.UBAfiuba
FACULTAD DE INGENIERÍA

Automatización de censos ciclistas mediante visión por computadora

Arq. Juan Cruz Ferreyra

Carrera de Especialización en Inteligencia Artificial

Director: Mg. Ing. Seyed Khodadad Pakdaman (FIUBA)

Jurados:

Mg. Lic. Yanina Giorgis (EMR)
Ing. Bruno Masoller (FIUBA)
Mg. Ing. Jonathan Cagua (FIUBA)

Ciudad de Rosario, octubre de 2025

Resumen

En esta memoria se presenta el desarrollo de un sistema de visión por computadora para automatizar los censos ciclistas en Rosario, Santa Fe. El sistema reemplaza el relevamiento manual realizado por el Ente de la Movilidad, permitiendo detectar, seguir y caracterizar ciclistas utilizando cámaras de videovigilancia. El trabajo integra técnicas de detección de objetos y clasificación de imágenes, aplicando conocimientos de visión por computadora y aprendizaje profundo.

Índice general

Resumen I

1. Introducción general 1
- 1.1. Contexto . . . 1
- 1.2. Motivación . . . 2
- 1.3. Estado del arte . . . 3
- 1.4. Alcance y objetivos . . . 5
  - 1.4.1. Objetivos específicos . . . 5
  - 1.4.2. Alcances técnicos . . . 5
  - 1.4.3. Límites del trabajo . . . 5

2. Introducción específica 7
- 2.1. Visión por computadora . . . 7
- 2.2. Visión por computadora clásica . . . 7
  - 2.2.1. SIFT y emparejamiento de keypoints . . . 8
- 2.3. Detección de objetos . . . 8
  - 2.3.1. YOLOv8 . . . 8
  - 2.3.2. RF-DETR . . . 8
- 2.4. Clasificación demográfica . . . 9
  - 2.4.1. EfficientNet . . . 9
  - 2.4.2. ResNet50 . . . 9
- 2.5. Transfer learning . . . 9
- 2.6. Data augmentation . . . 10
- 2.7. Requerimientos de software . . . 10

3. Diseño e implementación 11
- 3.1. Preparación de los datos para detección . . . 11
  - 3.1.1. Obtención de los datos . . . 11
  - 3.1.2. Separación de fotogramas . . . 13
  - 3.1.3. Filtro de fotogramas redundantes . . . 14
  - 3.1.4. Etiquetado de datos para la tarea de detección . . . 16
- 3.2. Entrenamiento del modelo de detección . . . 18
  - 3.2.1. Aumentación de datos . . . 18
  - 3.2.2. Ingesta de datos . . . 19
  - 3.2.3. Entrenamiento de los modelos YOLOv8 . . . 20
  - 3.2.4. Entrenamiento de los modelos RF-DETR . . . 21
- 3.3. Preparación de los datos para clasificación . . . 22
  - 3.3.1. Obtención de los datos . . . 22
- 3.4. Entrenamiento del modelo de clasificación . . . 23
  - 3.4.1. Aumentación de datos . . . 24
  - 3.4.2. Ingesta de datos . . . 25
  - 3.4.3. Búsqueda de hiperparámetros . . . 25

3.4.4. Entrenamiento de los modelos EfficientNet y Resnet 25

4. Ensayos y resultados **29**
4.1. Evaluación de los modelos de detección 29
4.1.1. Características de los datos 29
4.1.2. Evaluación general de los modelos 30
4.1.3. Evaluación específica de los modelos 31
4.2. Evaluación de los modelos de clasificación 32
4.2.1. Características de los datos 33
4.2.2. Ajuste del umbral de decisión 34
4.2.3. Evaluación de los modelos 35

5. Conclusiones **39**
5.1. Conclusiones generales 39
5.2. Próximos pasos 40

Bibliografía **43**

Índice de figuras

1.1. Censo realizado manualmente. 2

3.1. Muestreo horario de una cámara específica 12
3.2. Ubicación de cámaras de monitoreo 12
3.3. Detección de keypoints con SIFT 15
3.4. Emparejamiento con FLANN matcher 15
3.5. Embeddings visualizados con t-SNE 17
3.6. Transformaciones aplicadas durante la etapa de augmentation . . . 19
3.7. Curvas de entrenamiento de YOLO 20
3.8. Curvas de entrenamiento de RF'DETR 21
3.9. Ejemplos del conjunto de entrenamiento para clasificación 22
3.10. Ejemplos de aumentación en clasificación 24
3.11. Curvas de entrenamiento de EfficientNet 26
3.12. Curvas de entrenamiento de ResNet 27

4.1. Ejemplo del conjunto de evaluación para detección 30
4.2. Evaluación de la performance y latencia de los modelos de detección 30
4.3. Evaluación de la performance por clase de los modelos de detección 31
4.4. Evaluación de la performance en las clases relevantes de los modelos de detección 32
4.5. Ejemplos del conjunto de evaluación para clasificación 34
4.6. Resultados en el conjunto de evaluación para el modelo EfficientNet-b0 35
4.7. Resultados en el conjunto de evaluación para el modelo EfficientNet-b3 36
4.8. Resultados en el conjunto de evaluación para el modelo ResNet-50 36
4.9. Resultados en el conjunto de evaluación para el modelo ResNet-101 37

Índice de tablas

1.1. Comparativa YOLOv8 vs RF-DETR 4

3.1. Distribución de imágenes utilizadas 17
3.2. Distribución de imágenes utilizadas 23

4.1. Umbrales de decisión para cada modelo de clasificación 34

Capítulo 1

Introducción general

En el presente capítulo se introduce el problema que motiva el desarrollo de este trabajo, enmarcado en la necesidad de modernizar los métodos de recolección de datos sobre movilidad ciclista en la ciudad de Rosario. Se presentan el contexto actual y las limitaciones del sistema de censos manuales, así como la oportunidad de aplicar técnicas de visión por computadora para su automatización. Además, se revisa el estado del arte en soluciones similares, y se delimitan los objetivos y el alcance de la propuesta.

1.1. Contexto

El Ente de la Movilidad de Rosario [1] es un organismo autárquico descentralizado, tanto administrativa como financieramente, responsable de gestionar la movilidad urbana en todos sus modos. Entre sus competencias se incluyen el transporte público masivo, individual y especial; el transporte no motorizado; el uso privado de vehículos; y otros servicios conexos relacionados con la circulación en la ciudad. El Ente se caracteriza por su alto perfil técnico y por estar conformado por un equipo multidisciplinario que aborda de manera integral los desafíos asociados a la movilidad, promoviendo políticas activas orientadas a la sustentabilidad.

En este marco, se alienta una movilidad no motorizada, pública y colectiva. Esto implica fomentar prácticas más saludables y menos contaminantes, una utilización racional del espacio público y el diseño de servicios accesibles a la mayor cantidad de habitantes con la menor cantidad de recursos posibles. La promoción de medios de transporte sustentables, como la bicicleta, forma parte esencial de esta estrategia.

Como parte de sus tareas, el Ente de la Movilidad lleva a cabo censos ciclistas manuales en puntos estratégicos de la ciudad de Rosario. Estos relevamientos son realizados por trabajadores de la institución, quienes observan durante intervalos de 15 minutos el flujo vehicular en intersecciones seleccionadas y lo registran en papel, como se muestra en la figura 1.1. En cada intervalo se registran conteos discriminados por categorías: automóviles, transporte urbano de pasajeros (TUP), motocicletas, bicicletas y monopatines. En el caso de bicicletas y monopatines, se detalla si circulan por ciclovías o por la calzada junto a vehículos motorizados. Además, se distingue si se trata de bicicletas personales o públicas, y se anota la presencia de carga o de niños como pasajeros. Finalmente, se intenta identificar visualmente el género y el grupo etario de los ciclistas.

FIGURA 1.1. Censo realizado manualmente.

Los censos ciclistas representan una herramienta central para la toma de decisiones en materia de movilidad urbana. Al ofrecer información sistemática sobre el uso de la bicicleta y otros medios no motorizados, permiten evaluar el desempeño de la infraestructura existente, identificar zonas de alta circulación para priorizar mejoras, y diseñar políticas públicas fundamentadas en evidencia empírica. Además, la recolección periódica y consistente de estos datos permite construir series temporales que son fundamentales para monitorear la evolución del sistema de transporte urbano, identificar tendencias, y evaluar el impacto de intervenciones realizadas a lo largo del tiempo.

Este método, aunque valioso, presenta importantes limitaciones. En primer lugar, demanda una gran cantidad de recursos humanos, ya sea mediante la contratación temporal de censistas o la reasignación de personal calificado a tareas que no aprovechan sus competencias técnicas. En segundo lugar, la metodología manual dificulta la verificación posterior de los datos recolectados, especialmente en casos atípicos o inconsistentes, donde no es posible revisar la escena para confirmar su validez. También se encuentra sujeta a errores humanos y variaciones en la interpretación subjetiva, en particular al identificar características demográficas como el género. A estas limitaciones se suman problemas derivados del contexto de inseguridad urbana: en algunos puntos de la ciudad no es viable desplegar personal sin poner en riesgo su integridad física, lo que impide obtener datos representativos de toda la red vial en diferentes momentos del día.

En este contexto, surge la necesidad de explorar alternativas tecnológicas que permitan automatizar los censos ciclistas al utilizar recursos ya disponibles, como las cámaras de videovigilancia instaladas en la ciudad. La disponibilidad de datos confiables, precisos y frecuentes sobre la movilidad ciclista es fundamental para planificar infraestructura adecuada, mejorar la seguridad vial, evaluar políticas públicas y fomentar el uso de medios de transporte sustentables. La implementación de herramientas basadas en visión por computadora constituye, por tanto, una oportunidad concreta para modernizar el sistema de recolección de datos y avanzar hacia una gestión más inteligente de la movilidad urbana.

1.2. Motivación

La ciudad de Rosario cuenta con una amplia red de cámaras de videovigilancia distribuidas estratégicamente con el objetivo de mejorar la seguridad vial y ciudadana. Estos dispositivos, instalados en 70 puntos de control, forman parte de un sistema integrado que combina la lectura automática de patentes, la detección

de infracciones mediante inteligencia artificial, y el monitoreo continuo por parte del Centro Integrado de Operaciones Rosario (CIOR) [2] y el sistema 911. Si bien cada una de estas cámaras fue diseñada para cumplir una función específica en tiempo real, su potencial no ha sido completamente explotado, especialmente en lo que refiere a su uso diferido para tareas de análisis y planificación urbana.

Particularmente, los videos almacenados por las cámaras de monitoreo que graban las 24 horas y mantienen un registro histórico de la circulación en la ciudad representan una fuente valiosa de información para extraer datos sobre movilidad mediante técnicas de visión por computadora. A diferencia de otras cámaras que captan información sensible y cuyo acceso está restringido, estas grabaciones son aprovechables para fines analíticos sin comprometer la privacidad de los ciudadanos, siempre que se respeten los protocolos institucionales correspondientes.

El aprovechamiento de estos recursos ya disponibles permitiría superar muchas de las limitaciones del sistema de censos manuales descritas anteriormente en la sección 1.1. Automatizar la recolección de datos no solo libera recursos humanos y evita exponer personal a situaciones de riesgo, sino que habilita la posibilidad de cubrir una mayor cantidad de puntos de observación y obtener datos en distintos momentos del día. Esto se traduce en una información más rica, verificable y representativa, que habilita un monitoreo más fino y sostenido en el tiempo.

El impacto esperado de este enfoque es significativo: un incremento en la cantidad de datos, con mayor frecuencia y a menor costo, lo que habilita la construcción de series temporales robustas para detectar patrones, anomalías o tendencias en la movilidad ciclista. Esto resulta esencial tanto para anticipar conflictos como para evaluar intervenciones ya realizadas, por ejemplo, midiendo los efectos concretos de una obra sobre el flujo ciclista antes y después de su ejecución. Así, la toma de decisiones urbanas se vuelve más ágil, basada en evidencia y alineada con los principios de movilidad sustentable.

Finalmente, la solución propuesta se alinea con los desafíos y necesidades de la gestión pública. Al basarse en infraestructura ya desplegada y herramientas de software libre o reutilizable, permite escalar la metodología sin requerir grandes inversiones, replicarla en otras áreas de la ciudad o incluso en otras localidades con sistemas similares. Se trata, por lo tanto, de una solución técnica con fuerte orientación institucional, pensada para integrarse dentro del ecosistema público con impacto concreto en la gestión del transporte urbano.

1.3. Estado del arte

El análisis de la movilidad urbana ha sido históricamente una tarea llevada a cabo mediante metodologías manuales, como conteos presenciales o encuestas en puntos estratégicos. Si bien estos métodos permiten obtener información contextual directa, su carácter intensivo en recursos humanos, limitado en cobertura y poco escalable ha impulsado el desarrollo de alternativas tecnológicas [3]. Entre ellas, se encuentran soluciones basadas en sensores físicos instalados en la infraestructura urbana para registrar el paso de vehículos o peatones [4] [5]. Sin embargo, estos sensores tienen limitaciones en cuanto a la clasificación de los actores viales y su instalación requiere obras civiles, lo que restringe su despliegue masivo en ciudades con presupuestos acotados.

En los últimos años, el avance en técnicas de visión por computadora ha permitido reemplazar o complementar estos métodos mediante el análisis automático de imágenes y videos [6] [7] [8]. Aplicaciones exitosas se han implementado para el conteo vehicular, la detección de congestión, el análisis de comportamiento peatonal, la evaluación del uso del espacio público y, más recientemente, el monitoreo de movilidad ciclista [9] [10]. Estas soluciones se basan en modelos de detección de objetos, seguimiento y clasificación, entrenados con grandes volúmenes de datos y optimizados para operar en entornos urbanos complejos.

Para la detección de objetos en video, se dispone actualmente de múltiples arquitecturas entrenadas y validadas en entornos urbanos. Entre ellas se destacan los modelos de la familia YOLO (*You Only Look Once*) [11], ampliamente utilizados por su equilibrio entre precisión y velocidad, y RF-DETR [12], una arquitectura reciente basada en transformers desarrollada por Roboflow, que ha mostrado resultados competitivos y está disponible bajo licencia Apache 2.0, lo que la hace especialmente adecuada para contextos académicos e institucionales. Una comparación técnica entre ambos modelos se presenta en la tabla 1.1.

TABLA 1.1. Comparativa técnica entre los modelos de detección YOLOv8 y RF-DETR según arquitectura, licencia, desempeño y requisitos de uso.

Criterio	**YOLOv8**	**RF-DETR**
Arquitectura	CNN	Transformer
Licencia	GPL-3.0 (Ultralytics)	Apache 2.0
mAP (COCO val)	∼56 (YOLOv8m)	∼52 (según benchmarks)
Velocidad de inferencia	∼70 FPS (GPU media)	∼20 FPS (GPU similar)
Tamaño del modelo	35–200 MB (según variante)	∼220 MB

Para el seguimiento de objetos a lo largo del tiempo, algoritmos como DeepSort [13] o ByteTrack [14] permiten asociar detecciones entre cuadros consecutivos, manteniendo la identidad de los ciclistas incluso en situaciones de oclusión o cruce. Por otro lado, la caracterización demográfica puede lograrse mediante modelos de clasificación entrenados para estimar género y grupo etario a partir de imágenes faciales o corporales, al integrar información adicional valiosa para el análisis urbano. La combinación de las técnicas de detección, seguimiento y clasificación, conforma una base sólida y extensamente validada sobre la que se construye la solución propuesta en este trabajo.

Existen ejemplos documentados de uso de estas tecnologías en distintas ciudades del mundo para el monitoreo automatizado del tránsito [15] [16] [17]. Algunas iniciativas se centran en el conteo y seguimiento de vehículos en tiempo real para la gestión del flujo vial, mientras que otras, más enfocadas en la movilidad activa, han utilizado visión por computadora para evaluar el uso de ciclovías o detectar conflictos entre distintos modos de transporte. No obstante, muchas de estas soluciones son propietarias, de difícil acceso o desarrolladas en contextos tecnológicos y urbanos distintos al de Rosario.

En contraste, la propuesta desarrollada en este trabajo se basa en el aprovechamiento de infraestructura ya existente y utiliza herramientas de código abierto que permiten una implementación flexible y sostenible. Además, integra en un mismo flujo de procesamiento la detección de ciclistas, el análisis demográfico y

el seguimiento de trayectorias, para aportar un enfoque integral al problema. Esta combinación de factores convierte al sistema propuesto en una solución innovadora, adaptada al contexto local y con alto potencial de réplica en otras ciudades con recursos y desafíos similares.

1.4. Alcance y objetivos

El presente trabajo tiene como objetivo general el desarrollo de un sistema basado en visión por computadora para la automatización de censos ciclistas a partir de videos capturados por las cámaras de videovigilancia de la ciudad de Rosario. Esta automatización busca mejorar la eficiencia, seguridad y calidad de los datos recolectados, para facilitar su integración con los procesos ya establecidos por el Ente de la Movilidad.

1.4.1. Objetivos específicos

Para alcanzar este objetivo general, se establecen los siguientes objetivos específicos:

- Detectar bicicletas y personas en videos de vigilancia mediante modelos de detección entrenados o adaptados al entorno urbano local.
- Implementar una heurística que permita asociar personas con bicicletas para conformar la clase de ciclistas.
- Inferir características demográficas básicas (género y grupo etario) de los ciclistas detectados mediante modelos de clasificación visual.
- Analizar las trayectorias y direcciones de circulación de los ciclistas para extraer patrones de movimiento dentro del campo visual.
- Generar salidas que incluyan videos anotados, imágenes de ciclistas etiquetadas, y planillas de conteo compatibles con los formatos de registro del Ente de la Movilidad.
- Elaborar documentación técnica detallada y establecer un plan de evaluación para medir el desempeño y la efectividad del sistema.

1.4.2. Alcances técnicos

El sistema está diseñado para procesar videos grabados, no en tiempo real, al utilizar datos provenientes de las cámaras de monitoreo ya instaladas en la ciudad. La solución se implementa sobre hardware estándar y no requiere modificaciones físicas en la infraestructura existente. Los modelos utilizados han sido adaptados mediante técnicas de aprendizaje por transferencia, lo que permite ajustar redes preentrenadas a las condiciones particulares de las cámaras locales (ángulo, iluminación, calidad de imagen, etc.). La solución se focaliza exclusivamente en actores de la movilidad no motorizada, especialmente ciclistas, y no contempla en esta etapa otros tipos de usuarios viales.

1.4.3. Límites del trabajo

El trabajo no incluye los siguientes aspectos:

- Desarrollo o instalación de hardware específico para la captura de imágenes o videos.
- Procesamiento en tiempo real ni su integración directa con sistemas de vigilancia activa o gestión de tráfico.
- Predicción futura de comportamiento de ciclistas ni modelado estadístico de escenarios hipotéticos.
- Evaluación o diagnóstico del estado físico del equipamiento urbano, como ciclovías o señalización.

Capítulo 2

Introducción específica

En el presente capítulo se describen los principales componentes técnicos que integran el sistema de automatización de censos ciclistas. Se introduce el concepto de visión por computadora y su aplicación en el análisis de la movilidad urbana. Se detallan las tareas de detección de objetos, seguimiento de trayectorias y clasificación demográfica. Se presentan las arquitecturas seleccionadas para cada módulo junto con la inclusión de modelos basados en redes convolucionales y transformers, y se explica el uso de técnicas de transfer learning como estrategia para adaptar modelos preentrenados al contexto local. Finalmente, se enumeran las herramientas y bibliotecas empleadas, y se detaca su función dentro del flujo general de procesamiento del sistema.

2.1. Visión por computadora

La visión por computadora es una rama de la inteligencia artificial que tiene como objetivo permitir que las computadoras interpreten y comprendan imágenes y videos del mundo real, de manera similar a como lo hace el ojo humano. A través del uso de modelos matemáticos y redes neuronales profundas, esta disciplina permite identificar objetos, seguir su movimiento y extraer información relevante a partir de datos visuales. [18]

En el ámbito de la movilidad urbana, la visión por computadora se ha convertido en una herramienta fundamental para el monitoreo del tránsito, el análisis del uso del espacio público y la automatización de censos de distintos modos de transporte [19] [20]. Su capacidad para operar sobre imágenes capturadas por cámaras ya instaladas en la vía pública permite extender el alcance de las mediciones sin necesidad de nueva infraestructura, lo que la vuelve especialmente atractiva para su implementación en contextos municipales.

2.2. Visión por computadora clásica

Dentro del campo de la visión por computadora, se denomina enfoque clásico al conjunto de técnicas basadas en algoritmos deterministas, previos a la adopción generalizada de modelos de aprendizaje profundo [21]. Estas herramientas, aunque en muchos casos reemplazadas por redes neuronales, continúan siendo útiles para resolver tareas específicas con bajo costo computacional y buena interpretabilidad. En el marco de este trabajo, se emplearon varios métodos clásicos como soporte para las tareas principales de detección, seguimiento y clasificación.

2.2.1. SIFT y emparejamiento de keypoints

Uno de los algoritmos utilizados fue SIFT (*Scale-Invariant Feature Transform*) [22], una técnica diseñada para detectar puntos de interés en una imagen de manera robusta ante cambios de escala, rotación o iluminación. Cada punto de interés o keypoint se describe mediante un vector característico que permite su comparación con otros puntos similares en imágenes distintas. Esta propiedad facilita el emparejamiento entre dos imágenes, para identificar regiones que contienen la misma información visual. En este trabajo, se empleó esta herramienta durante el proceso de selección de fotogramas para el etiquetado manual. La comparación entre imágenes consecutivas mediante el emparejamiento de keypoints [23] permitió detectar fotogramas visualmente redundantes y conservar únicamente aquellos más representativos para optimizar el proceso de anotación.

2.3. Detección de objetos

La detección de objetos es una tarea clave en visión por computadora que permite localizar e identificar múltiples instancias de distintas clases dentro de una imagen o video [24]. A diferencia de la clasificación, que asigna una única etiqueta global, este enfoque devuelve bounding boxes junto con las categorías de los objetos detectados. En el marco de este trabajo, la detección constituyó el primer paso fundamental para identificar bicicletas, personas y otros vehículos en los videos de las cámaras de videovigilancia, con una influencia directa en la calidad de los módulos posteriores de seguimiento, conteo y clasificación. Para esta tarea se seleccionaron dos arquitecturas de referencia en el estado del arte: YOLOv8 y RF-DETR.

2.3.1. YOLOv8

YOLOv8 (*You Only Look Once*) [11] es parte de una familia de modelos de detección de objetos en tiempo real reconocida por su eficiencia y precisión. Su funcionamiento consiste en dividir la imagen en una cuadrícula y predecir simultáneamente las coordenadas, clases y probabilidades de los objetos presentes. Ofrece variantes de distintos tamaños que permiten su adaptación a diversas capacidades de hardware. En este trabajo se utilizó una versión intermedia, ajustada mediante técnicas de transfer learning a las condiciones particulares de las cámaras de Rosario.

2.3.2. RF-DETR

RF-DETR [12] es una implementación de código abierto basada en la arquitectura DETR (*DEtection TRansformer*), desarrollada por Roboflow. A diferencia de los modelos basados en redes convolucionales, RF-DETR utiliza mecanismos de atención propios de los transformers, lo que le permite capturar relaciones espaciales globales dentro de una imagen, siendo especialmente útil en escenas complejas o con alta densidad de objetos. Aunque más demandante en términos computacionales, ofrece un rendimiento de precisión comparable al de las CNN avanzadas. Fue seleccionado en este trabajo por su licencia permisiva (Apache 2.0) y por su valor como enfoque complementario al de YOLOv8 dentro del mismo flujo de trabajo.

2.4. Clasificación demográfica

La clasificación de imágenes es una tarea fundamental en visión por computadora que consiste en asignar una o más etiquetas a una imagen, con el objetivo de categorizar su contenido en función de patrones visuales detectables [25]. A diferencia de la detección de objetos, que localiza instancias dentro de una escena, la clasificación permite caracterizar visualmente esas instancias con atributos específicos. En este proyecto, esta técnica se utiliza para realizar una clasificación demográfica de los ciclistas identificados, con el objetivo de estimar su género a partir de imágenes tomadas desde una perspectiva elevada. Dado que no se dispone de vistas frontales ni detalles faciales, la predicción se basa en rasgos generales como la vestimenta, la postura corporal y el contexto visual.

2.4.1. EfficientNet

EfficientNet [26] es una familia de redes convolucionales modernas que logra una alta precisión con un bajo número de parámetros, lo que la convierte en una opción ideal para tareas con recursos computacionales limitados. En particular, la variante EfficientNet-B0 permitió obtener buenos resultados incluso con volúmenes moderados de datos, gracias a su arquitectura optimizada. En este trabajo se propuso su uso como modelo base para la clasificación de género, debido a su capacidad para generalizar bien en contextos variados con vistas parciales o no frontales.

2.4.2. ResNet50

ResNet50 [27] es una arquitectura clásica ampliamente utilizada en tareas de clasificación de imágenes. Su profundidad y estabilidad la convierten en una opción confiable para trabajos donde se requiere una primera aproximación robusta y fácil de ajustar. Al ser compatible con la mayoría de los frameworks y contar con pesos preentrenados disponibles, resultó útil para experimentar con distintos niveles de ajuste fino (fine-tuning) sobre el conjunto de ciclistas extraído de los videos. Su uso en este trabajo permitió establecer una línea base sólida para comparar el rendimiento de modelos más modernos como EfficientNet.

2.5. Transfer learning

El transfer learning es una técnica que permite reutilizar modelos previamente entrenados en grandes conjuntos de datos generales y ajustarlos a tareas específicas con menor cantidad de datos [28]. En lugar de entrenar desde cero, se parte de pesos ya optimizados y se realiza un ajuste fino (fine-tuning), lo que reduce el tiempo de entrenamiento y mejora la capacidad del modelo para generalizar con datos limitados. Algunos dataset ampliamente utilizados para el preentrenamiento de modelos son ImageNet [29] y COCO [30].

En este trabajo se aplicó aprendizaje por transferencia tanto en los modelos de detección (YOLOv8 y RF-DETR) como en los de clasificación demográfica (EfficientNet y ResNet50). Dado que los datasets originales contienen imágenes urbanas y de personas en diferentes contextos, existe un solapamiento visual relevante con las escenas capturadas por las cámaras de videovigilancia de Rosario. Esto hace

que el uso de modelos preentrenados no solo sea eficiente, sino especialmente adecuado para las condiciones reales del trabajo.

2.6. Data augmentation

La técnica de data augmentation consiste en generar nuevas instancias de entrenamiento a partir de transformaciones aplicadas sobre los datos originales, con el objetivo de mejorar la capacidad de generalización del modelo y reducir el sobreajuste [31]. Estas transformaciones pueden incluir rotaciones, cambios de escala, recortes, alteraciones de brillo o contraste, entre otras. El data augmentation puede implementarse de forma offline, al generar un conjunto aumentado antes del entrenamiento, o de manera online, al aplicar las transformaciones en tiempo real durante el proceso de entrenamiento. En este trabajo se optó por la modalidad online, ya que permite mayor flexibilidad, ahorro de espacio de almacenamiento y una mayor variabilidad en cada época del entrenamiento.

2.7. Requerimientos de software

El desarrollo del sistema se apoyó en diversas bibliotecas y frameworks especializados en visión por computadora y aprendizaje profundo. A continuación, se enumeran las principales herramientas utilizadas, junto con una breve descripción de su función en el trabajo:

- OpenCV: utilizada para el preprocesamiento de imágenes, transformaciones geométricas y operaciones básicas sobre fotogramas [32].
- PyTorch: framework empleado para la carga, modificación y entrenamiento por transferencia de modelos de clasificación como EfficientNet y ResNet50 [33].
- Ultralytics: implementación moderna de los modelos YOLO, utilizada para la descarga, ajuste y reentrenamiento de arquitecturas YOLOv8 [34].
- Roboflow RF-DETR: repositorio específico para la utilización y entrenamiento del modelo RF-DETR, basado en arquitecturas transformer [35].
- Roboflow Supervision: biblioteca auxiliar para la orquestación de tareas de visión por computadora, manejo de videos e imágenes, creación de datasets y visualización [36].

Capítulo 3

Diseño e implementación

En este capítulo se presentan las etapas de diseño e implementación del sistema desarrollado, e incluye desde la preparación de los datos hasta la integración de sus distintos componentes. Se describe el proceso de obtención y selección de imágenes a partir de videos urbanos, así como el etiquetado manual realizado para las tareas de detección. Se detallan los modelos utilizados para detección de objetos, junto con la metodología de entrenamiento y evaluación adoptada. Además, se describe la utilización de dichos modelos para la extracción de imágenes de ciclistas, empleadas en la construcción de los conjuntos de datos y en la evaluación de modelos de clasificación por género. Se presentan los modelos empleados para esta segunda tarea, junto con su correspondiente proceso de entrenamiento y validación.

3.1. Preparación de los datos para detección

Esta sección describe el proceso de preparación de los datos utilizados para entrenar y evaluar los modelos del trabajo. Incluye la descripción de las tareas para la obtención de los videos, la extracción y selección de fotogramas, y la generación de las anotaciones necesarias para el entrenamiento y evaluación de los modelos de detección.

3.1.1. Obtención de los datos

El conjunto de datos utilizado en este trabajo proviene de grabaciones obtenidas por cámaras de videovigilancia urbana, administradas por el Ente de la Movilidad de la ciudad de Rosario. Mediante un pedido formal realizado por el equipo de trabajo, se obtuvo acceso a grabaciones provenientes de doce cámaras distribuidas en distintos puntos de la ciudad. El material fue entregado por el Ente exclusivamente para su utilización en el marco de este trabajo académico, con fines analíticos internos de la institución y respetando los protocolos institucionales correspondientes.

De cada una de las cámaras se recibieron tres videos de aproximadamente dos horas de duración, registrados en distintos momentos del día con el objetivo de capturar variedad en las condiciones de iluminación, el flujo vehicular y los modos de transporte presentes en la imagen. Esta estrategia permitió construir un conjunto de imágenes representativo de las distintas situaciones que pueden encontrarse en el espacio urbano. La figura 3.1 ejemplifica tres fotogramas obtenidos de la misma cámara en los que se observa la diversidad de características.

FIGURA 3.1. Ejemplos de fotogramas extraídos de la cámara ubicada en Av. Pellegrini y Corrientes, tomados en distintos momentos del día.

La selección de cámaras priorizó aquellas ubicadas sobre arterias principales y en sectores que cuentan con ciclovías, ya que se trata de zonas especialmente relevantes para el análisis de la movilidad ciclista. Además, se procuró una distribución geográfica diversa para abarcar distintos sectores de la ciudad y sus características de tránsito. La figura 3.2 muestra la localización de cada una de las cámaras utilizadas en este trabajo.

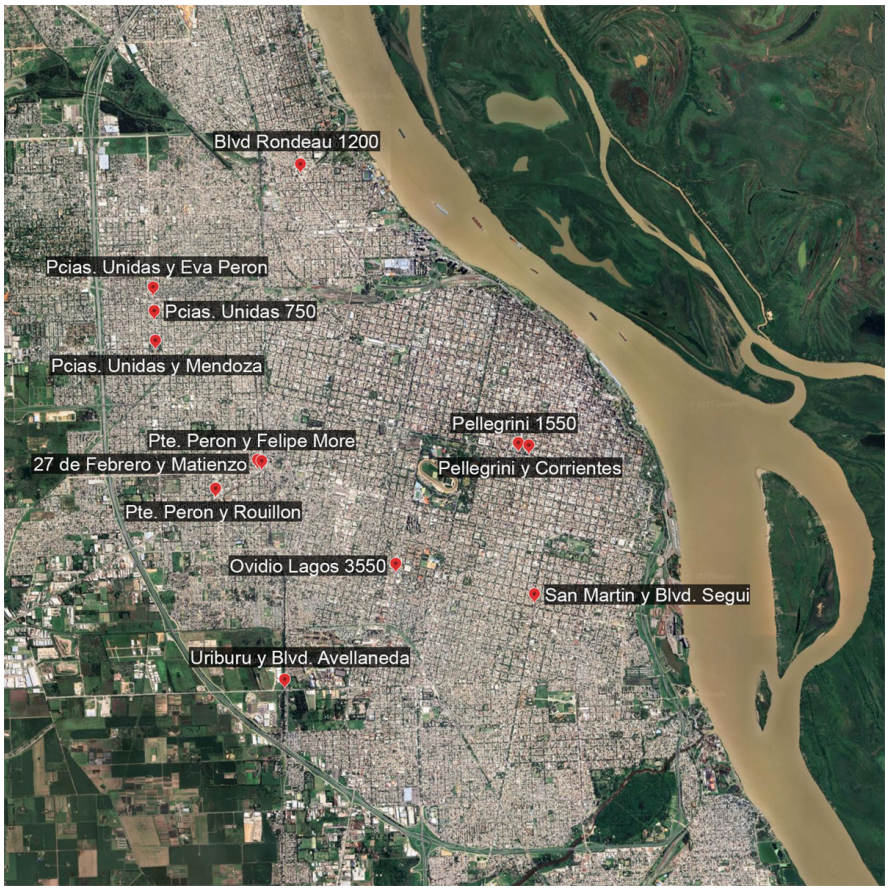

FIGURA 3.2. Distribución geográfica de las cámaras de monitoreo utilizadas en este trabajo.

Los videos fueron entregados en formatos estandarizados, mayoritariamente en `.avi`, y presentan en su nomenclatura la ubicación geográfica de la cámara. Además, cada grabación incluye un reloj digital visible dentro del encuadre, lo cual permitió asociar de manera confiable cada fotograma con su fecha y hora correspondiente. La resolución predominante en los archivos es de 704 × 596 píxeles, aunque también se incluyen grabaciones en alta definición (2592 × 1520 píxeles) provenientes de cámaras recientemente instaladas. En total, el conjunto de videos ocupa aproximadamente 35 GB de almacenamiento. No se aplicaron recortes sobre el campo visual original de las cámaras, por lo que todo el contenido fue preservado para el análisis posterior.

Para mejorar la capacidad de generalización del sistema y reforzar la presencia de ciclistas en el conjunto de datos, se incorporaron imágenes provenientes de fuentes complementarias. Estas imágenes adicionales provienen de tomas aéreas realizadas con drones durante eventos ciclistas en la ciudad de Rosario y de cámaras de monitoreo no fijas que forman parte de la red principal. Esta decisión respondió tanto a razones técnicas como prácticas: las bicicletas, motocicletas y peatones son objetos de menor tamaño que los vehículos de cuatro ruedas y suelen estar subrepresentados en escenas urbanas, lo que dificulta su detección por parte de modelos preentrenados.

3.1.2. Separación de fotogramas

Para construir el conjunto de imágenes sobre el que se entrenaron los modelos de detección y clasificación, se diseñó un proceso específico de selección de fotogramas a partir de los videos obtenidos. El objetivo fue reducir la cantidad de imágenes redundantes y, al mismo tiempo, maximizar la representación de los modos de transporte más difíciles de detectar, como las bicicletas y motocicletas. Para ello, se estableció una estrategia de muestreo basada en detecciones preliminares realizadas con un modelo YOLO preentrenado en el dataset públicamente disponible COCO.

Se procesó un fotograma cada s segundos de video y se aplicaron umbrales de confianza diferenciados: 0,6 para vehículos de cuatro ruedas y 0,1 para los de dos ruedas. Esta decisión buscó evitar falsos positivos en los vehículos de mayor tamaño, ya bien representados en modelos preentrenados, y permitir la captura de instancias menos evidentes de bicicletas y motocicletas, incluso si el modelo original las detectaba con baja confianza.

A partir de las detecciones obtenidas, se definieron reglas de conservación de fotogramas. En caso de no detectarse vehículos de dos ruedas:

- Se conserva el fotograma si contiene al menos un vehículo de gran porte (camión o colectivo).
- En ausencia de vehículos de gran porte, se conserva únicamente si incluye más de un automóvil y han transcurrido al menos diez segundos desde la última imagen guardada.

Cuando se detectan vehículos de dos ruedas, se aplica un análisis más detallado sobre un conjunto de $2n + 1$ fotogramas: el actual, junto con n anteriores y n posteriores, espaciados uniformemente a lo largo de los s segundos adyacentes. En

este trabajo se utilizó $n = 3$ y $s = 2$ segundos como configuración base. Para elegir el fotograma más representativo dentro de ese conjunto, se calculó un puntaje compuesto por múltiples criterios.

El puntaje se definió a partir de tres factores: (i) la presencia de detecciones de vehículos de dos ruedas con baja confianza, lo cual permite capturar casos donde el modelo falla o presenta dudas; (ii) la distribución espacial de dichas detecciones dentro del encuadre, con un esquema de grilla $p \times p$ que asigna mayor peso a los cuadrantes menos representados (en este trabajo se utilizó $p = 4$); y (iii) la diversidad de clases presentes en el fotograma, con el objetivo de incluir imágenes con distintos tipos vehiculares en lugar de aquellas dominadas por una única categoría.

Formalmente, el puntaje total S se expresa mediante la ecuación 3.1, donde se combinan los pesos espaciales, de confianza y de diversidad de clase:

$$S = \sum_{i=1}^{N} w_q^{(i)} \cdot w_c^{(i)} + \lambda \cdot D \tag{3.1}$$

- S: puntaje total del fotograma.
- N: número de detecciones de vehículos de dos ruedas.
- $w_q^{(i)}$: peso espacial del cuadrante correspondiente a la i-ésima detección, calculado como:
$$w_q = \log\left(1 + \frac{h + k}{c + k}\right)$$
donde h es la cantidad máxima de detecciones históricas en cualquier cuadrante, c la cantidad acumulada en el cuadrante actual y k una constante positiva de suavizado.
- $w_c^{(i)}$: peso de la confianza de la i-ésima detección, definido como:
$$w_c = \log\left(1 + \frac{1}{\text{conf} + 1}\right)$$
- D: cantidad de clases vehiculares únicas presentes en el fotograma.
- λ: parámetro de regularización para el término de diversidad. En este trabajo se fijó $\lambda = 0{,}03$.

Este enfoque permitió construir un conjunto de imágenes diverso, equilibrado y enfocado en los principales desafíos que presenta el entorno urbano de Rosario. Las detecciones asociadas a los fotogramas seleccionados se almacenaron en archivos XML compatibles con herramientas de etiquetado, para funcionar como anotaciones preliminares que facilitaron la posterior tarea de curación manual.

3.1.3. Filtro de fotogramas redundantes

En muchos casos, la extracción de imágenes genera fotogramas muy similares o directamente duplicados. Esto ocurre principalmente en escenas con vehículos detenidos por semáforos, congestión o circulación lenta, donde la variabilidad

entre fotogramas consecutivos es escasa. Para evitar esta redundancia y optimizar el conjunto de entrenamiento, se desarrolló un procedimiento basado en la detección y emparejamiento de puntos clave.

El proceso parte de las detecciones obtenidas durante la etapa de separación de fotogramas. A cada cámara se le asignó un polígono manualmente definido que delimita las zonas de estacionamiento, con el fin de excluir las detecciones correspondientes a vehículos estáticos. Luego, para cada par de imágenes consecutivas, se identificaron los keypoints sobre las regiones correspondientes a las cajas detectadas (sin considerar los sectores de estacionamiento) mediante el algoritmo SIFT, tal como se muestra en la figura 3.3. En ella se observa cómo los puntos clave tienden a concentrarse en los bordes de los objetos en movimiento, lo cual facilita su posterior comparación con otros fotogramas.

FIGURA 3.3. Ejemplo de detección de keypoints mediante el algoritmo SIFT sobre las regiones de interés de un fotograma.

El emparejamiento de puntos clave entre imágenes se realizó con un algoritmo FLANN-based matcher, cuyos resultados se ilustran en la figura 3.4. En la imagen las líneas indican las correspondencias entre keypoints detectados. A partir de ello, se calculó un coeficiente de similitud como la razón entre la cantidad de keypoints correctamente emparejados y la mayor cantidad de keypoints detectados en cualquiera de las dos imágenes del par. Si este coeficiente supera un umbral predefinido (en este trabajo, $0{,}05$), se asigna una etiqueta de redundancia al segundo fotograma del par.

FIGURA 3.4. Ejemplo de emparejamiento de puntos clave entre dos fotogramas consecutivos mediante un FLANN-based matcher.

Posteriormente, se incorporó un paso adicional de verificación sobre las secuencias marcadas como redundantes. En ciertas ocasiones, una imagen se asemeja notablemente a la anterior debido al escaso desplazamiento de los vehículos, situación que puede repetirse durante varios segundos consecutivos. Sin embargo, no resulta conveniente eliminar todas esas imágenes sin más. Por este motivo, se analiza cada grupo continuo de fotogramas redundantes y se conserva no solo el primero, sino también aquellos que difieren sustancialmente entre sí. Esta estrategia evita descartar imágenes que, si bien comparten similitud local con sus vecinas inmediatas, aportan información distinta en una ventana temporal más amplia.

Luego de aplicar este filtro, se obtuvo un conjunto final de aproximadamente 19 500 imágenes provenientes de cámaras de videovigilancia, complementadas por unas 500 imágenes aéreas tomadas con drones durante eventos ciclistas en la ciudad de Rosario.

3.1.4. Etiquetado de datos para la tarea de detección

Como primer paso, se definió una separación entre los datos destinados al entrenamiento de los modelos y aquellos reservados para su evaluación. En particular, se seleccionaron los videos correspondientes a la cámara ubicada en Ovidio Lagos 3550 como conjunto de evaluación, con el objetivo de contar con una fuente externa y no vista durante el entrenamiento. Los videos seleccionados para este fin forman parte del subconjunto de alta calidad mencionado en la Sección 3.1.2 y representan la resolución y condiciones esperadas para futuras tareas de inferencia.

Las detecciones obtenidas durante la etapa de separación de fotogramas funcionaron como anotaciones preliminares sobre las que se construyó el proceso de etiquetado manual. Para llevar adelante esta tarea se utilizó la herramienta de código abierto CVAT (*Computer Vision Annotation Tool*), que fue descargada y ejecutada mediante contenedores Docker.

Dado que el conjunto de imágenes disponibles superó las 19 500 instancias, no resultó viable realizar una anotación completa. Por tal motivo, se adoptó un enfoque de active learning, basado en la selección progresiva de subconjuntos de imágenes. En una primera etapa, se etiquetó un lote inicial, sobre el que se entrenó un modelo preliminar. Para esta tarea se utilizó YOLOv8m, por su bajo tiempo de entrenamiento y validación. Posteriormente, se aplicó una estrategia de validación cruzada sobre ese conjunto para identificar debilidades en el modelo. A partir de dicho análisis, se seleccionó un segundo lote de imágenes que permitiera abordar los errores detectados. Este proceso se repitió de manera iterativa hasta alcanzar un desempeño satisfactorio. El conjunto final de imágenes etiquetadas se empleó luego para entrenar los modelos definitivos YOLO en sus distintas versiones, así como el modelo RF-DETR.

Al finalizar el proceso, se contó con un conjunto de 629 imágenes anotadas manualmente para entrenamiento, provenientes de videos de distintas cámaras de monitoreo de la ciudad, de imágenes aéreas capturadas mediante drones durante eventos ciclistas y de fuentes complementarias. Además, se reservaron 100 imágenes correspondientes exclusivamente a la cámara ubicada en Ovidio Lagos 3550, utilizadas como conjunto de evaluación. La tabla 3.1 resume la distribución total de imágenes según su uso y fuente.

TABLA 3.1. Distribución de imágenes utilizadas en el trabajo según su uso y fuente.

Fuente	Entrenamiento	Validación	Evaluación
Cámaras de monitoreo (CCTV)	300	60	100
Drones y fuentes complementarias	300	0	0
Total	**600**	**60**	**100**

La selección del lote inicial de imágenes para anotación presentó un desafío particular, dada la gran cantidad de fotogramas disponibles. Con el objetivo de conformar un subconjunto lo más representativo y diverso posible, se optó por una estrategia de aprendizaje no supervisado basada en representaciones latentes. Para ello, se entrenó un modelo BYOL (*Bootstrap Your Own Latent*), una técnica de autosupervisión que permite aprender embeddings representativos de las imágenes. Una visualización de un subconjunto de estas representaciones proyectadas a dos dimensiones mediante t-SNE puede apreciarse en la figura 3.5, donde se observa la diversidad escénica capturada por el modelo. Cada punto del gráfico representa una imagen, ilustrada con su miniatura correspondiente. La distribución refleja la diversidad visual presente en el conjunto de datos.

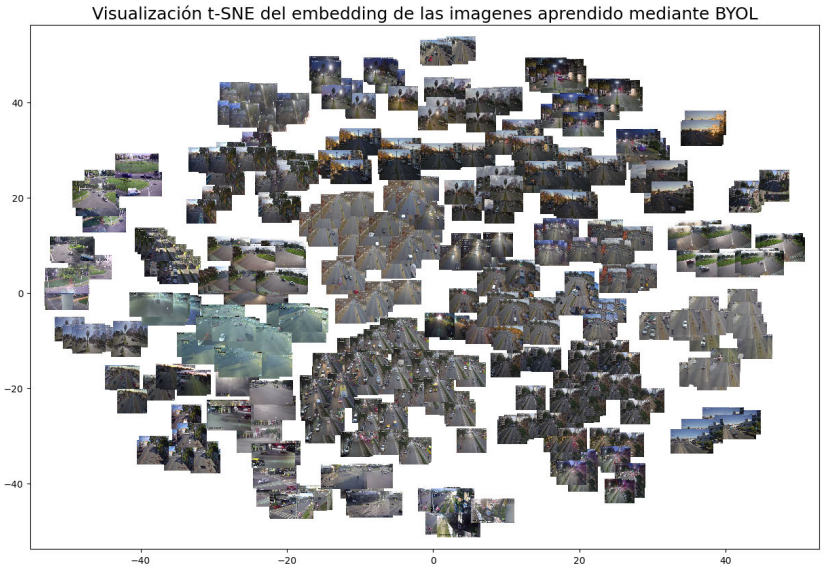

FIGURA 3.5. Proyección bidimensional mediante t-SNE de un subconjunto de embeddings generados por el modelo BYOL.

Una vez obtenidos los embeddings para la totalidad de las imágenes del conjunto de cámaras de monitoreo, se realizó una selección de instancias disímiles entre sí, para priorizar así la variedad de escenas, condiciones lumínicas y distribuciones espaciales de los objetos. De este proceso se extrajeron 195 imágenes para integrar el conjunto de entrenamiento y validación, y otras 100 imágenes adicionales provenientes de la cámara de evaluación que no fueron utilizadas durante el proceso de active learning.

3.2. Entrenamiento del modelo de detección

Esta sección describe el proceso de entrenamiento de los modelos de detección seleccionados para el sistema: YOLOv8 en sus variantes nano, small y medium, y el modelo RF-DETR, basado en transformers. Se detallan las estrategias de data augmentation aplicadas al conjunto de imágenes con el objetivo de mejorar la capacidad de generalización de los modelos. Finalmente, se presenta una evaluación comparativa del desempeño alcanzado por cada arquitectura durante el entrenamiento y validación definidos en la sección 3.1.4.

3.2.1. Aumentación de datos

Para mejorar la capacidad de generalización de los modelos, se aplicó data augmentation en tiempo real durante el entrenamiento. Esto implicó la transformación dinámica de las imágenes mediante una secuencia de operaciones aleatorias, definidas a partir de hiperparámetros específicos. Las transformaciones seleccionadas buscaron representar variaciones plausibles en escenas urbanas que, si bien no estaban presentes en el conjunto de entrenamiento, podrían aparecer en contextos reales durante tareas de inferencia.

Entre las operaciones aplicadas, se destacan las siguientes:

- Escalado y recorte: se modificaron el tamaño y el encuadre de las imágenes para simular diferentes distancias relativas entre la cámara y los vehículos, ya que se considera que futuras instalaciones pueden presentar configuraciones distintas a las utilizadas para la recolección original de los datos.
- Composición: el armado de mosaicos a partir de recortes de hasta cuatro imágenes permitió combinar en una misma escena diferentes modos de transporte capturados en ubicaciones diversas de la ciudad. Esta estrategia contribuyó a generar imágenes más variadas, con lo que se integraron objetos que no suelen coexistir en las tomas originales.
- Cambios de color: se aplicaron alteraciones en la saturación y tonalidad para incrementar la variabilidad en los colores de vehículos y vestimenta de peatones, lo que favoreció una mejor generalización frente a escenarios no vistos.
- Contraste y brillo: si bien el conjunto de imágenes incluía registros tomados en distintos momentos del día, se modificaron estos parámetros para simular condiciones de iluminación aún más diversas.
- Espejado horizontal: esta transformación permitió duplicar virtualmente las situaciones de tránsito observadas, sin romper la coherencia estructural de la escena al no alterar su lógica vial.
- Rotación leve: se aplicó una rotación aleatoria en el rango de $\pm 5^\circ$ con el fin de robustecer el modelo frente a desalineaciones menores producidas por la instalación o el movimiento natural de las cámaras.
- Transformaciones de perspectiva: se introdujeron deformaciones geométricas leves para emular las distintas vistas posibles que pueden surgir del posicionamiento variable de las cámaras en futuras implementaciones.

- Difuminado: se aplicó desenfoque parcial a ciertas imágenes con el objetivo de simular registros obtenidos con cámaras de menor calidad, lo que permitió aumentar la robustez del modelo ante posibles escenarios adversos.

Un ejemplo de los resultados de estas transformaciones en las imágenes puede observarse en la figura 3.6. Allí se observan modificaciones en la perspectiva, color, iluminación y composición espacial.

FIGURA 3.6. Ejemplos de transformaciones aplicadas sobre imágenes del conjunto de entrenamiento.

3.2.2. Ingesta de datos

Para el entrenamiento de los modelos, todas las imágenes fueron redimensionadas a un tamaño fijo de 640×640 píxeles. Dado que las imágenes originales presentaban dimensiones y proporciones variadas, se aplicó padding para preservar su aspect ratio original y evitar deformaciones geométricas. Esta decisión ofreció múltiples ventajas: permitió la compatibilidad con modelos preentrenados como YOLOv8 y RF-DETR, facilitó el procesamiento en lotes mediante GPU al estandarizar las dimensiones de entrada y favoreció un equilibrio entre resolución suficiente para detectar objetos pequeños y eficiencia computacional.

Debido a que la combinación de técnicas de aumento de datos y redimensionamiento puede alterar el tamaño aparente de los objetos dentro de una imagen, se optó por un criterio de anotación conservador durante el etiquetado manual: se incluyeron incluso aquellos objetos lejanos cuya clase podría ser identificada visualmente por una persona. Esta estrategia, conocida como anotación al máximo, permitió preservar la coherencia entre las etiquetas y las transformaciones aplicadas. De este modo, al ampliar una imagen y aplicar recortes durante el procedimiento de aumento, etiquetas inicialmente demasiado pequeñas para que el modelo pudiera aprender de ellas adquieren un tamaño adecuado. Aunque en algunos casos el escalado produce cajas de menor calidad, estas instancias aportan variabilidad al conjunto y contribuyen a robustecer la capacidad general del modelo. Además, esta decisión permite conformar un conjunto de entrenamiento flexible, que no solo resulta útil para su aplicación en este trabajo, sino que también habilita en el futuro el entrenamiento de modelos con entradas de mayor resolución o esquemas de inferencia basados en ventanas.

3.2.3. Entrenamiento de los modelos YOLOv8

El entrenamiento de los modelos Nano, Small y Madium de la familia YOLOv8 se realizó con la API de Ultralytics. El código de entrenamiento corrió en Google Colab con GPU NVIDIA T4 y los pesos preentrenados en el conjunto de COCO. Dado que la tarea de detección de diferentes categorías de automóviles mantiene similitudes con el conjunto original, se optó por congelar las capas del backbone y reentrenar únicamente la cabeza de detección. La búsqueda de hiperparámetros se efectuó de manera manual, para evaluar distintos valores de learning rate y diferentes optimizadores (SGD, Adam y AdamW), obteniéndose los mejores resultados con un valor del orden de 10^{-3} y el optimizador AdamW.

El tamaño de lote se fijó en 16 imágenes, limitado por la memoria disponible en el entorno de entrenamiento. La función de pérdida utilizada fue la predeterminada en YOLOv8, que combina términos de localización, clasificación y confianza. El proceso se ejecutó durante 100 épocas, con la incorporación de un criterio de detención temprana con paciencia de 10 épocas sobre la métrica mAP de validación. Se estableció una semilla de valor 42 para favorecer la reproducibilidad de los resultados.

La figura 3.7 muestra la evolución de las métricas de entrenamiento y validación durante las 100 épocas para dos de los modelos considerados, Nano y Medium, utilizadas para monitorear posibles signos de sobre ajuste.

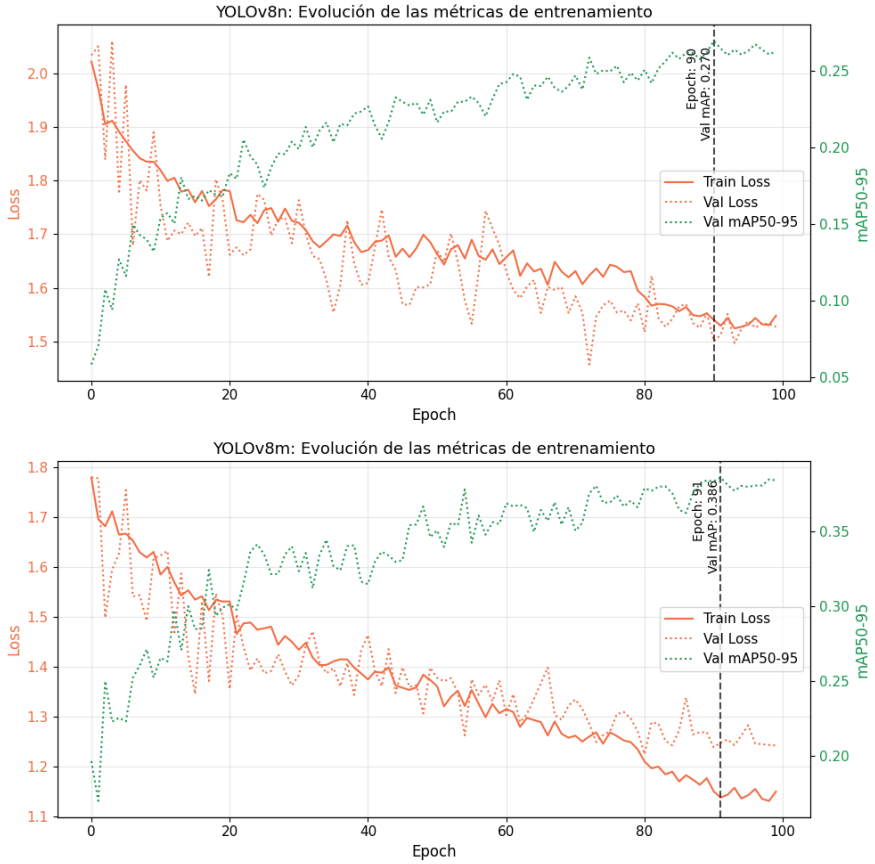

FIGURA 3.7. Evolución de las métricas de entrenamiento y validación durante las 100 épocas para los modelos YOLOv8-n y YOLOv8-m.

3.2.4. Entrenamiento de los modelos RF-DETR

El entrenamiento de los modelos Base, Nano y Small de la familia RF-DETR se realizó con la API de Roboflow. El código de entrenamiento corrió en Google Colab con GPU NVIDIA T4 y los pesos preentrenados en el conjunto de COCO. La búsqueda de hiperparámetros se efectuó de manera manual, para evaluar distintos valores de learning rate y el learning rate del encoder, obteniéndose los mejores resultados con un valor del orden de 10^{-4} y 10^{-5}, respectivamente. El optimizador utilizado fue AdamW.

El entrenamiento de los modelos de la familia RF-DETR resultó computacionalmente más demandante que el de los modelos YOLOv8. Por este motivo, se optó por utilizar un lote de 2 imágenes con acumulación de gradiente de 8 pasados, lo que equivale a un tamaño de lote efectivo de 16 imágenes. La función de pérdida utilizada fue la predeterminada por RF-DETR, que utiliza Hungarian matching para asignar las predicciones a las anotaciones, y posteriormente calcula la pérdida combinada de localización y clasificación. Debido al mayor tiempo de entrenamiento requerido, se redujo la cantidad de épocas totales a 60. Se incluyó un criterio de detención temprana con paciencia de 10 épocas.

La figura 3.8 muestra la evolución de las métricas de entrenamiento y validación durante las 60 épocas para dos de los modelos considerados, Nano y Base, utilizadas para monitorear posibles signos de sobre ajuste.

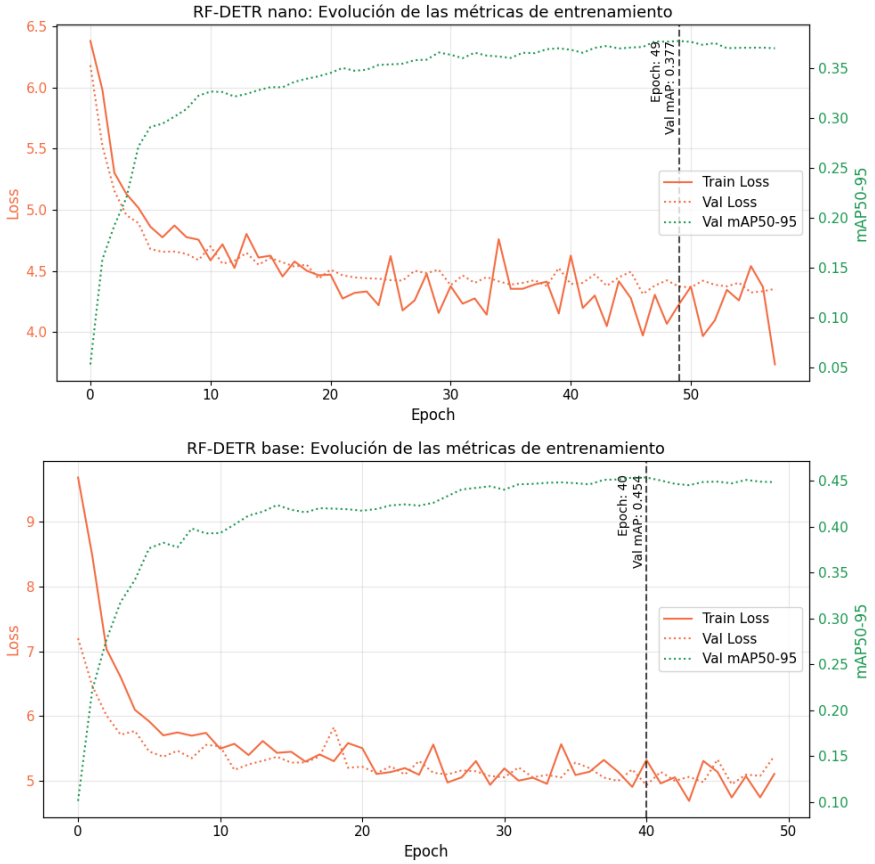

FIGURA 3.8. Evolución de las métricas de entrenamiento y validación durante las 50 épocas para los modelos RF-DETR Base y RF-DETR Nano.

3.3. Preparación de los datos para clasificación

Esta sección describe el proceso de preparación de los datos utilizados para entrenar y evaluar los modelos de clasificación demográfica. Incluye la extracción de ciclistas y peatones a partir de los videos procesados, la selección de instancias representativas y el etiquetado del género como variable objetivo para los experimentos de clasificación.

3.3.1. Obtención de los datos

Para llevar a cabo el proceso de preparación de los datos utilizados para entrenar y evaluar los modelos de clasificación demográfica se implementaron dos procedimientos complementarios: uno automático, que constituye la fuente principal de instancias dado que refleja el modo de operación esperado en producción, y otro manual, empleado de manera secundaria para mejorar la representatividad y corregir desbalances en el conjunto. La combinación de ambas estrategias permitió conformar un corpus diverso y con calidad suficiente para llevar adelante las fases posteriores de entrenamiento y validación. La figura 3.9 muestra ejemplos por clase de las imágenes obtenidas mediante ambos procedimientos. En ella se observa diversidad en términos de ángulo de captura, vestimenta, entorno y condiciones lumínicas.

FIGURA 3.9. Ejemplos anonimizados de imágenes de ciclistas y peatones obtenidas mediante los procedimientos automático y manual.

El procedimiento automático se basó en el procesamiento de todos los fotogramas extraídos durante la etapa de detección mediante el procedimiento explicado en la subsección 3.1.2. Para esto se incluyeron también aquellos marcados como repetidos, ya que fueron designados de esa manera debido a la duplicación de vehículos de cuatro ruedas, no de ciclistas o peatones. Al igual que para la tarea de detección, se seleccionaron los fotogramas pertenecientes a la cámara ubicada en Ovidio Lagos 3550, como fotogramas de evaluación. De este modo, se consiguió evitar cualquier riesgo de data leaking.

Para la extracción de los recortes, se definieron polígonos de interés próximos a la cámara con el objetivo de conservar únicamente las instancias con mayor nivel de detalle, y se aplicaron reglas mínimas de tamaño para garantizar la legibilidad. A partir de las detecciones generadas por los modelos previos, se construyeron anotaciones en formato Pascal VOC. Luego, estas fueron filtradas según la intersección con los polígonos y clasificadas como peatones, ciclistas o motociclistas, en función de las clases asignadas a cada bounding box que intersecta la persona.

Una vez generados los recortes automáticos, se llevó a cabo una verificación manual orientada a garantizar la calidad visual y la diversidad de las muestras. Posteriormente, las instancias fueron etiquetadas en la herramienta CVAT con las categorías `masculino` y `femenino`, aunque se dejaron sin clasificar aquellos casos en los que la ambigüedad impidió una asignación confiable. Estas imágenes reservadas podrían aprovecharse en futuras investigaciones sobre estrategias de aprendizaje con incertidumbre, sin afectar la robustez del conjunto utilizado en el presente trabajo.

En paralelo, se realizó un proceso de extracción manual directamente sobre los videos crudos, tanto provenientes de cámaras de videovigilancia como de secuencias aéreas obtenidas con dron. Este procedimiento permitió incorporar ciclistas en posiciones típicas de las escenas de inferencia, así como mitigar el fuerte desbalance de género presente en las instancias automáticas. Durante esta etapa, la selección de fotogramas y el etiquetado demográfico se efectuaron de manera simultánea para asegurar la consistencia en las categorías asignadas y eficiencia en la construcción del conjunto.

Finalmente, y con el objetivo de aumentar la capacidad de generalización del modelo, se incorporaron imágenes peatonales obtenidas de múltiples fuentes en la web y curadas manualmente. Estas imágenes representan aproximadamente el 10 % del set de entrenamiento final.

En la tabla 3.2 se presenta el resumen de la cantidad de instancias obtenidas según la estrategia empleada y las distribuciones alcanzadas por uso y clase.

TABLA 3.2. Distribución de imágenes utilizadas en el trabajo por origen y uso.

Fuente	**Entrenamiento**	**Evaluación**
Extracción automatizada	1551 F / 2808 M	90 `femenino` / 110 M
Extracción manual	1084 F / 982 M	0
Fuentes complementarias	250 F / 250 M	0
Total	**2885 F / 4040 M**	**200**

3.4. Entrenamiento del modelo de clasificación

Esta sección describe el proceso de entrenamiento de los modelos de clasificación considerados para el sistema, y se incluye EfficientNet en sus variantes b0 y b3, y ResNet en sus variantes 50 y 101. Se detallan las estrategias de data augmentation aplicadas al conjunto de imágenes con el objetivo de mejorar la capacidad de generalización de los modelos. Finalmente, se presenta una evaluación comparativa del desempeño alcanzado por cada arquitectura durante el entrenamiento y validación definidos en la sección 3.3.1.

3.4.1. Aumentación de datos

Con el objetivo de incrementar la robustez y la capacidad de generalización del modelo de clasificación, se aplicó data augmentation en tiempo real durante el entrenamiento. Esta estrategia permitió ampliar la diversidad del conjunto de imágenes sin necesidad de recolectar nuevos datos, lo cual posibilitó contar con variaciones que simulan escenarios no existentes en los datos originales. Las transformaciones fueron definidas a partir de un conjunto de hiperparámetros ajustables, lo que permitió controlar su intensidad y frecuencia de aplicación.

Un ejemplo ilustrativo de estas transformaciones puede observarse en la figura 3.10, donde se evidencian variaciones en geometría, color y textura que enriquecen el conjunto de entrenamiento.

FIGURA 3.10. Transformaciones aplicadas sobre imágenes del conjunto de entrenamiento para la tarea de clasificación. Se incluyen variaciones geométricas, cromáticas y de textura.

Las operaciones implementadas incluyeron:

- Transformaciones espaciales: se aplicaron escalados, traslaciones, rotaciones aleatorias y cambios de perspectiva, con el fin de simular la variabilidad en el posicionamiento de la cámara y la orientación de los objetos en la escena.
- Espejado: la inversión horizontal introdujo diversidad adicional sin alterar la coherencia de los elementos representados.
- Efectos de desenfoque: se incorporaron desenfoques leves, tanto gaussianos como por movimiento, para reflejar posibles condiciones de captura asociadas a cámaras de distinta calidad o a situaciones de dinámica urbana.
- Ajustes de color: se modificaron brillo, contraste, saturación y tonalidad, lo que aumentó la variabilidad cromática de las imágenes y contribuyó a mejorar la generalización frente a condiciones lumínicas heterogéneas.
- Corrección gamma: se aplicaron ajustes suaves sobre la luminancia, con el objetivo de robustecer el modelo ante cambios de exposición.
- Eliminación aleatoria de regiones: se introdujo la técnica de random erasing, que consiste en suprimir fragmentos de la imagen de forma aleatoria para

evitar que el modelo dependa en exceso de regiones específicas durante la clasificación.

3.4.2. Ingesta de datos

Para el entrenamiento de los modelos de clasificación, todas las imágenes fueron redimensionadas a un formato cuadrado mediante el estiramiento del lado menor, de modo que cada arquitectura recibiera como entrada la dimensión requerida por su diseño (por ejemplo, 224 x 224 píxeles en el caso de EfficientNet-b0). Este procedimiento aseguró la compatibilidad entre las imágenes y los modelos preentrenados, además de permitir un procesamiento uniforme en lotes. Finalmente, se aplicó una normalización basada en las estadísticas de conjuntos de referencia ampliamente utilizados, lo que favoreció la estabilidad numérica durante el entrenamiento y la transferencia efectiva de conocimiento desde los modelos base.

3.4.3. Búsqueda de hiperparámetros

Tanto la limitada cantidad de imágenes disponibles como la diferencia significativa entre las características del conjunto ImageNet y las de la tarea específica de clasificación planteada, motivaron la elección de un enfoque basado en el ajuste fino de modelos preentrenados. En este sentido, se seleccionaron arquitecturas de redes convolucionales ampliamente reconocidas por su desempeño en tareas de clasificación de imágenes, tales como EfficientNet y ResNet, en dos variantes cada una.

Para llevar adelante este procedimiento resultó indispensable explorar para cada arquitectura la cantidad de capas a congelar antes de iniciar el ajuste de parámetros, para equilibrar la preservación de representaciones generales y la capacidad de especialización. Este análisis fue particularmente relevante para los modelos más complejos, donde un exceso de capas entrenables incrementaba el riesgo de sobreajuste dada la magnitud reducida del conjunto de datos.

Además de la cantidad de capas a congelar, otros hiperparámetros clave como dropout, tasa de aprendizaje, weight decay y optimizador (con variantes Adam, AdamW y SGD), se seleccionaron mediante una búsqueda sistemática con la librería Optuna. Dicha herramienta, basada en técnicas bayesianas, permitió explorar espacios de búsqueda adaptados a cada arquitectura. Se incluyeron también parámetros asociados a data augmentation (ángulo de rotación, variaciones de brillo y contraste), considerados fundamentales para robustecer los modelos. Cada variante fue ensayada en un mínimo de 30 ejecuciones de 15 épocas cada una.

3.4.4. Entrenamiento de los modelos EfficientNet y Resnet

El procedimiento de entrenamiento se mantuvo similar en todos los modelos, con diferencias puntuales en tres aspectos fundamentales: la cantidad de capas congeladas, el tamaño de las imágenes de entrada requerido por cada arquitectura y el tamaño del lote, condicionado por las restricciones del hardware disponible. De este modo, modelos más livianos como EfficientNet-b0 permitieron utilizar lotes mayores, mientras que variantes de mayor capacidad, como ResNet-101, debieron entrenarse con lotes más reducidos. Esta estrategia buscó maximizar la eficiencia del entrenamiento sin comprometer la estabilidad de la convergencia.

El conjunto de validación fue diseñado con especial cuidado para asegurar un balance de clases y evitar sesgos en la selección de los mejores hiperparámetros. Se conformó con aproximadamente 150 a 200 imágenes por clase, equivalentes a cerca del 5 % del conjunto de entrenamiento. Este esquema garantizó un criterio estable para comparar configuraciones y redujo la variabilidad entre ejecuciones. El conjunto de hiperparámetros con mayor precisión de validación fue adoptado para el entrenamiento final de cada arquitectura, y constituyó la base de la evaluación comparativa entre modelos.

Finalmente, los modelos seleccionados se entrenaron durante un máximo de 50 épocas, con paciencia de 15 épocas sobre el conjunto de validación. Para evitar una eliminación prematura de pesos prometedores, se implementó un scheduler de la tasa de aprendizaje que incluyó cinco épocas de calentamiento inicial, seguidas por un esquema coseno decreciente. La función de pérdida empleada fue Weighted Cross Entropy, adecuada para compensar el desbalance de clases. La evolución de las métricas de entrenamiento y validación de cada arquitectura se ilustra a continuación en la figura 3.11 para los modelos de arquitectura EfficientNet, y en la figura 3.12 para los modelos de arquitectura ResNet.

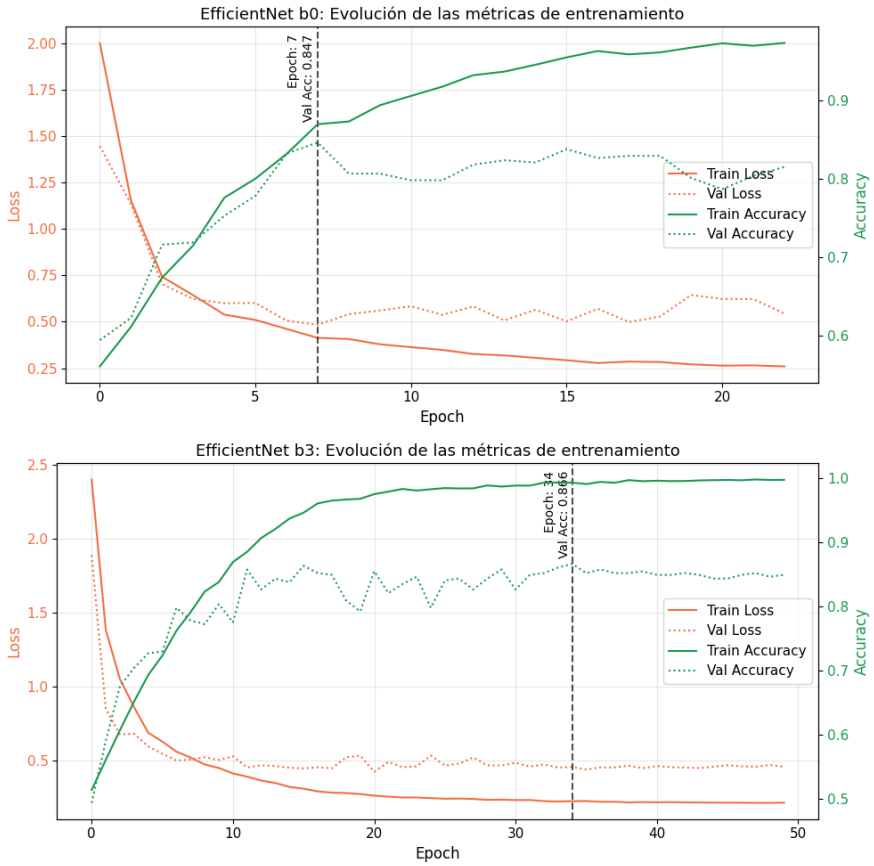

FIGURA 3.11. Evolución de las métricas de entrenamiento y validación durante las 50 épocas para los modelos EfficientNet-b0 y EfficientNet-b3.

Los resultados de la familia EfficientNet permiten observar dinámicas de entrenamiento distintas, aunque aún no definitivas respecto al rendimiento comparativo de cada variante. En el caso de EfficientNet-b0, la rápida estabilización en torno a un máximo de validación de 0,847 en la época 7, seguida por una ausencia de

mejoras, podría interpretarse como una tendencia temprana a caer en mínimos locales, lo que limita la capacidad de aprovechar el resto de las épocas. Por su parte, EfficientNet-b3 mostró un proceso más prolongado, con oscilaciones que alcanzaron un máximo de 0,866 recién en la época 34, lo cual refleja una mayor flexibilidad del modelo pero también un mayor riesgo de sobreajuste si no se controla adecuadamente. En cualquier caso, estos valores corresponden al conjunto de validación y, por tanto, ofrecen un indicio preliminar de desempeño más que una conclusión definitiva sobre la capacidad de generalización de los modelos.

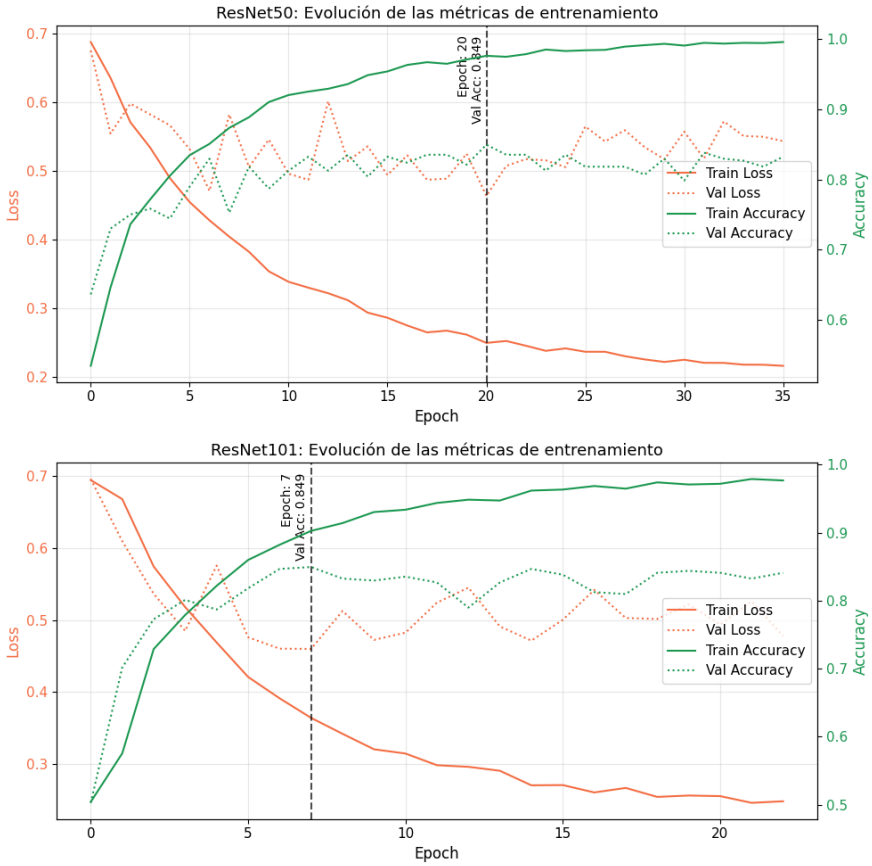

FIGURA 3.12. Evolución de las métricas de entrenamiento y validación durante las 50 épocas para los modelos ResNet-50 y ResNet-101.

Los resultados de la familia ResNet evidencian una dinámica de entrenamiento más acelerada, con ambos modelos alcanzando su máximo de validación en la época 20 para ResNet 50, y en la época 7 para ResNet 101. Ambos modelos alcanzaron un máximo de validación de 0,849, lo que es comparable al obtenido en los modelos EfficientNet. Sin embargo, la rapidez con la que se alcanzaron estos valores sugiere un riesgo elevado de sobreajuste, especialmente en el caso de ResNet 101, donde la complejidad del modelo podría estar contribuyendo a una adaptación excesiva a las particularidades del conjunto de entrenamiento. Este riesgo fue mitigado mediante la implementación de detención temprana, conservando como modelo a evaluar aquel con el mejor desempeño de validación.

Capítulo 4

Ensayos y resultados

En este capítulo se presentan los resultados obtenidos durante la evaluación de los modelos de detección y clasificación desarrollados. Se detallan las características del conjunto de evaluación utilizado, el procedimiento seguido para la evaluación y los resultados obtenidos, y se incluyen métricas cuantitativas y análisis cualitativos.

4.1. Evaluación de los modelos de detección

Esta sección detalla el procedimiento seguido para evaluar los modelos de detección entrenados, así como las características del conjunto de evaluación utilizado y los resultados obtenidos.

4.1.1. Características de los datos

Como fue explicado en la subsección 3.1.2, los fotogramas utilizados para construir el conjunto de evaluación fueron obtenidas de la cámara ubicada en Ovidio Lagos 3550, que no fue utilizada en el entrenamiento de los modelos de detección.

Esta camara provee imágenes en una resolución de 2592 x 1520 píxeles, lo que representa una ventaja significativa en comparación con las otras cámaras disponibles, que ofrecen resoluciones considerablemente más bajas. Otra de las ventajas que presentó esta cámara es la diversidad de condiciones de iluminación disponibles en los videos grabados, que incluyen tanto condiciones diurnas como nocturnas. Esta variedad contribuye a evaluar el desempeño de los modelos en escenarios más realistas y desafiantes.

Al igual que en el conjunto de entrenamiento, las imágenes del conjunto de evaluación fueron anotadas incluyendo todas las instancias a detectar presentes en la escena, independientemente de su tamaño o nivel de oclusión. Debido a la alta resolución de las imágenes, esta técnica permitió etiquetar instancias ubicadas hasta a 100 metros de la cámara. Indudablemente la detección de estas instancias no solo representa una tarea excesiva para los modelos, sino que carece de fines prácticos en producción. El motivo de emplear esta técnica de anotación responde a la posibilidad de utilizar estas imágenes dentro del conjunto de entrenamiento y etiquetar un nuevo lote para la evaluación de próximas iteraciones del entrenamiento. En consonancia con estas apreciaciones, para la evaluación de los modelos de detección se estableció un polígono de interés que incluye las detecciones que se espera sean útiles para las tareas de conteo vehicular y de demografía ciclista. La figura 4.1 muestra un fotograma de ejemplo con aquellas etiquetas incluidas dentro del polígono de interés.

FIGURA 4.1. Ejemplos de fotograma con etiquetas manuales dentro del polígono de interés.

4.1.2. Evaluación general de los modelos

Para la evaluación de los modelos de detección se realizó inferencia sobre los 100 fotogramas que componen el conjunto de evaluación. Tras filtrar las detecciones no incluidas dentro del polígono de interés se procedió a calcular las métricas mAP@50 y mAP@50-95 usadas comúnmente para comparar el desempeño de diferentes modelos en tareas de detección. Como baseline se seleccionó el modelo YOLOv8m con los pesos preentrenados en el conjunto de COCO y disponibilizado por Ultralytics. Para realizar la comparación, las etiquetas correspondientes a la clase `van` (no presente en el dataset de COCO) fueron mapeadas a la clase `auto`, por lo que el modelo baseline contó con una clase menos en relación con los modelos entrenados para este trabajo.

La figura 4.2 muestra los resultados generales obtenidos tras el entrenamiento de los modelos. Ambas métricas, mAP@50 y mAP@50-95, fueron calculadas promediando el valor obtenido para cada clase.

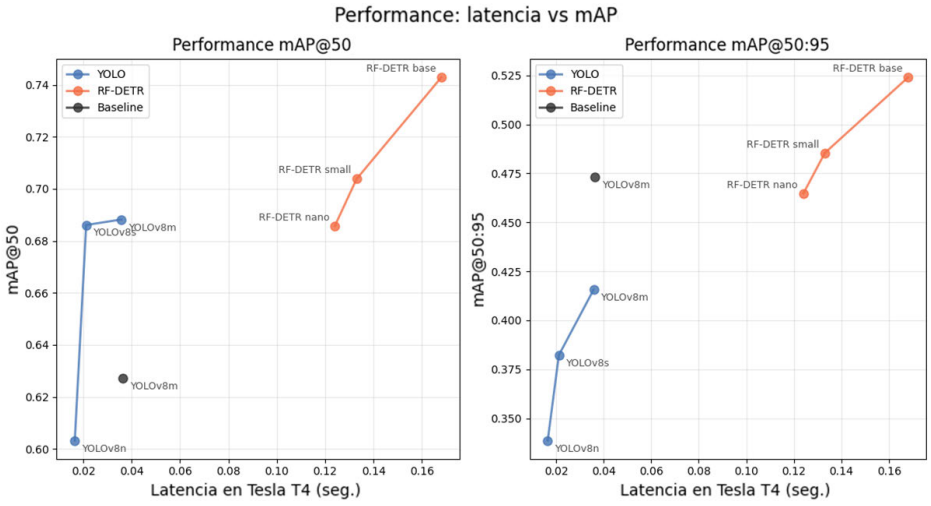

FIGURA 4.2. Comparativa de la performance y la latencia de los modelos.

Las métricas mAP@50 y mAP@50-95 de los modelos YOLO son menores que aquellas obtenidas para los modelos RF-DETR, sin embargo el peor desempeño se ve compensado por su menor latencia. Practicamente la totalidad de los modelos muestran un mAP@50 mayor al obtenido por el modelo utilizado como baseline. Se observa, por otro lado, que el modelo baseline obtuvo mejor resultado en la métrica mAP@50-95 en relación con los modelos entrenados de su misma familia.

Si bien el modelo baseline muestra un desempeño aceptable en la métrica mAP@50-95, es importante recordar que resulta del promedio obtenido para cada clase. De este modo, una clase con muy baja performance puede significativamente afectar de manera negativa la métrica y viceversa. Por este motivo, en la subsección 4.1.3 se profundiza en los resultados obtenidos para cada una de las clases relevantes.

4.1.3. Evaluación específica de los modelos

Para facilitar la comparativa entre los diferentes modelos en la evaluación específica se agruparon las clases `auto`, `camión` y la introducida `van` (no presente en el modelo baseline) promediando los valores obtenidos para las métricas mAP@50 y mAP@50-95. Por otro lado, las clases de mayor importancia para el trabajo (`persona`, `bicicleta`, `motocicleta` y `colectivo`) se reportan de manera individual.

En la figura 4.3 se observa que el modelo baseline presenta un desempeño considerablemente mejor en ambas métricas en la clase que combina `auto`, `camión` y `van`. Mientras que el valor obtenido en la clase `colectivo` es comparable al del resto de los modelos, se observa que en las clases `persona`, `bicicleta` y `motocicleta`, su performance es pobre. Respecto a los modelos entrenados, podemos observar la misma tendencia encontrada durante la evaluación general, en donde los modelos de la familia RF-DETR obtuvieron un puntaje superior en casi todos los casos en comparación con aquellos de la familia YOLO.

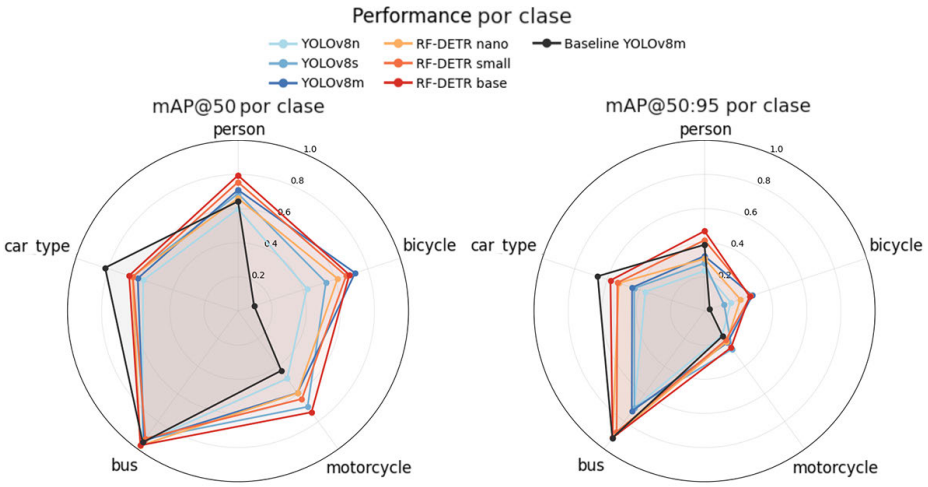

FIGURA 4.3. Los modelos entrenados muestran una mejor performance en las clases relavantes para el trabajo.

Esta figura contribuye a entender el motivo por el que el modelo baseline mostró un mAP@50-95 mayor al resto de los modelos YOLO a pesar de no haber sido entrenado en el dataset etiquetado para este trabajo. El valor obtenido por este

modelo se encuentra sobredimensionado por su gran performance para las clases `auto`, `camión` y `colectivo`, detectables fácilmente por la mayoría de los modelos populares entrenados en el dataset de COCO. Los modelos entrenados mediante aprendizaje por transferencia presentaron un desafío mayor a partir de la introducción de la clase `van`. La baja representatividad de esta nueva clase en el dataset y la pertenencia de sus instancias a la clase `auto` en el modelo original, generaron errores de etiqueta entre ambas que disminuyeron el puntaje obtenido por los modelos entrenados. Sin embargo, la introducción de esta clase en el dataset busca sentar las bases para la expansión de las capacidades del sistema. Para los requerimientos actuales, las clases `auto`, `camión` y `van` serán tratadas como una misma clase tras efectuar la detección, y eventuales confusiones de etiqueta entre ellas no implicarán errores relevantes.

Por otro lado, las clases `persona`, `bicicleta`, `motocicleta` y `colectivo` representan clases de real interés para el trabajo. Entre estas, la última mencionada presenta un gran desempeño en todos los modelos, incluido el baseline, por lo que no amerita un análisis detallado. Los resultados obtenidos para la métrica mAP@50 en la detección de las otras tres clases se muestran el la figura 4.4. Se percibe que el modelo RF-DETR obtuvo la mejor calificación para las clases `persona` y `motocicleta`, mientras que YOLOv8m lo hizo para la clase `bicicleta`. Resulta interesante notar que, si bien los modelos de mayor tamaño dentro de cada familia tienden a obtener mejores métricas, casi todos los modelos presentan resultados comparables. Si bien YOLOv8n no resulta adecuado para su implementación en producción dado los requerimientos del trabajo, los restantes modelos de la familia realizan un trabajo aceptable para la detección de al menos dos de las tres clases estudiadas. En consideración del menor tiempo de latencia (una fracción del correspondiente a los modelos RF-DETR), parecen ser los modelos adecuados en un sistema que busque balancear ambos criterios.

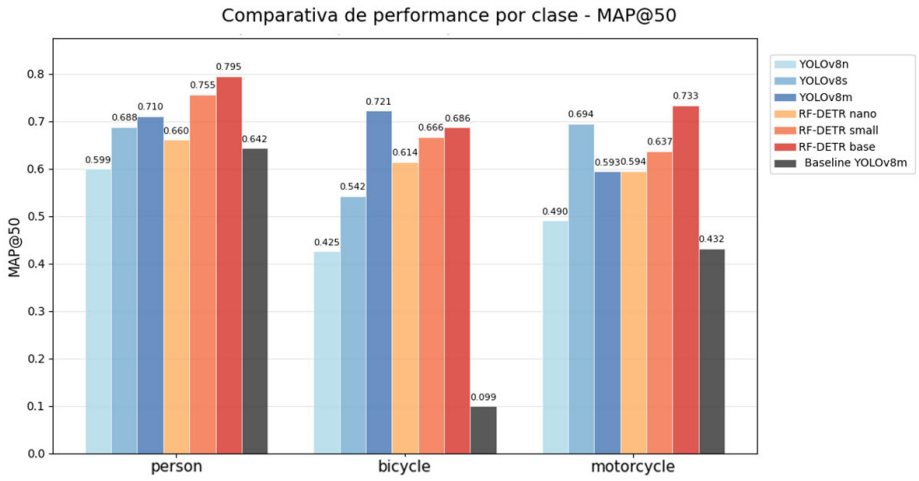

FIGURA 4.4. Desempeño de los modelos en las clases de interés.

4.2. Evaluación de los modelos de clasificación

Esta sección detalla el procedimiento seguido para evaluar los modelos de clasificación entrenados, así como las características del conjunto de evaluación utilizado y los resultados obtenidos.

4.2.1. Características de los datos

Como fue explicado en la subsección 3.3.1, las imágenes utilizadas para construir el conjunto de evaluación fueron obtenidas de los fotogramas correspondientes a la cámara ubicada en Ovidio Lagos 3550, la que no fue utilizada en el entrenamiento de los modelos de detección ni de clasificación.

La utilización de esta cámara, si bien presentó la ventaja de ofrecer fotogramas en una resolución aceptable (2592 x 1520 píxeles), también implicó ciertos desafíos adicionales. En primer lugar, y debido a que los videos disponibles fueron grabados durante el invierno, se observó una mayor cantidad de ropa en los ciclistas, lo que dificultó la identificación de género en muchos casos incluso para el ojo humano. Además, las grabaciones disponibles, correspondientes a ese cámara, fueron tomadas en tres períodos a lo largo del día: de 8:00 a 8:30, de 12:30 a 13:30 y de 17:00 a 19:00. Esto implicó que muchas imágenes presentaran condiciones de iluminación subóptimas, lo que afectó la calidad visual. Si bien estas condiciones impidieron la identificación de la clase correspondiente para algunos ciclistas, también contribuyeron a disponer de un conjunto de evaluación de características complejas que proporciona métricas realistas sobre el desempeño de los modelos.

Como criterio para seleccionar los recortes de ciclistas que forman parte del conjunto de evaluación, se consideraron principalmente aquellos que se encontraban recorriendo la escena sobre la ciclovía, y aproximadamente uno o dos segundos antes de desaparecer del encuadre para los ciclistas que se dirigían en dirección a la cámara, y uno o dos segundos desde su aparición en escena para aquellos que se alejaban de la cámara hacia el fondo de la imagen. De este modo, se buscó asegurar que los ciclistas incluidos en el conjunto de evaluación estuvieran relativamente cerca de la cámara. Esta característica es consistente con el objetivo de identificar la demografía de los ciclistas en movimiento, para lo que la inferencia se desarrollará en aquellos frames que provean la mayor confianza.

En total, el conjunto de evaluación final estuvo compuesto por 200 imágenes, las que fueron anotadas manualmente para asignar la clase correspondiente a cada ciclista visible en la escena. En total, se identificaron 90 ciclistas femeninas y cerca de 300 ciclistas masculinos. Para conservar el balance del conjunto de entrenamiento y evitar un sesgo hacia la clase mayoritaria, se decidió incluir tan solo 110 ciclistas masculinos seleccionados aleatoriamente entre los disponibles. La figura 4.5 muestra ejemplos por clase de las imágenes utilizadas para la evaluación de los modelos.

FIGURA 4.5. Ejemplos anonimizados de imágenes de ciclistas y peatones obtenidas para la evaluación de los modelos de clasificación.

4.2.2. Ajuste del umbral de decisión

En la evaluación de los modelos de clasificación se empleó un umbral de decisión específico para cada arquitectura, determinado a partir de un conjunto de validación independiente. Este conjunto se conformó con imágenes de licencia abierta obtenidas de internet, complementadas con capturas de las mismas cámaras utilizadas en el entrenamiento pero que no habían sido incluidas ni en el conjunto de entrenamiento ni en el de evaluación.

El procedimiento de ajuste del umbral consistió en una búsqueda en grilla, evaluando el desempeño de los modelos en intervalos entre 0,1 y 0,9 con un paso de 0,05. Dado que se trata de un problema de clasificación binaria, los falsos negativos de una clase se traducen en falsos positivos de la clase complementaria. En consecuencia, una mejora notable en una clase suele implicar un sesgo marcado en la otra. Por ello, la métrica F1-score promedio no resultó adecuada para la selección del umbral. En su lugar, se optó por optimizar el Recall de cada clase, reduciendo al mismo tiempo la brecha de desempeño entre ambas.

En todos los modelos entrenados, el umbral de decisión seleccionado fue superior a 0,5. Este resultado sugiere que los modelos tienden a asignar mayor confianza a la clase `masculino`, que no solo corresponde a la clase mayoritaria en el conjunto de entrenamiento, sino que también presenta características visuales más definidas y homogéneas en comparación con la clase `femenino`. La tabla 4.1 resume los valores de umbral determinados para cada arquitectura.

TABLA 4.1. Umbrales de decisión para cada modelo de clasificación

Modelo	**Umbral de decisión**
EfficientNet-b0	0,7
EfficientNet-b3	0,8
ResNet-50	0,8
ResNet-101	0,65

4.2.3. Evaluación de los modelos

Para la evaluación de los modelos de clasificación, se procedió a realizar inferencia sobre cada una de las 200 imágenes que componen el conjunto de evaluación. Previo a la ingesta de los datos en el modelo, se realizó el mismo preprocesamiento de redimensionamiento y normalizado de las imágenes descrito en la subsección 3.4.2.

Las métricas utilizadas para la evaluación de cada uno de los modelos fueron Precision, Recall y F1-Score, que fueron calculadas a partir de la matriz de confusión generada al comparar las predicciones del modelo con las etiquetas reales asignadas manualmente. Estas métricas fueron calculadas para cada una de las clases individualmente, así como también los promedios macro, que ofrecen una visión general del desempeño del modelo.

La figura 4.6 presenta la matriz de confusión y las métricas asociadas para el modelo EfficientNet-b0. Allí puede observarse que el modelo se desempeña equilibradamente entre ambas clases, con un F1-Score promedio de 0,829. La misma métrica individualizada para cada clase es de 0,815 para la clase `femenino` y de 0,843 para la clase `masculino`. Las métricas de Precision y Recall también son similares entre ambas clases, lo que indica que el modelo no presenta un sesgo significativo hacia ninguna de ellas.

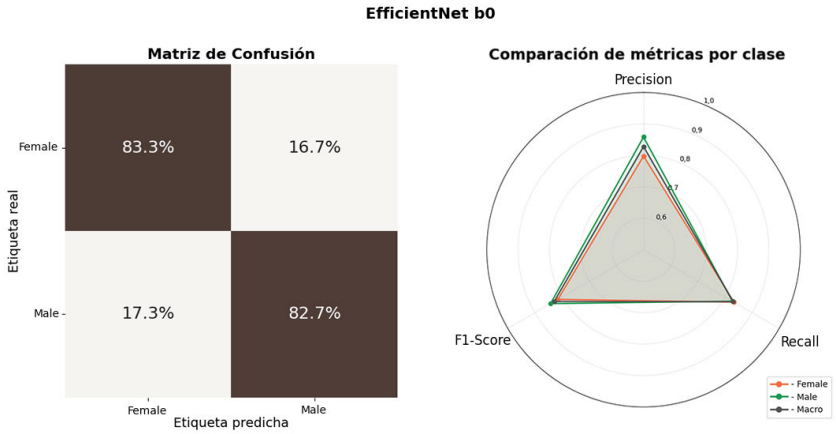

FIGURA 4.6. Los resultados para el modelo EfficientNet-b0 en el conjunto de evaluación muestran un desempeño equilibrado entre las clases con un F1-Score promedio de 0,829.

Por su parte, la figura 4.7 muestra los resultados para el modelo EfficientNet-b3, que exhibe un desempeño casi perfecto para las imágenes de clase real `masculino`, pero considerablemente peor para imágenes de clase real `femenino`, lo que se traduce en un F1-Score promedio de 0,852. Si bien el F1-Score para ambas clases individualizadas es similar (0,818 para `femenino` y 0,886 para `masculino`), la diferencia en las métricas de Precision y Recall entre ambas clases es significativa. En particular, el modelo presenta una Precision muy alta para la clase `femenino` (0,984), pero un Recall considerablemente más bajo (0,700), lo que da cuenta de ciertas limitaciones del modelo para clasificar adecuadamente las imágenes de esta clase. Para la clase `masculino`, el modelo presenta un Recall de 0,990, y una Precision de 0,801 afectada negativamente por la clasificación errónea de las imágenes de clase `femenino`.

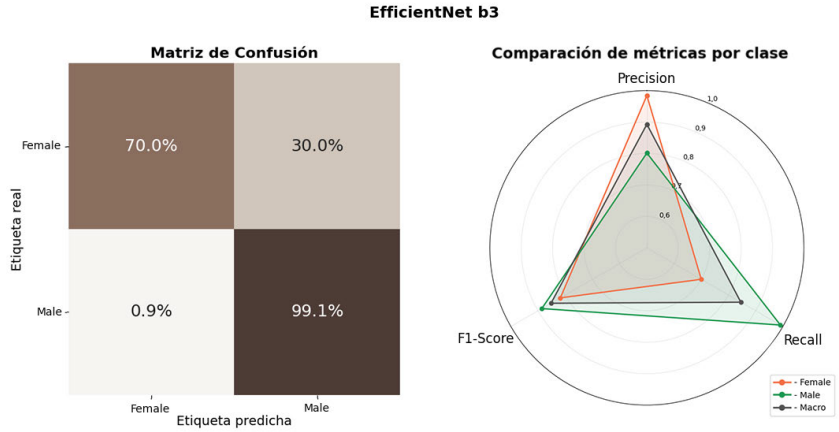

FIGURA 4.7. Los resultados para el modelo EfficientNet-b3 en el conjunto de evaluación muestran un desempeño casi perfecto para las imágenes de clase real `masculino`, pero considerablemente peor para imágenes de clase real `femenino`.

En cuanto al desempeño de los modelos de la familia ResNet, la figura 4.8 presenta los resultados para el modelo ResNet-50. Allí puede observarse que el modelo presenta un desempeño satisfactorio y equilibrado para ambas clases, con un F1-Score promedio de 0,869. La misma métrica individualizada para cada clase es de 0,857 para la clase `femenino` y de 0,881 para la clase `masculino`. Para cada una de las clases, `femenino` y `masculino`, se obtuvieron valores de Precision de 0,848 y 0,889 respectivamente, y valores de Recall de 0,867 y 0,873 respectivamente, lo que indica que el modelo no presenta un sesgo significativo hacia ninguna de las clases.

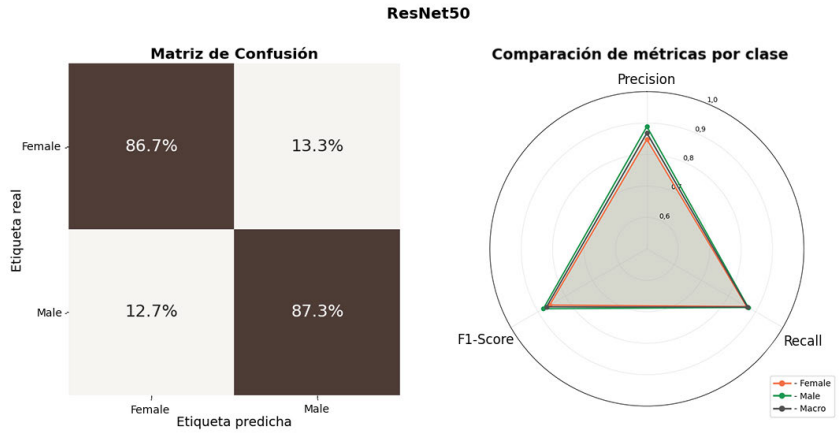

FIGURA 4.8. Los resultados para el modelo ResNet-50 en el conjunto de evaluación muestran un desempeño satisfactorio y equilibrado para ambas clases.

Finalmente, la figura 4.9 muestra los resultados para el modelo ResNet-101, el que exhibe un desempeño satisfactorio para ambas clases, aunque levemente mejor para la clase `masculino`, lo que se traduce en un F1-Score promedio de 0,868. El F1-Score promedio, con un valor de 0,868, es similar al obtenido por el modelo ResNet-50 . Las métricas de Precision son similares entre sí para ambas clases,

con 0,864 para la clase `femenino` y 0,875 para la clase `masculino`. Por otro lado, las métricas de Recall presentan una diferencia más significativa, con un valor de 0,844 para la clase `femenino` y 0,891 para la clase `masculino`, lo que indica que el modelo tiene una mayor capacidad para identificar correctamente las imágenes de esta última clase.

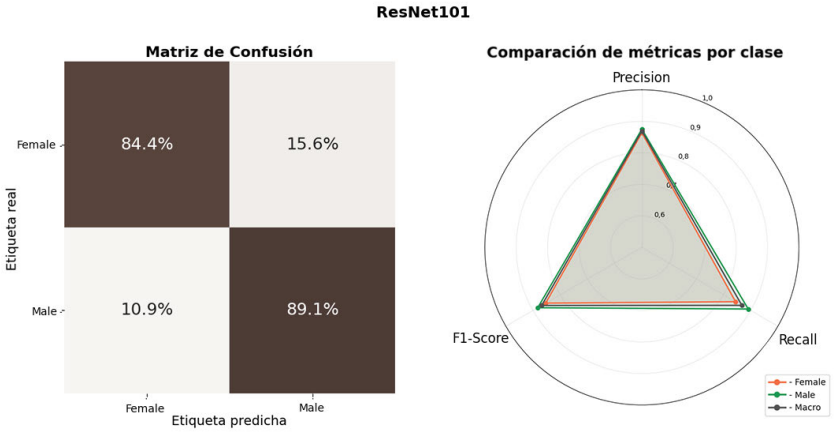

FIGURA 4.9. Los resultados para el modelo ResNet-101 en el conjunto de evaluación muestran un desempeño satisfactorio para ambas clases, aunque un valor de Recall levemente mejor para la clase masculina.

Capítulo 5

Conclusiones

En este capítulo se presenta una síntesis de los principales resultados alcanzados en el desarrollo del sistema de automatización de censos ciclistas mediante visión por computadora. Se evalúa el cumplimiento de los objetivos propuestos, las metodologías empleadas y los desafíos encontrados durante la implementación. Finalmente, se identifican las líneas de continuación que permitirían mejorar y expandir las capacidades del sistema desarrollado.

5.1. Conclusiones generales

El trabajo logró completar satisfactoriamente casi la totalidad de los requerimientos trazados en la planificación. A continuación se itemizan los resultados principales.

- Cumplimiento de requerimientos de detección: se implementó un sistema para detectar y contar vehículos en videos de cctv. Se logró utilizar este sistema para reconocer y extraer ciclistas. Se entrenaron modelos de dos familias diferentes y se evaluaron en su precisión y latencia.
- Cumplimiento de requerimientos de clasificación: se implementó satisfactoriamente un sistema para clasificar imágenes de ciclistas según su género. Se entrenaron modelos de dos familias diferentes y se compararon sus resultados para evaluar la pertiencia de cada uno de ellos con la tarea.
- Planificación y cronograma: el trabajo se desarrolló utilizando como guía la planificación inicial. Sin embargo, demandó mayor tiempo que el previsto originalmente debido a factores externos a la realización de este.
- Gestión de riesgos: dentro de los riesgos previstos se incluyó la posibilidad de una baja calidad de etiquetado de los datos. La mitigación de este riesgo implicó el etiquetado manual por parte del autor de cada una de las imágenes utilizadas en el trabajo. Si bien esto contribuyó a aumentar la calidad de los datos, también requirió destinar una gran cantidad de tiempo a la tarea.
- Modificaciones sobre lo planeado: los requerimientos originales incluían la clasificación en grupos etarios de los ciclistas. Debido a la practicamente nula representatividad en los datos de algunos de los grupos etarios definidos, este requerimiento no fue considerado para la realización del trabajo.
- Utilidad de las técnicas: resultó de gran valor la gestión de dependencias utilizando la herramienta poetry y el registro de experimentos utilizando la herramienta mlflow. También fue importante la división del trabajo en

cuatro tareas principales, etiquetado y entrenamiento para detección y clasificación, así como en múltiples subtareas. Finalmente, la parametrización de las tareas mediante la utilización de un archivo de configuración externo permitió realizar numerosos experimentos en cada una de las etapas. En contraste, si bien fue importante para la realización de pruebas y entrenamiento de los modelos, la utilización de Google Colab demandó tiempo considerable de configuración específica para cada una de las tareas.

En términos generales, se completó el trabajo según lo planeado. El aporte principal del proyecto radica en la exploración de técnicas clásicas para facilitar el trabajo de etiquetado de imágenes, así como la comparativa de diferentes arquitecturas para tareas usuales de detección y clasificación.

5.2. Próximos pasos

El sistema desarrollado para el presente trabajo posee ciertas líneas de continuación y mejora que vale la pena destacar.

- Incorporación de nuevas clases vehiculares: la utilización del sistema en producción puede contribuir a aumentar la cantidad de datos disponibles para futuras iteraciones de entrenamiento. En este sentido, la representatividad de algunas clases vehiculares incluidas en este trabajo puede incrementarse. Aún más, ciertas clases no consideradas dentro de los requerimientos pueden incorporarse, como los monopatines eléctricos. Este modo de transporte se encuentra en expansión en la ciudad de Rosario, por lo que puede volverse importante su registro en el futuro cercano.
- Mejora de la capacidad de generalización del modelo de clasificación: si bien los modelos de clasificación entrenados poseen una buena capacidad de inferencia frente a condiciones novedosas debido a su entrenamiento en imágenes mixtas de ciclistas y peatones, el fuerte de los mismos se encuentra en su utilización en las primeras. Continuar el etiquetado para aumentar la representatividad de personas en diversas situaciones urbanas permitirá la utilización del modelo para eventos en los cuales la movilidad ciclista no sea la principal.

Además, durante la implementación del trabajo surgieron algunos interrogantes que pueden ser el puntapié de investigaciones futuras.

- Evaluación de las técnicas de muestreo utilizadas para el etiquetado de imágenes: para el etiquetado de fotogramas utilizados en el entrenamiento de modelos de detección se utilizaron numerosas heurísticas combinando modelos CNN de licencia abierta y técnicas de visión por computadora clásica. Si bien la aplicación de estos procedimientos se funda en el sentido común, no deja de ser intrigante cuál es el aporte de estos en el desempeño final de los modelos. Una línea de investigación posible implica la comparación del entrenamiento de modelos utilizando las técnicas mencionadas respecto al etiquetado de imágenes obtenidas por un muestreo aleatorio. Los resultados de tal empresa pueden contribuir a considerar los beneficios obtenidos, si hay, en relación con el tiempo empleado.

- Gestión de la incertidumbre en las tareas de clasificación binaria: durante el etiquetado de instancias según género para entrenar modelos de clasificación binaria, numerosas imágenes resultaron indescifrables para el ojo humano y fueron descartadas. Resulta interesante la posibilidad de explorar la utilización de estas instancias durante el entrenamiento (ya sea como una tercera clase, o como una entrada con confianza similar para ambas clases). Este mecanismo podría evitar que un modelo asigne clases aleatoriamente en producción al poder designar una instancia como irreconocible.

Bibliografía

[1] Ciudad de Rosario. *Ente de la Movilidad*. http://emr.gob.ar/info_emr.php. Oct. de 2025. (Visitado 20-10-2025).

[2] Ciudad de Rosario. *Centro Integrado de Operaciones*. https://www.rosario.gob.ar/ArchivosWeb/personal/Que_es_el_CIOR.pdf. Oct. de 2025. (Visitado 20-10-2025).

[3] Caban et al. *The comparison of automatic traffic counting and manual traffic counting*. https://iopscience.iop.org/article/10.1088/1757-899X/710/1/012041. 2019.

[4] Bugdol et al. *Vehicle detection system using magnetic sensors*. https://www.researchgate.net/publication/287944531_Vehicle_detection_system_using_magnetic_sensors. 2014.

[5] Ren et al. *Vehicle Tracking Through Magnetic Sensors: ACase StudyofTwo-lane Road*. https://arxiv.org/pdf/2209.09020. 2018.

[6] Vasu et al. *Vehicle-counting with Automatic Region-of-Interest and Driving-Trajectory detection*. https://arxiv.org/abs/2108.07135. 2021.

[7] Ribeiro et al. *Combining YOLO and Visual Rhythm for Vehicle Counting*. https://arxiv.org/abs/2501.04534. 2025.

[8] Yoo et al. *Vehicle Counting using Computer Vision: A Survey*. https://ieeexplore.ieee.org/document/9824432. 2022.

[9] Gildea et al. *Computer vision-based assessment of cyclist-tram track interactions for predictive modeling of crossing success*. https://www.sciencedirect.com/science/article/pii/S0022437523001482. 2023.

[10] Masalov et al. *Specialized Cyclist Detection Dataset: Challenging Real-World Computer Vision Dataset for Cyclist Detection Using a Monocular RGB Camera*. https://ieeexplore.ieee.org/document/8813814. 2019.

[11] Glenn Jocher, Ayush Chaurasia y Jing Qiu. *Ultralytics YOLOv8*. Ver. 8.0.0. 2023. URL: https://github.com/ultralytics/ultralytics.

[12] Jacob Robinson y Roboflow. *RF-DETR: Real-Time Object Detection Model*. 2024. URL: https://github.com/roboflow/rf-detr.

[13] Nicolai Wojke, Alex Bewley y Dietrich Paulus. «Simple Online and Realtime Tracking with a Deep Association Metric». En: *arXiv preprint arXiv:1703.07402* (2017). URL: https://arxiv.org/abs/1703.07402.

[14] Yifu Zhang et al. «ByteTrack: Multi-Object Tracking by Associating Every Detection Box». En: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2022. URL: https://arxiv.org/abs/2110.06864.

[15] South Moravia Brno. *Data from sky*. https://datafromsky.com/traffic-monitoring/. Oct. de 2025. (Visitado 20-10-2025).

[16] United Kingdom. *Vision track*. https://visiontrack.com/automotive/. Oct. de 2025. (Visitado 20-10-2025).

[17] United States S.F. *Meegle*. https://www.meegle.com/en_us/topics/computer-vision/computer-vision-in-vehicle-tracking. Oct. de 2025. (Visitado 20-10-2025).

[18] Gendy et al. *Advancements in Computer Vision: A Comprehensive Survey of Image Processing and Interdisciplinary Applications*. https://drpress.org/ojs/index.php/ajst/article/view/27303. 2024.
[19] Liu et al. *Smart Traffic Monitoring System using Computer Vision and Edge Computing*. https://arxiv.org/abs/2109.03141. 2021.
[20] Bureau of Transportation Statistics. *Computer Vision for Traffic Monitoring*. https://rosap.ntl.bts.gov/view/dot/79662/dot_79662_DS1.pdf. 2024.
[21] Richard Szeliski. *Computer Vision: Algorithms and Applications, 2nd ed.* https://szeliski.org/Book/. 2022.
[22] David G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf. 2004.
[23] OpenCV Team. *Feature Matching with FLANN*. https://docs.opencv.org/3.4/d5/d6f/tutorial_feature_flann_matcher.html. 2020.
[24] Community contribution. *Object detection page on wikipedia*. https://en.wikipedia.org/wiki/Object_detection. 2025. (Visitado 20-10-2025).
[25] Community contribution. *Image recognition page on wikipedia*. https://en.wikipedia.org/wiki/Computer_vision#Recognition. 2025. (Visitado 20-10-2025).
[26] Mingxing Tan y Quoc V. Le. «EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks». En: *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019, págs. 6105-6114. URL: https://proceedings.mlr.press/v97/tan19a.html.
[27] Kaiming He et al. «Deep Residual Learning for Image Recognition». En: *arXiv preprint arXiv:1512.03385* (2015). URL: https://arxiv.org/abs/1512.03385.
[28] Community contribution. *Transfer learning page on wikipedia*. https://en.wikipedia.org/wiki/Transfer_learning. 2025. (Visitado 20-10-2025).
[29] ImageNet Team. *ImageNet Library*. https://www.image-net.org/. 2009.
[30] COCO Team. *COCO Library*. https://cocodataset.org/#home. 2014.
[31] Community contribution. *Data augmentation page on wikipedia*. https://en.wikipedia.org/wiki/Data_augmentation. 2025. (Visitado 20-10-2025).
[32] OpenCV Team. *OpenCV Library*. https://opencv.org/. 2024.
[33] PyTorch Team. *PyTorch: An open source machine learning framework*. https://pytorch.org/. 2024.
[34] Ultralytics. *Ultralytics YOLOv8*. https://github.com/ultralytics/ultralytics. 2024.
[35] Roboflow. *Roboflow RF-DETR*. https://github.com/roboflow/rf-detr. 2024.
[36] Roboflow. *Roboflow Supervision*. https://github.com/roboflow/supervision. 2024.