema mantendrá una identidad temporal única para cada vehículo dentro de un mismo flujo de video. Este algoritmo asociará detecciones entre fotogramas aprovechando tanto detecciones de alta como de baja confianza, mejorando el seguimiento en condiciones de oclusión parcial y reduciendo significativamente los cambios de identificación.
3. **Extracción de características para ReID con Vision Transformer (ViT**): Cada vehículo detectado será procesado por un modelo basado en la arquitectura Vision Transformer (ViT) para generar un vector de características (embedding) de alta dimensionalidad. Este enfoque captura aspectos distintivos del vehículo de manera robusta, siendo independiente de la perspectiva, iluminación y oclusiones menores.
4. **Reidentificación (ReID) multicámara mediante aprendizaje métrico:** Los embeddings generados por el modelo ViT se utilizarán para comparar vehículos avistados en diferentes cámaras. La similitud se calculará utilizando métricas de distancia (distancia coseno) y técnicas de post-procesamiento como re-ranking para mejorar la precisión en el emparejamiento y reducir falsos positivos.

5. **Procesamiento de múltiples videos**: La arquitectura permitirá la carga y el procesamiento simultáneo o secuencial de videos provenientes de múltiples cámaras, facilitando el análisis de escenarios de vigilancia distribuida.
6. **Visualización de resultados**: El sistema incluirá una interfaz que superpondrá los identificadores únicos y las trayectorias de los vehículos sobre el video. Además, proporcionará funcionalidades para consultar el historial de apariciones de un vehículo específico a través de las diferentes cámaras disponibles.

2.4 Limitaciones de la solución

1. **Condiciones de iluminación**: La precisión del sistema puede verse afectada en condiciones de baja iluminación o luz artificial inadecuada, especialmente durante la noche, lo que impacta tanto en la detección con YOLOv8 como en la extracción de características con Vision Transformer**.**
2. **Oclusiones parciales o totales**: Si un vehículo es bloqueado por otros objetos o no es visible completamente, puede dificultar su correcta detección y seguimiento. Aunque ByteTrack está diseñado para manejar oclusiones temporales, las oclusiones prolongadas pueden llevar a la pérdida del seguimiento.
3. **Ángulos de cámara limitados**: El rendimiento del sistema depende de la ubicación y el campo visual de las cámaras; ángulos muy inclinados o alejados pueden reducir la precisión en la detección y en la generación de embeddings para la ReID**.**
4. **Calidad del video**: Videos con baja resolución, compresión excesiva o alta tasa de distorsión pueden afectar negativamente la detección con YOLOv8 y la extracción de características con el modelo ViT**,** reduciendo la confiabilidad del ReID multicámara.
5. **Diferencias en la apariencia del vehículo**: Cambios en las condiciones del vehículo, como suciedad, daños, accesorios añadidos, o diferentes condiciones climáticas

(lluvia, nieve), pueden afectar la consistencia de los embeddings generados por el modelo de ReID**,** llevando a falsos negativos.

6. **Dependencia del modelo de seguimiento**: El sistema depende del rendimiento del algoritmo de seguimiento ByteTrack**,** el cual puede generar errores de asociación en secuencias con alta densidad vehicular, trayectorias cruzadas o movimientos bruscos, resultando en cambios de Ids.
7. **Variabilidad entre cámaras**: Diferencias significativas en la calibración de color, resolución o perspectiva entre las múltiples cámaras pueden introducir desafíos adicionales para el módulo de ReID, que debe ser robusto a estas variaciones de dominio.
8. **Capacidad computacional:** El procesamiento en tiempo real de múltiples flujos de video con modelos Deep Learning (YOLOv8, ViT) requiere recursos hardware significativos (GPUs), lo que podría limitar el despliegue en entornos con recursos restringidos.

2.5 Riesgos y beneficios de la solución

Los riesgos se representan en la siguiente tabla:

Tabla 2.3

Riesgos de la solución

ID	Riesgo	Impacto	Probabilidad	Mitigación
01	Falsos positivos/negativos en la detección.	Alto	Media	Ajustar umbrales de confianza.

				Validar continuamente los datos.
02	Variabilidad de iluminación	Medio	Media	Preprocesado adaptativo.
03	Sobrecarga computacional	Alto	Media	Despliegue en GPU dedicada.

Los beneficios que el proyecto plantea son:

Tabla 2.4

Beneficios de la solución

ID	Beneficio	Descripción
01	Automatización de la vigilancia vehicular	El sistema detecta y rastrea vehículos reduciendo la carga operativa.
02	ReID multicámara	Posibilidad de seguir una misma unidad vehicular en distintas cámaras, facilitando análisis de rutas.
03	Escalabilidad y modularidad	Arquitectura basada en módulos independientes que permite actualizar o reemplazar componentes sin tener que rehacer todo el sistema.

2.6 Usuarios de la solución

Los usuarios que podrían verse relacionados directamente con el proyecto son:

1. Operador de vigilancia: Inicia la plataforma, carga los videos y observa en pantalla los vehículos detectados.
2. Administrador del sistema y desarrollador: se encargarán de mantener el entorno, actualizar el modelo y dar soporte como nuevas funciones al proyecto.

2.7 Prototipado

En el prototipado del interfaz desarrollado en Figma, el flujo inicia con la carga de videos. Al invocar la acción “cargar videos”, se abre un selector de archivos que permite escoger un video etiquetándolo con el día al que pertenece.

Una vez cargados los videos, el usuario puede dirigirse al módulo de “Buscar”, donde un control deslizante de días facilita la recolección de un día específico. Al confirmar la búsqueda, la pantalla central despliega únicamente el video o los fragmentos que caen dentro del criterio establecido, agilizando la tarea de hallar el momento preciso de interés.

El lienzo principal muestra el video con recuadros en torno a los vehículos detectados junto a un identificador único asignado a cada vehículo. Los operadores podrán observar si la detección es fidedigna.

La opción de “Seleccionar un vehículo” despliega una pantalla donde se pueden visualizar todos los vehículos detectados en el video. Adicional a esto, en la parte inferior se pueden utilizar filtros de color y modelo vehicular.

Cuando seleccionan una miniatura, se visualiza el recorrido que hizo el vehículo, así como sus características y cuenta con un botón “Detalles” para ver a más profundidad el recorrido y tiempo de recorrido del vehículo detectado.

2.8 Diseño de la solución

El sistema se organiza en cuatro capas principales:

Capa de Ingreso de Datos:

Aquí se gestionan la carga y el preprocesamiento de los videos de las distintas cámaras.

Capa de Procesamiento de Visión:

Esta capa constituye la etapa inicial de análisis visual. Recibe los flujos de video raw de las múltiples cámaras y ejecuta las siguientes operaciones:

- **Preprocesamiento:** Cada fotograma es normalizado mediante ajuste de dimensiones (resizing a 640x640 píxeles), corrección de contraste y balance de blancos para optimizar la entrada al modelo de detección.
- **Detección de Vehículos:** Se emplea el modelo **YOLOv8** preentrenado en el dataset **COCO** (Common Objects in Context). Específicamente, se filtran las detecciones para considerar solamente las clases relacionadas con vehículos:
    - Coche (car)
    - Motocicleta (motorcycle)
    - Autobús (bus)
    - Camión (truck)
    - Furgoneta (van)

Este enfoque permite una detección rápida y precisa de los vehículos de interés, ignorando otras clases presentes en COCO como personas o animales. El modelo genera bounding boxes con sus respectivos porcentajes de confianza para cada vehículo detectado.

- **Preparación de ROIs:** Las regiones de interés (ROIs) correspondientes a cada bounding box son recortadas y preparadas para ser procesadas por las capas subsiguientes.

- **Extracción de características:** El extractor de características no extrae valores numéricos simples como “alto: 1.5m”, “largo: 4.2m” o “color: azul RGB(0,0,255)”.

  En su lugar, funciona de una manera abstracta:

  1. **Procesamiento con Red Neuronal (ViT):** La imagen del vehículo se introduce en el modelo Vision Transformer.
  2. **Extracción de Patrones Abstractos:** A medida que la imagen pasa por las capas de la red neuronal, el modelo analiza y combina patrones de bajo nivel (bordes, texturas, colores) para formar conceptos de nivel superior (formas de faros, estilos de parrillas, contornos de ventanas, adhesivos, abolladuras, etc.).
  3. **Creación del “Embedding” o “Huella Digital”:** El resultado final es un vector numérico de alta dimensionalidad (512 números). Este vector es una representación densa y comprimida de toda la apariencia visual del vehículo.

Tabla 2.5

Representación de embeddings

ID Vehículo	Embedding (Vector de 512 números)
#152	[0.12, -0.45, 0.88, 1.24, -0.93, …, 0.027] (Este vector único es la representación numérica de todas sus características combinadas)
#153	[-0.87, 0.21, 1.45, -0.62, 0.11, …, -0.451]

Capa de Tracking y ReID:

Esta capa constituye el núcleo del sistema y opera en dos fases principales:

1. **Seguimiento en cámara (Tracking):** Utilizando el algoritmo **ByteTrack, esta** fase mantiene la identidad consistente de cada vehículo dentro de un mismo flujo de video. ByteTrack aprovecha todas las detecciones (alta y baja confianza) y utiliza un filtro de Kalman para predecir la posición futura de los tracks existentes. Mediante el algoritmo húngaro, asocia las detecciones actuales con los tracks basándose principalmente en la similitud por intersección sobre unión (IoU), asignando un ID único persistente mientras el vehículo permanezca en la escena. Esto permite manejar oclusiones temporales y reapariciones.
2. **Reidentificación Multicámara (ReID):** Para cada vehículo detectado y trackeado, se extrae un embedding profundo utilizando un modelo basado en Vision Transformer (ViT). Este convierte la región de interés (ROI) del vehículo en un vector de características de alta dimensionalidad que encapsula su apariencia visual de manera robusta e independiente al punto de vista. Estos embeddings se almacenan en una base de datos temporal indexada por tiempo, cámara e ID de track. Un módulo de matching compara los nuevos embeddings contra los existentes utilizando distancia coseno y técnicas de re-ranking para establecer correspondencias entre las diferentes vistas de un mismo vehículo en cámaras distintas, construyendo así una trayectoria global.

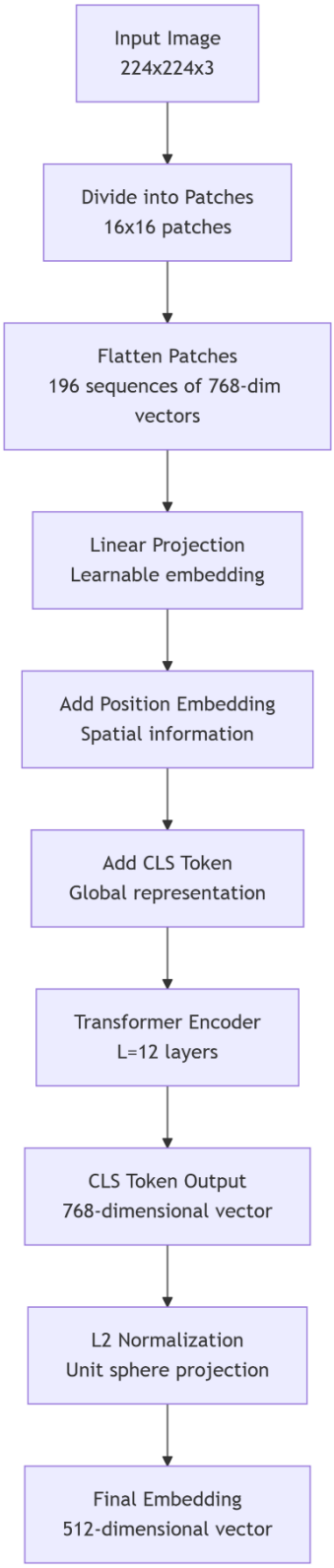
   Método de comparación: La distancia coseno se utiliza para comparar los vectores de características generados por el modelo de ReID debido a sus propiedades matemáticas ideales para medir similitud en espacios de alta dimensionalidad. A diferencia de la distancia euclidiana, que mide la distancia absoluta entre dos puntos, la distancia coseno calcula el coseno del ángulo entre dos vectores, lo que la hace invariante a la magnitud de estos [23].

   Esto es particularmente ventajoso en ReID vehicular porque:

1. **Enfoca en la dirección, no en la magnitud:** La "apariencia" única de un vehículo está codificada en la *dirección* del vector en el espacio de características. Dos imágenes del mismo vehículo, incluso con diferentes niveles de brillo o contraste (que afectarían la magnitud del vector), tendrán direcciones muy similares.
2. **Robusta a variaciones de iluminación:** Cambios en la iluminación que alteran globalmente los valores de píxeles (aumentando o disminuyendo la magnitud del *embedding*) no afectan el ángulo entre vectores, por lo que la similitud calculada permanece estable.
3. **Eficacia en alta dimensionalidad:** Funciona excepcionalmente bien en espacios con cientos o miles de dimensiones, como los *embeddings* generados por modelos Transformer o CNN, donde la "maldición de la dimensionalidad" puede hacer que otras métricas sean menos efectivas.

Figura 2.1

Proceso de extracción de características o embedding



Descripción Detallada de las Etapas:

1. **Input Image (Imagen de Entrada):**
    - La imagen del vehículo se redimensiona a una resolución fija (ej. 224x224 píxeles).
    - Representada como un tensor de dimensiones: [3, 224, 224] (canales, altura, anchura).
2. **Patch Partition (División en Parches)** [24]**:**
    - La imagen se divide en parches pequeños de tamaño fijo (ej. 16x16 píxeles).
    - Para una imagen 224x224, se obtienen 196 parches (224/16 = 14 → 14x14 = 196 parches).
    - Cada parche se aplana en un vector 1D de longitud 16*16*3 = 768.
3. **Linear Projection (Proyección Lineal)** [24]**:**
    - Cada vector de parche (768-dim) se proyecta a una dimensión constante D (ej. D=768) usando una capa lineal entrenable (nn.Linear).
    - Esto transforma cada parche en un "token" o vector de características de dimensión D.
4. **Position Embedding (Embedding Posicional)** [24]**:**
    - Se añaden embeddings posicionales aprensibles a cada token. Estos le indican al modelo la posición original de cada parche en la imagen.
    - Es crucial porque el Transformer, por sí mismo, no tiene noción de orden espacial.
5. **[CLS] Token (Token Especial)** [25]**:**
    - Se antepone un token especial [CLS] (class token) a la secuencia de tokens de parches.

- Este token actuará como un acumulador de información global de toda la imagen. Su estado final en la última capa se utilizará como el **embedding** global de la imagen.

6. **Transformer Encoder (Codificador Transformer)** [26]
    - La secuencia de tokens ([CLS] + tokens_de_parches) se alimenta a una pila de L capas Transformer (ej. L=12).
    - Cada capa consta de:
        - **Multi-Head Self-Attention (MHSA):** Permite que cada token (incluido [CLS]) interactúe y atienda a todos los demás tokens de la secuencia. Esto es lo que captura las relaciones contextuales globales (ej. Que la forma del capó y la marca del vehículo están relacionadas).
        - **Feed-Forward Network (FFN):** Una red neuronal fully-connected para cada token que aplica transformaciones no lineales.
7. **Output (Salida):**
    - Tras pasar por las L capas, se toma la representación final del token [CLS] (un vector de dimensión D=768) como la representación codificada de toda la imagen.
8. **L2 Normalization (Normalización L2)** [27]**:**
    - Este vector [CLS] se normaliza para que tenga norma unitaria (se proyecta sobre una esfera de radio 1).
    - Esto es fundamental porque la **similitud coseno** entre dos vectores normalizados es equivalente a su producto punto.
    - embedding_normalizado = embedding / $||\text{embedding}||_2$

9. **Final Embedding (Embedding Final):**
    - El resultado es un vector de características denso y de dimensionalidad fija (ej. 512, 768, o 1024 dimensiones) que representa la "firma" única del vehículo.
    - Este *embedding* es:
        - **Discriminativo:** Diferentes vehículos tendrán *embeddings* muy separados en el espacio vectorial.
        - **Robusto:** El mismo vehículo bajo diferentes condiciones (luz, ángulo, oclusión parcial) tendrá *embeddings* muy cercanos.

Capa de Presentación:

Esta capa consolida toda la información generada y proporciona una interfaz visual para el usuario. Incluye:

- Un visualizador de video que reproduce los flujos con overlays superpuestos que muestran los bounding boxes, los Ids de track asignados por ByteTrack y los Ids globales asignados por el sistema de ReID.
- Un catálogo de vehículos interactivo que permite buscar vehículos y visualizar sus trayectorias completas a través del área monitorizada.
- El uso de la librería Leaflet de JavaScript que permite visualizar datos georeferenciados así como crear un mapa interactivo.

Esta arquitectura de cuatro capas asegura un flujo de procesamiento modular y escalable, donde cada componente puede ser mejorado o reemplazado independientemente (por ejemplo, actualizando el detector a una versión más nueva de YOLO o el extractor de características a un Transformer más avanzado).

2.9 Experimento del modelo

2.9.1 Entrenamiento

El entrenamiento se realizó sobre el dataset VeRi-776 (Ids vehiculares repartidos entre train y test), utilizando un loader específico que en train, lee la carpeta 'image_train', y en test concatena 'image_query' + 'image_test'. Las etiquetas se derivan del ID de vehículo en el nombre de archivo. La composición de mini-lotes emplea un muestreador PK, es decir, una estrategia que, en cada mini-lote selecciona P identidades distintas y para cada identidad seleccionada, se toman K instancias de esa identidad. En este caso, con P = 8 entidades y K = 4 instancias por identidad, de modo que cada lote (de 32 imágenes) contenga suficientes pares positivos y negativos para el aprendizaje estable [28].

Sobre esta base de datos y loader, se efectuó un ajuste fino supervisado de un Visión Transformer (ViT-Base/16, 224x224) al que se añadió un BNNeck (BatchNorm1d sobre el embedding) y una cabeza lineal de clasificación con tantas salidas como identidades del split de entrenamiento. Durante el forward de entrenamiento el modelo produce:

- Un embedding discriminativo
- Puntajes de clase

Ambos alimentan una pérdida compuesta formada por Cross-Entropy con label smoothing (para maximizar la separabilidad entre identidades) y Circle Loss (para optimizar, dentro del lote PK, la cohesión de positivos y la separación de negativos).

Se dejó apagada la Triplet Loss porque la combinación Cross-Entropy + Circle Loss ya daba una señal estable. Se entrenó con AdamW usando 10 épocas de calentamiento y luego un descenso tipo coseno hasta la época 60. En entrenamiento se aplicó aumentos estándar: redimensionar a 224x224, volteo horizontal, padding + recorte aleatorio, ligero cambio de color, normalización ImageNet y borrado aleatorio. En test solo redimensionar y normalizar. Cada 5 épocas se validaron con el protocolo de VeRi-776 (query vs gallery,

excluyendo “mismo ID y misma cámara”), reportando Rank-1/Rank-5 y mAP, almacenando el mejor modelo según mAP [28].

Se eligió ViT-Base/16 con BNNeck y un muestreo PK(8x4) porque así el modelo ve en cada lote suficientes ejemplos de la misma y de distintas identidades, lo que facilita aprender un embedding realmente discriminativo; se combinó Cross-Entropy (con label smoothing) para separar clases y Circle Loss para empujar positivos difíciles juntos y negativos lejos; se dejó Triplet apagada para evitar su inestabilidad y el “mining” sensible al ruido. Se utilizó AdamW con calentamiento y coseno porque estabiliza el ajuste fino de Vit y mejora la generalización; las aumentaciones (flip, crop, random erasing) simulan cambios de cámara, iluminación y oclusiones, reduciendo el overfitting. Se valida cada 5 épocas con el protocolo cross-camera de VeRi-776 y se almacena el mejor MAP porque refleja mejor la calidad del ranking completo que solo el Top-1 [29]

2.9.2 Captura de videos para testing

Se utilizaron varias cámaras de seguridad ubicadas en puntos estratégicos de la Escuela Superior Politécnica del Litoral situada en Guayaquil, Ecuador. Estas cámaras proporcionaron tomas reales de vehículos para su posterior análisis.

2.9.3 Testing

Se realizaron pruebas sobre las imágenes del dataset VeRi-776, para medir la precisión del modelo entrenado y después se realizaron pruebas con videos reales, para ver la eficacia del algoritmo en detección, tracking, extracción de embeddings y emparejamiento. Los videos con los que se hicieron las pruebas están enfocados en el campus de la Escuela Superior Politécnica del Litoral.

2.10 Arquitectura

Esta arquitectura modular facilita la escalabilidad (sustitución de modelos o aceleradores de hardware), la mantenibilidad (cada componente encapsula una

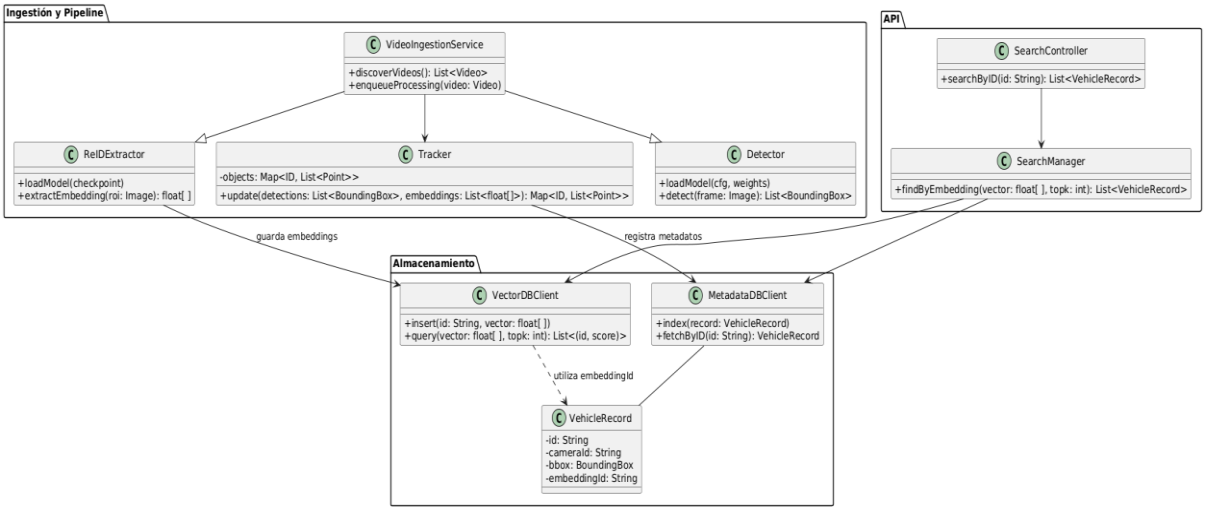
responsabilidad clara) y el cumplimiento de requisitos de rendimiento y privacidad (filtros de anonimización, niveles de logs y permisos). A continuación, se detallan, para cada una de las cinco vistas del modelo “4 + 1” de Kruchten, los diagramas que ilustran la estructura lógica, los casos de uso, las interacciones, el flujo de procesos y el despliegue físico de la solución.

2.9.1 Modelo 4+1

2.10.1 Diagrama de clases

Figura 2.2

Diagrama de clases



Ingestión y análisis: Un servicio recibe y encola los videos, un detector localiza cada coche en los fotogramas, un extractor genera su “huella digital” (embedding) y un tracker agrupa esas huellas para darle a cada vehículo un ID y una trayectoria.

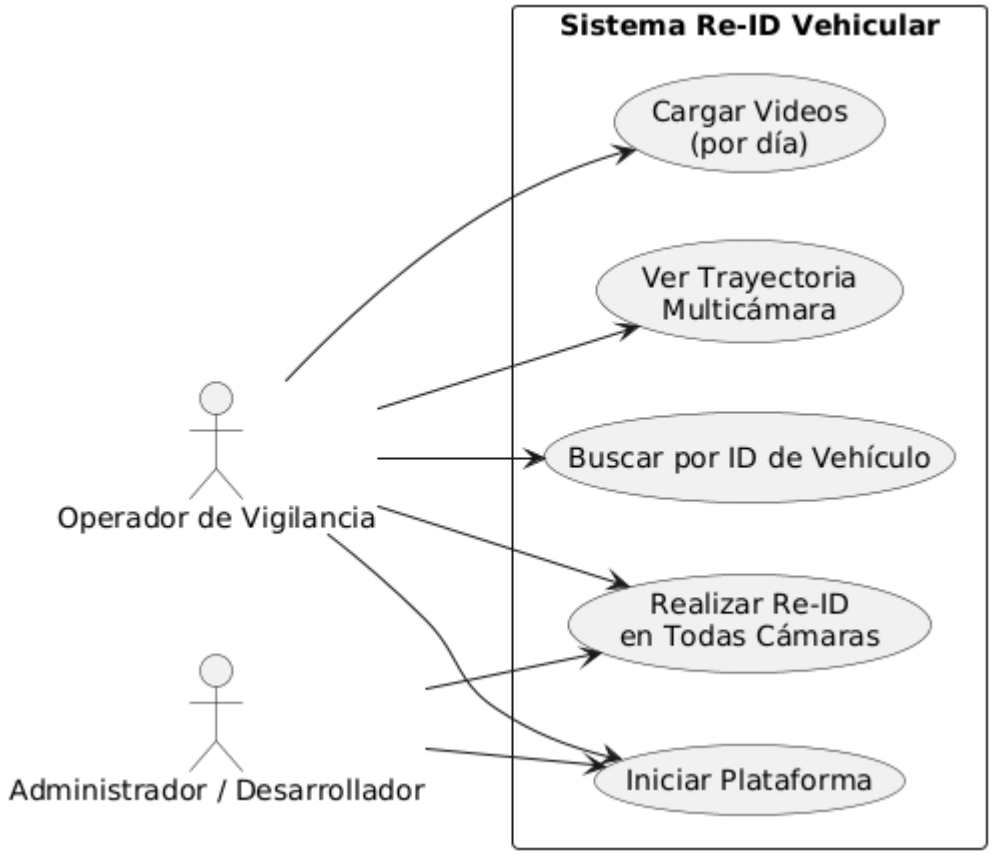
Almacenamiento: Los embeddings van a una base de vectores para búsquedas rápidas, y las fichas de aparición (ID, cámara, hora y posición) se guardan en una base de metadatos.

Búsqueda: La API recibe peticiones de búsqueda por ID, consulta primero la base de vectores para encontrar coincidencias y luego recupera las fichas en la base de metadatos para devolver al usuario todas las apariciones del vehículo sin tener que reprocesar el video.

2.10.1.1 Diagrama casos de uso

Figura 2.3

Diagrama casos de uso

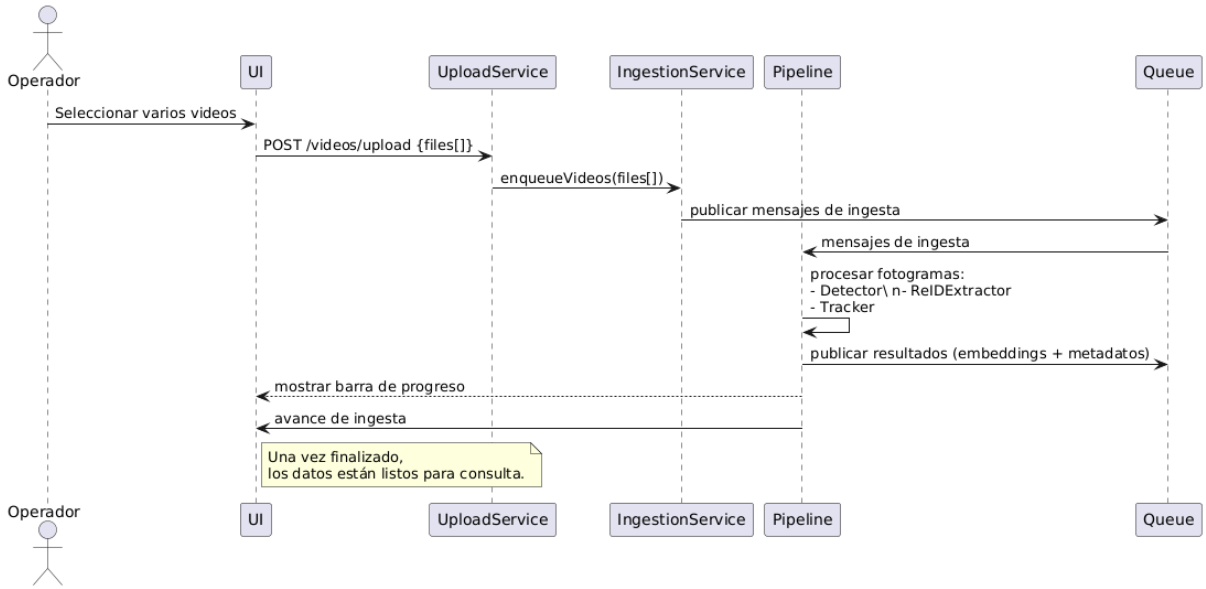


El operador de vigilancia inicia la plataforma, carga uno o varios videos, y luego solicita la Reidentificación de vehículos en todas las cámaras o la consulta por un ID de vehículo concreto, recibiendo a cambio todas sus apariciones y trayectorias. El administrador también puede acceder a la plataforma para mantener el entorno, actualizar modelos y parámetros, y desplegar nuevas funciones.

2.10.1.2 Diagrama secuencia

Figura 2.4

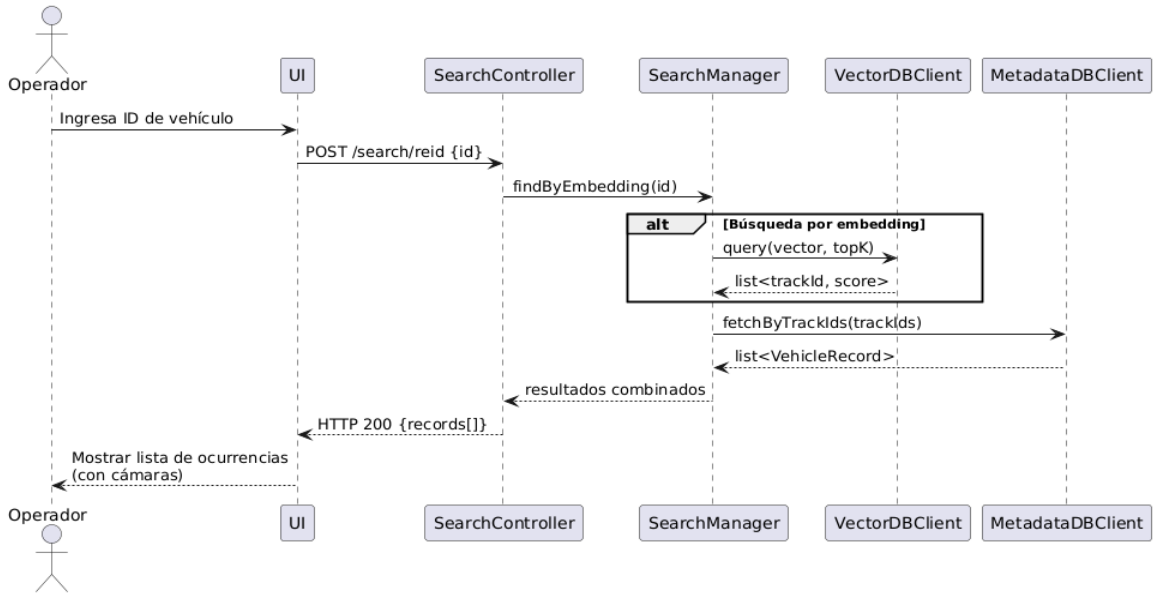
Diagrama de secuencia para seleccionar videos



El operador elige varios archivos de video y pulsa “Subir” en la pantalla. El sistema recibe esos videos y los pone en una lista de espera. Uno a uno, cada video es procesado: primero se extraen las imágenes, luego se buscan los coches en cada fotograma y, finalmente, se guardan todos los datos (qué coche, en qué cámara y a qué hora). Mientras esto sucede, la interfaz muestra una barra que avanza hasta completar el proceso y avisa al Operador de que ya puede hacer búsquedas.

Figura 2.5

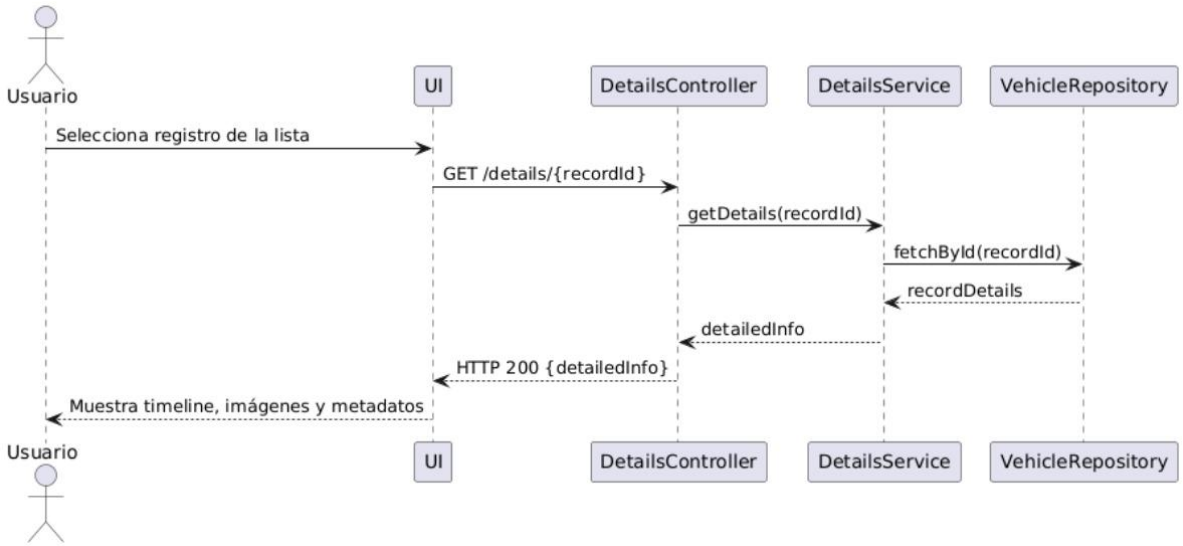
Diagrama de secuencia para seleccionar vehículo con ID



El operador introduce en la pantalla el número de identificación de un coche que ya se había detectado anteriormente y pulsa “Buscar”. El sistema compara rápidamente las características de ese coche con todas las guardadas y recupera las veces que apareció en cada cámara. Después muestra en la pantalla una lista clara de todas las apariciones: cámara, hora y posición exacta en el fotograma.

Figura 2.6

Diagrama de secuencia para seleccionar vehículo en una lista



Aquí el Usuario selecciona un elemento de la lista de resultados (un ID único de vehículo). La UI ejecuta GET /details/{recordId} al DetailsController, que llama a DetailsService.getDetails(recordId). Este servicio consulta el VehicleRepository.fetchById(recordId) para extraer toda la información: timeline de posiciones, miniaturas de fotogramas, tiempos y atributos. Esa información detallada (detailedInfo) retorna en cascada al controlador, que responde a la UI con HTTP 200 {detailedInfo}, y la interfaz muestra la línea de tiempo, las imágenes y los metadatos asociados.

2.10.1.3 Diagrama flujo de procesos

Figura 2.7

Diagrama de flujo de procesos

Este diagrama reúne los tres escenarios en un solo recorrido:

Subida: el operador carga los videos y el sistema los procesa automáticamente.

Búsqueda: una vez listo, el operador introduce un ID y ve al instante todas las apariciones del coche.

Capítulo 3

3.1 Resultados y análisis

Para evaluar el funcionamiento del sistema se realizaron dos pruebas:

- Testing con dataset de entrenamiento (Veri-776).
- Testing con videos reales.

3.1.1 Testing con dataset de entrenamiento

El dataset consta de 11579 imágenes que proporcionan los datos de ID del vehículo (vehicleID), ID de la cámara a la que pertenece la toma (cameraID) y un ID para el tipo de vehículo (typeID). Para efectos de análisis de sólo vehículos se eliminaron todos los que correspondían a typeID = 9 ya que son buses. Luego de este filtro quedaron 9749 imágenes para analizar.

Se probaron 3 vehicleID aleatorios (5, 101, 177) para realizar el proceso de identificación, extracción de características y tracking. Cada vehículo tiene varias tomas dentro de una misma cámara. Las cuáles representan "frames" de un vídeo.

Tabla 3.1

Imágenes en las que aparece un vehículo específico

vehicleID	# de identificaciones manuales	# de identificaciones del sistema
5	58	58
101	42	42
177	42	42

Luego de obtener todas las imágenes de un vehículo se obtuvieron los embeddings respectivos de cada imagen, los mismos permitieron comparar la similitud con otros embeddings generados.

Tabla 3.2

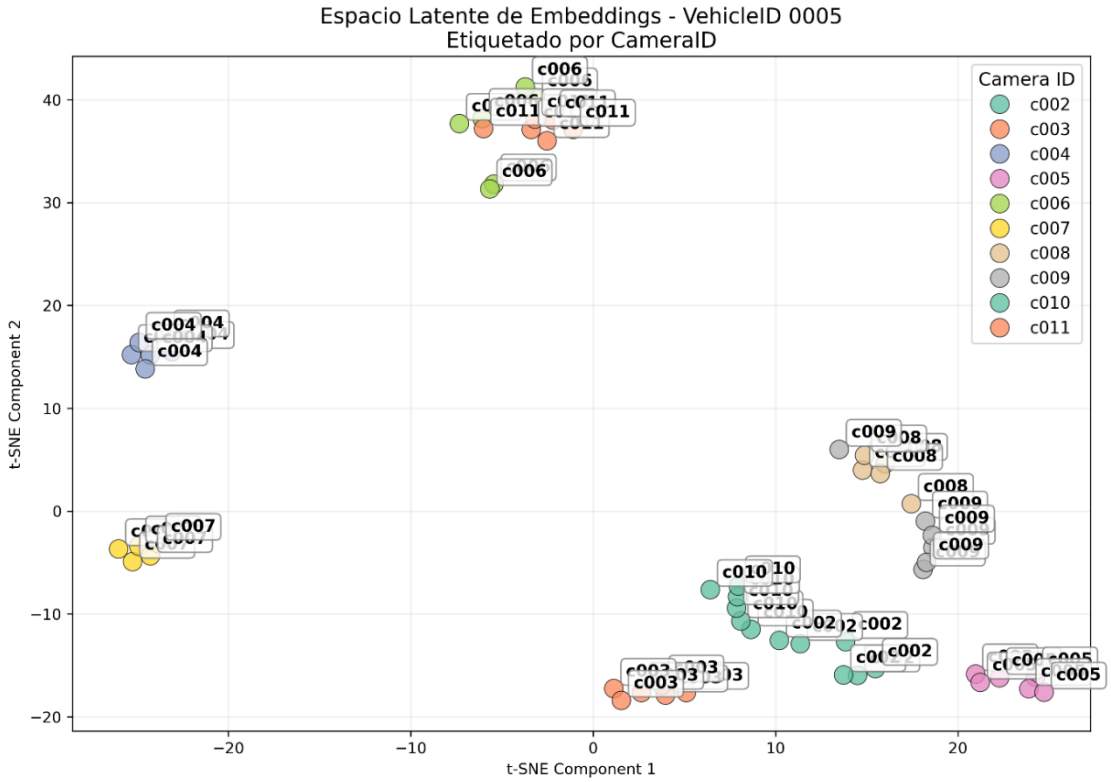
Embeddings generados por vehículo

vehicleID	# de embeddings por ID
5	58
101	42
177	42

Los embeddings obtenidos fueron graficados en un espacio vectorial en el cual se pudo observar asociaciones según su similitud de enfoque, posición y detalles visuales.

Figura 3.1

Espacio vectorial de embeddings del vehículo con ID 5



Un caso particular se da en el vehículo con ID 5 en el que se puede observar que las tomas borrosas o con un enfoque diferente se encuentran totalmente a la izquierda (cámara 4 y 7 respectivamente) en el espacio vectorial (compárese con figura 3.1.2). Otro caso se da en las tomas de la cámara 8 y 9 donde existen obstrucciones parciales del vehículo y al graficarlos se encuentran cercano el uno del otro.

Tabla 3.3

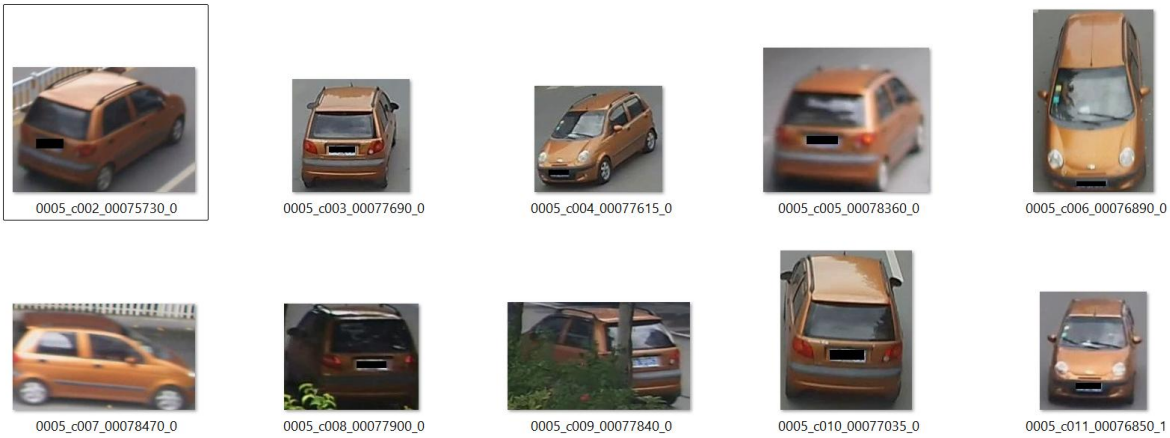
ReID en diferentes cámaras

vehicleID	# cámaras únicas en las que se identifica manualmente	# cámaras únicas en las que se identifica por el sistema
5	11	11
101	7	7
177	7	7

El sistema logra reidentificar el vehículo con ID 5 en todas las cámaras (y con diferente ángulo) en las que se tenía etiquetado el dataset de Veri-776.

Figura 3.2

Reidentificación del vehículo con ID 5



Para finalizar el testing del dataset Veri-776, se realizó el tracking de toda la ruta del vehículo

Figura 3.3

Tracking del vehículo con ID 5

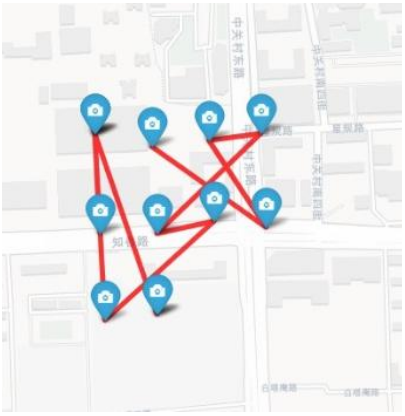
```
--- Construyendo la secuencia de cámaras a partir del log ---
--- Secuencia final para el mapa: ['c002', 'c006', 'c011', 'c006', 'c011', 'c006', 'c010', 'c004', 'c003', 'c009', 'c008
', 'c005', 'c007'] ---
Total de cámaras en la ruta: 13

--- Trayectoria del Vehículo ID: 5 ---
c002 -> c006 -> c011 -> c006 -> c011 -> c006 -> c010 -> c004 -> c003 -> c009 -> c008 -> c005 -> c007

¡ÉXITO! Mapa interactivo guardado en: vehicle_route.html
Abre ese archivo en tu navegador para ver la ruta.
```

Figura 3.4

Seguimiento del vehículo con ID 5 en mapa



Nota: Las líneas trazadas se usaron para dar conectividad a las cámaras más no como referencia de ruta literal.

3.1.2 Testing con videos reales

Durante esta fase de testing se probó seguir o trackear 1 vehículo específico en 7 cámaras diferentes, que pertenecen a diferentes bloques de la universidad ESPOL. Arrojando como resultado:

- Las instancias en que los vehículos fueron detectados por primera vez.
- El recorrido que realizaron los vehículos en el campus de ESPOL.
- Los vectores asociados a las características de cada vehículo.

Ejemplo: Auto 1

Figura 3.5

Lista de vehículos encontrados en un video

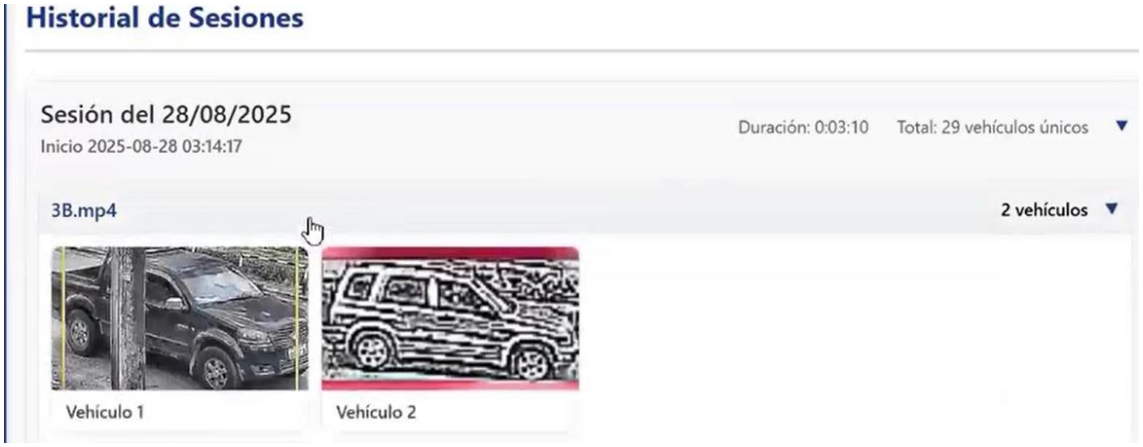
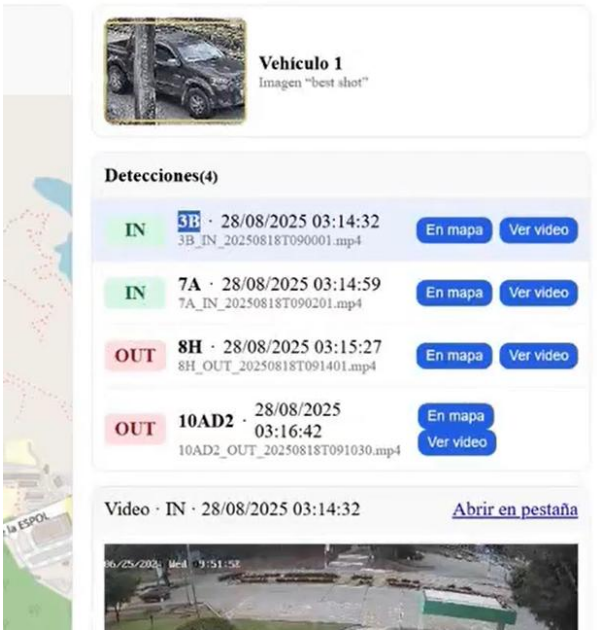


Figura 3.6

Aparición del vehículo 1 en diferentes cámaras



Nota: En la parte izquierda de la imagen aparece el recorrido del vehículo y la posición de las cámaras por las que fue detectado.

Figura 3.7

Seguimiento del vehículo 1

Nuestro sistema de inteligencia artificial, diseñado para seguir e identificar vehículos a lo largo de un video, ha demostrado un rendimiento alto y prometedor. Este éxito es consecuencia directa de las decisiones tomadas en el entrenamiento del modelo y en el diseño

de la arquitectura, que combinó un detector de objetos (YOLOv8) con un extractor de características de última generación (ViT).

Los resultados numéricos, que son el estándar en la industria para medir la precisión de estos sistemas, confirman su eficacia:

- **Una precisión del 94.70% (Rank-1):** El Rank-1 nos dice con qué frecuencia acierta en su primera opción. Un 94.7% significa que, en **95 de cada 100 casos**, el sistema identifica correctamente el vehículo al primer intento. Esto es un indicador bueno de que el modelo ha "aprendido" a crear una "huella digital" única y muy precisa para cada automóvil.
- **Una Precisión Promedio (mAP) del 77.96%:** El mAP es una métrica más compleja y exhaustiva. No solo premia el acierto perfecto, sino que evalúa la calidad general de todas las posibles coincidencias que propone el modelo. Un mAP del 78% indica que, además de ser muy certero en su primera opción, el sistema es consistentemente bueno en rankear las coincidencias correctas cerca de la parte superior de la lista. Esto sugiere que sus predicciones son robustas y fiables sobre una amplia variedad de escenarios.

Tabla 3.4

Resultados de evaluación

Resultados de Evaluación	
Rank-1 Accuracy	94.70%
mAP (mean Avg. Precision)	77.96%

Ejemplo de comparación de vehículos

Para muestra de los resultados, se tomaron imágenes genéricas de internet para mostrar de forma resumida cómo funciona el proyecto con la extracción de características y comparación de estas.

Figura 3.8

Comparación de vehículos iguales y muestra de embeddings

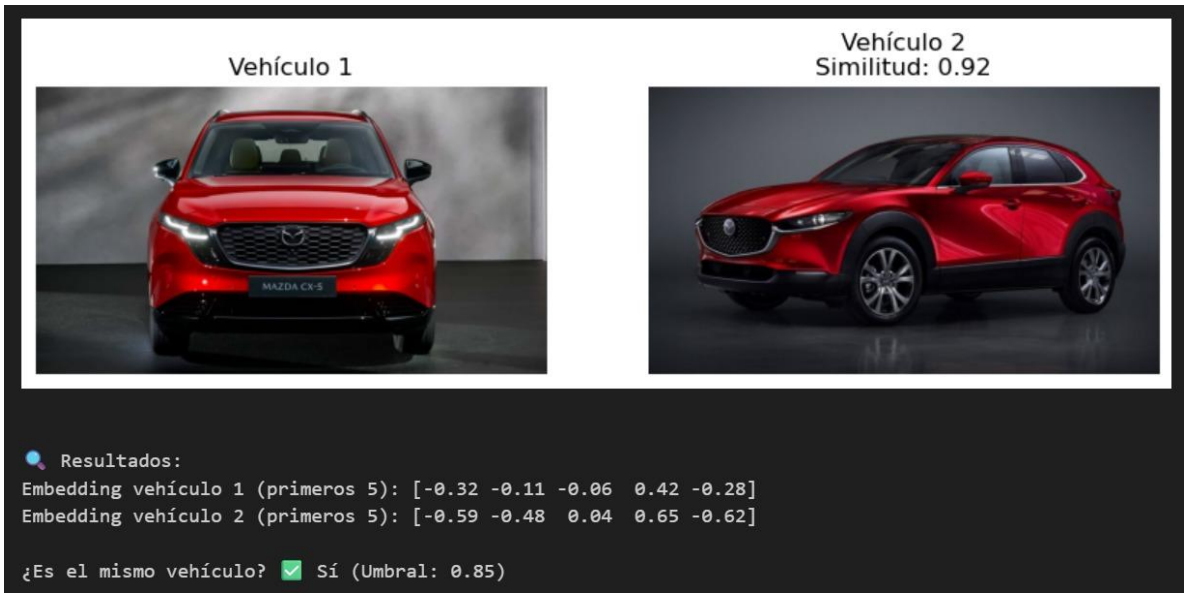
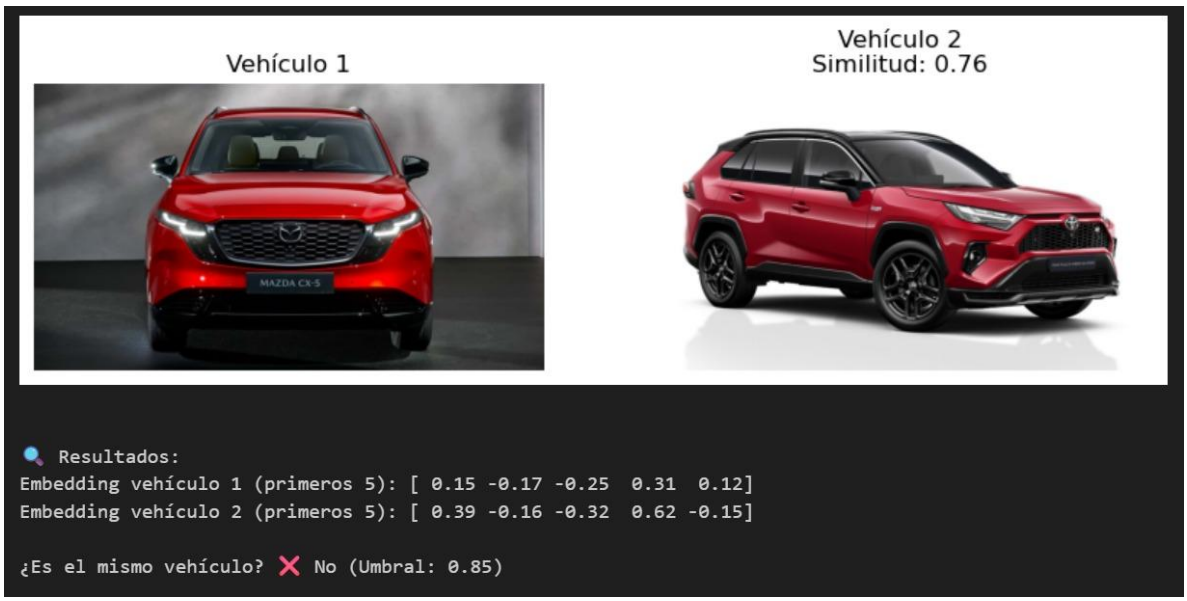


Figura 3.9

Comparación de vehículos diferentes y muestra de embeddings



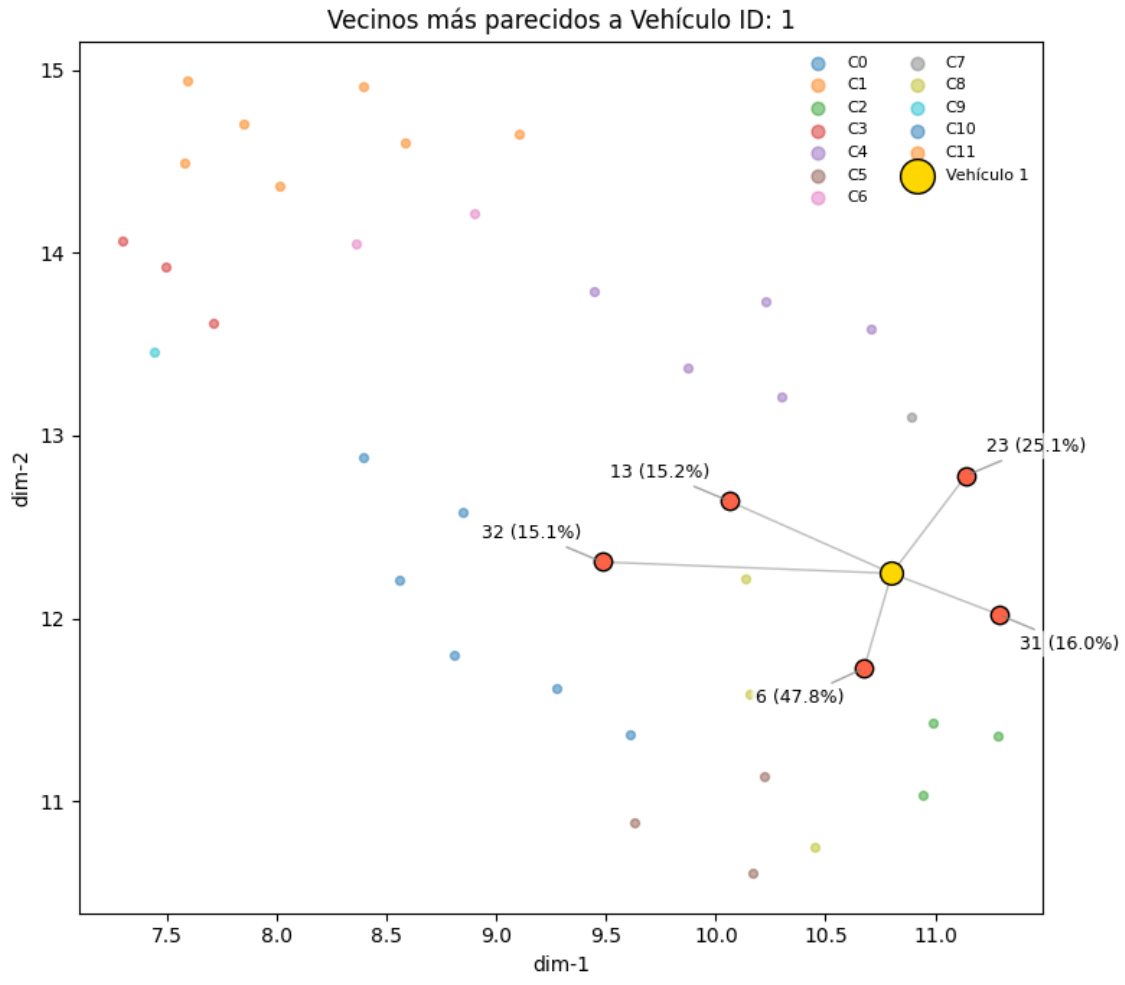
3.1.3 Análisis del Comportamiento en Video

Coincidiendo con lo que indican estas métricas, durante las pruebas observamos que el sistema logra Reidentificar con éxito a los mismos vehículos de forma consistente en la mayoría de los videos. Esto es especialmente claro en tomas donde los coches están a una distancia óptima de la cámara, permitiendo al modelo capturar suficientes detalles (como la forma de la carrocería, logos, o golpes) para crear una firma única de cada uno.

Para observar a más detalle las similitudes y diferencias entre los diferentes tipos de vehículos detectados, se realizó una proyección 2D del espacio de latencia de los embeddings, de manera que gráficamente se pueden relacionar los vehículos.

Figura 3.10

Proyección 2D del espacio latente de los embeddings, centrada en el vehículo ID = 1



La figura muestra cada vehículo como un punto obtenido al proyectar su embedding a dos dimensiones con UMAP, para ver cómo se agrupan por similitud porque tiende a preservar vecindarios del espacio original, de ahí que los puntos similares aparezcan cercanos. Los colores de C0 a C11 corresponden a agrupamientos no supervisados con k-means sobre los embeddings L2-normalizados, usados aquí solo para dar contexto visual a la distribución. El punto amarillo marca el vehículo consultado y las líneas grises enlazan sus cinco vecinos más parecidos medidos en el espacio original mediante similitud coseno (el porcentaje mostrado es la similitud; es una medida de parecido, no una probabilidad). Para fijar un número de grupos razonable, se evaluó un rango de K y se eligió el que maximizó el silhouette score calculado con distancia coseno en los embeddings originales, que compara qué tan cercano queda cada punto a su grupo frente a grupos vecinos y valores más altos implican una separación más clara.

Sin embargo, como es natural en cualquier sistema de visión por computadora, enfrentamos desafíos. En algunos videos específicos donde los vehículos se encuentran muy lejos de la cámara, el rendimiento puede disminuir. La razón es comprensible: a mayor distancia, los vehículos se ven más pequeños y con menos detalles discernibles, lo que dificulta que el modelo encuentre características suficientes para distinguirlos con total certeza.

Para pruebas en general, se ha determinado un rango óptimo:

- **Rango Óptimo:** El sistema opera con la máxima confiabilidad (Rank-1 > 85%) cuando los vehículos tienen un tamaño superior a 10,000 píxeles² (equivalente a un bounding box de aproximadamente 100x100 píxeles). En las tomas obtenidas de las cámaras de ESPOL, se traduce en una distancia menor a 10 metros. En este rango, el extractor de características (ViT) puede discernir detalles finos como logos, daños en la carrocería y características específicas de la parrilla.

Capítulo 4

4.1 Conclusiones y recomendaciones

Tras el desarrollo e implementación de la propuesta, que incluyó la integración de YOLOv8 para la detección, ByteTrack para el seguimiento y un modelo ViT para la extracción de características, junto con diversas mejoras técnicas, se obtuvieron los siguientes resultados clave que permiten considerar los siguientes puntos.

4.1.1 Conclusiones

1. **Respecto al Objetivo General:** Se logró cumplir con el objetivo propuesto. El sistema desarrollado integra detección automática de vehículos, seguimiento multi-cámara y reidentificación a través de embeddings discriminativos. Los resultados experimentales confirman su eficacia: se alcanzó un 94,70% en Rank-1 y un 77,96% en mAP, lo cual demuestra que la solución es capaz de identificar de manera consistente a los mismos vehículos en distintos escenarios y cámaras.
2. **Respecto al Objetivo Específico 1:** Se cumplió el desarrollo de un algoritmo de detección automática de vehículos mediante YOLOv8. Las pruebas realizadas tanto en el dataset VeRi-776 como en videos reales del campus ESPOL mostraron que el detector identifica correctamente los vehículos en las secuencias evaluadas.
3. **Respecto al Objetivo Específico 2:** El sistema implementó un codificador de imágenes basado en Vision Transformer (ViT) para la extracción de características discriminativas de cada vehículo. La generación de embeddings permitió representar a los vehículos en un espacio latente donde se observó agrupamiento por similitud (Figura 3.2.6), validando mediante el cálculo de silhoutte score. Esto demuestra que los embeddings no solo diferencian entre vehículos distintos, sino que también agrupan de forma consistente múltiples vistas de un mismo vehículo.
4. **Respecto al Objetivo Especifico 3:** Los experimentos de reidentificación mostraron resultados satisfactorios, por ejemplo, el vehículo con ID 5 fue correctamente

reidentificado en las 11 cámaras en las que aparecía, coincidiendo con el dataset de prueba. Esto valida la capacidad del sistema para vincular apariciones dispersas de un mismo vehículo, incluso con cambios de ángulos o condiciones parciales de visibilidad.

4.1.2 Recomendaciones y trabajos futuros

A partir de los resultados y limitaciones identificadas, se proponen las siguientes líneas de trabajo futuro:

- **Mejora del desempeño en escenarios de larga distancia:** La principal debilidad encontrada radica en la reidentificación de vehículos muy pequeños o lejanos. Una recomendación clave es investigar e incorporar técnicas de super-resolución o modelos específicamente entrenados para reconocer objetos a baja resolución para mitigar este problema.
- **Incremento de la diversidad del dataset:** El modelo podría beneficiarse de ser entrenado y evaluado en conjuntos de datos más diversos que incluyan condiciones climáticas adversas (lluvia, nieve, niebla), horas nocturnas y una variedad más amplia de tipos de vehículos y ángulos de cámara. Esto mejoraría su generalización y robustez en entornos del mundo real.
- **Implementación de un sistema de ReID multicámara en tiempo real:** El siguiente paso natural es desarrollar una arquitectura centralizada que pueda agregar y consultar la galería de vectores provenientes de múltiples flujos de video de diferentes cámaras de forma simultánea y en tiempo real, creando un verdadero sistema de rastreo mediante cámara.

Referencias

[1] A. García, “Tras dos años, la videovigilancia con inteligencia artificial de Telconet aún enfrenta retos operativos en Guayaquil,” Primicias, 21-Nov-2024. [Online]. Available: https://www.primicias.ec/seguridad/municipio-camaras-videovigilancia-inteligencia-artificial-telconet-pendientes-retos-83767/. [Accessed: 22-Nov-2024].

[2] “Cámaras de Videovigilancia – Servicio Integrado de Seguridad ECU 911,” Gob.ec. [Online]. Available: https://www.ecu911.gob.ec/camaras-de-videovigilancia/. [Accessed: 22-Nov-2024].

[3] J. M. Martínez, "Trabajo Final de Grado: Detección y seguimiento de objetos," Universitat Politècnica de Catalunya, 2020. [Online]. Available: https://upcommons.upc.edu/bitstream/handle/2117/129222/memoria.pdf

[4] D. Gónzalez, M. Amaguaya, "Diseño e implementación de un sistema de reconocimiento de vehículos," Universidad Politécnica Salesiana, Trabajo de titulación, 2022. [Online]. Available: https://dspace.ups.edu.ec/bitstream/123456789/24168/1/UPS-GT004218.pdf

[5] D. Ameijeiras Sánchez, H. R. González Diez, y Y. Hernández Heredia, “Revisión de algoritmos de detección y seguimiento de objetos con redes profundas para videovigilancia inteligente”, Rev. Cuba. Cienc. Inform., vol. 14, núm. 3, pp. 165–195, 2020.

[6] V. Jiménez and T. Mauricio, “Evaluación de rendimiento de YOLOv5 y algoritmos de seguimiento en una Jetson Nano 2GB,” 2023.

[7] A. Pérez Pena, “Detección e identificación de persoas cun robot móbil equipado cun LiDAR 3D e cámaras RGBD,” 2024.

[8] N. Wojke, A. Bewley y D. Paulus, “Simple Online and Realtime Tracking with a Deep Association Metric,” *Proc. IEEE Int. Conf. Image Process.* 2017.

[9] N. Cohen y I. Klein, «Adaptive Kalman-Informed Transformer», arXiv, 2024, doi: 10.48550/ARXIV.2401.09987.

[10] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "ByteTrack: Multi-Object Tracking by Associating Every Detection Box," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 1–21.

[11] Z. Liu et al., "[2401.10643] A Comprehensive Survey on Deep-Learning-based Vehicle ReIDentification: Models, Data Sets and Challenges", https://arxiv.org/abs/2401.10643

[12] B. Lavi, M. Fatan, I. Ullah, "[1807.05284] Survey on Deep Learning Techniques for Person ReIDentification Task", https://arxiv.org/abs/1807.05284

[13] Saji Santander, D. A. (2019). Reconocimiento de montos manuscritos en cheques a través de modelos de detección de objetos basados en redes convolucionales.

[14] M. Shouji, C. Wei-shi, D. Bo, "[2401.06960] Transformer for Object ReIDentification: A Survey", https://arxiv.org/abs/2401.06960

[15] A. Kuruparan, W. Lin, "[2110.07933] Relation Preserving Triplet Mining for Stabilising the Triplet Loss in ReIDentification Systems", https://arxiv.org/abs/2110.07933

[16] “Using DeepSORT object tracker with YOLOv5,” Kaggle.com, 01-Jun-2023. [Online]. Available: https://www.kaggle.com/code/nityampareek/using-Deepsort-object-tracker-with-yolov5/notebook. [Accessed: 22-Nov-2024].

[17] “Driving Video object tracking,” Kaggle.com, 01-Apr-2024. [Online]. Available: https://www.kaggle.com/code/kirollosashraf/driving-video-object-tracking. [Accessed: 22-Nov-2024].

[18] S. Ren, K. He, R. Girshick, y J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks”, arXiv [cs.CV], 2015.

[19] Á. Ramajo Ballester, J. González Cepeda, J. M. Armingol Moreno, and A. de la Escalera Hueso, “Reidentificación de vehículos mediante técnicas de Deep Learning,” in XLIII Jornadas de Automática: libro de actas: 7, 8 y 9 de septiembre de 2022, Logroño (La Rioja), Servizo de Publicacións da UDC, 2022, pp. 1031–1039.

[20] Javier de Fuentes, Vehicle tracking and identification for mobility applications using computer vision. Upc.edu.

[21] Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., & Wang, X. (2022). ByteTrack: Multi-Object Tracking by Associating Every Detection Box. European Conference on Computer Vision (ECCV 2022). doi:10.1007/978-3-031-20047-2_1. Disponible en: https://www.ecva.net/papers/eccv_2022/papers_ECCV/papers/136820001.pdf

[22] Y. Yao, J. Zhang, F. Shen, and H. T. Shen, "Vision Transformer for Vehicle ReIDentification," in Proc. IEEE Int. Conf. Multimedia Expo (ICME), 2022.

[23] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a Similarity Metric Discriminatively, with Application to Face Verification," in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), 2005, vol. 1, pp. 539–546.

[24] A. Dosovitskiy et al., “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2021.

[25] S. He et al., "TransReID: Transformer-based Object ReIDentification," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2021, pp. 15013–15022.

[26] J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL-HLT, 2019, pp. 4171–4186.

[27] A. Hermans, L. Beyer, and B. Leibe, "In Defense of the Triplet Loss for Person ReIDentification," arXiv preprint arXiv:1703.07737, 2017.

[28] Luo, H., et al. Bag of Tricks and A Strong Baseline for Deep Person ReIDentification. CVPRW 2019. (BNNeck, label smoothing, PK, prácticas de entrenamiento).

[29] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Y. Wang, and Q. Tian, “Circle Loss: A Unified Perspective of Pair Similarity Optimization,” in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2020, pp. 6398–6407.