UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

NUEVO MÉTODO DE DETECCIÓN DE PEATONES DE DOS ETAPAS UTILIZANDO UNA NUEVA CABEZA DE CLASIFICACIÓN CON TÉCNICAS DE GENERALIZACIÓN DE DOMINIO

TESIS PARA OPTAR AL GRADO DE DOCTOR EN INGENIERÍA ELÉCTRICA

DANIEL ALEJANDRO SCHULZ MELGAREJO

PROFESOR GUÍA:
CLAUDIO PÉREZ FLORES

MIEMBROS DE LA COMISIÓN:
PAMELA GUEVARA ALVEZ
JOSÉ DELPIANO COSTABAL
CESAR AZURDIA MEZA

Este trabajo ha sido parcialmente financiado por ANID a través del proyecto FONDECYT 1231675, y de los proyectos basales AFB220002 e IMPACT #FB210024, y por el Departamento de Ingeniería Eléctrica de la Universidad de Chile.

SANTIAGO DE CHILE
2024

RESUMEN DE LA TESIS PARA
OPTAR AL GRADO DE DOCTOR
EN INGENIERÍA ELÉCTRICA
POR: Daniel Alejandro Schulz Melgarejo
FECHA: 2024
PROFESOR GUÍA: Claudio Pérez Flores

NUEVO MÉTODO DE DETECCIÓN DE PEATONES DE DOS ETAPAS UTILIZANDO UNA NUEVA CABEZA DE CLASIFICACIÓN CON TÉCNICAS DE GENERALIZACIÓN DE DOMINIO

La detección de peatones es una tarea importante en visión computacional, debido a muchas aplicaciones prácticas que requieren alto desempeño. Los métodos basados en *Deep Learning* han sido muy efectivos en esta tarea en los últimos años. Este trabajo presenta un nuevo detector de peatones de dos etapas basado en redes neuronales, que usa una nueva cabeza de clasificación, agregando la función *Triplet Loss* a las funciones de regresión de *bounding boxes* y clasificación estándar, para mejorar la capacidad de generalización de un detector de objetos existente. La función *Triplet Loss* se aplica a las características correspondientes a las regiones de interés generadas por la red de propuestas de región, de manera de agrupar los ejemplos de peatones en el espacio de características. Se realizaron experimentos con dos detectores, Faster R-CNN y Cascade R-CNN, utilizando el *backbone* HRNet preentrenado con ImageNet. Los mejores resultados se obtuvieron utilizando un *pipeline* de entrenamiento progresivo. El método desarrollado obtuvo resultados en el estado del arte, en el *benchmark* CityPersons, con un rendimiento destacado en la partición *Heavy*, la más difícil. Este trabajo demuestra la eficacia de la función *Triplet Loss* para mejorar la capacidad de generalización de los detectores de peatones existentes.

Agradecimientos

Este trabajo ha sido parcialmente financiado a través del proyecto FONDECYT 1231675 de la Agencia Nacional de Investigación y Desarrollo (ANID), además del financiamiento Basal de ANID, Proyectos AFB220002 e IMPACT #FB210024, y por el Departamento de Ingeniería Eléctrica de la Universidad de Chile.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

1.1. Motivación

El problema de detección de objetos en ambientes no controlados ha sido un tema recurrente en la comunidad científica de visión computacional en los últimos años. Se han realizado bastantes esfuerzos por resolverlo, y se ha avanzado bastante en el tema, pero en términos generales, el problema general de detección de objetos sigue estando abierto, aunque se ha progresado en resolver versiones acotadas del mismo. Este problema consiste en encontrar la posición y el tamaño de distintos tipos de objetos en una imagen digital, distinguirlos entre si y también del fondo, lo que es complejo ya que el fondo suele ser bastante heterogéneo y los objetos presentan gran variabilidad dentro de una misma clase, dadas las distintas condiciones de iluminación y adquisición. En esta línea, avances significativos se han logrado en el último tiempo, en específico en la última década, donde el Aprendizaje Profundo (*Deep Learning* en inglés), ha sido una de las técnicas más utilizadas en visión computacional, debido a su desempeño superior en comparación con los métodos de la generación anterior, los que correspondían generalmente a modelos donde la extracción de características se optimizaba en forma manual de acuerdo al problema, y luego se aplicaba algún modelo de clasificación. En los últimos años se han desarrollado muchas aplicaciones que utilizan *Deep Learning* y Redes Neuronales Convolucionales, aparte de la ya mencionada detección de objetos, como por ejemplo, Reconocimiento Facial [1], Clasificación de Género [2, 3], Reconocimiento de Iris [4], Clasificación de Litologías de Rocas [5], Recuperación de Imágenes de Marcas Registradas [6], Algoritmos Genéticos aplicados a CNNs [7, 8] y Segmentación Semántica [9].

En el área de detección de objetos, se han ido mejorando los resultados en forma incremental mediante el uso de *Deep Learning*, aplicando dichas técnicas a una serie de *benchmarks*,

por ejemplo, Pascal VOC [10] o MS-COCO [11]. Aquí se pueden destacar los trabajos de Girshick et al. [12, 13], Ren et al. [14], Cai et al. [15] los cuales desarrollaron métodos genéricos de detección de objetos, con resultados ampliamente superiores a los previamente publicados con técnicas de extracción de características manuales.

Una de las tareas clave en visión computacional es la detección de peatones, donde se han desarrollado varios modelos en el último tiempo [16–18]. El rendimiento muestra una mejora constante con el tiempo, especialmente con el auge de los métodos basados en *Deep Learning*, donde en ciertos *benchmarks*, el desempeño se acerca al rendimiento humano [19], por ejemplo, en el *benchmark* Caltech [20]. Muchas aplicaciones del mundo real requieren un alto rendimiento en la detección de peatones, como la conducción de vehículos autónomos, la navegación robótica, la vigilancia por video, el reconocimiento de acciones y el *tracking* [21–23]. En el caso de la conducción de vehículos autónomos, un método robusto de detección de peatones es un elemento clave a desarrollar, de modo de evitar posibles accidentes de tránsito. A modo de ejemplo, estudios muestran que los peatones tendieron a sufrir más lesiones cuando ocurría un choque entre vehículos y peatones. Según la Administración Nacional de Seguridad del Tráfico en Carreteras (NHTSA), los accidentes de tráfico en los Estados Unidos generaron 7.388 muertes de peatones, mientras que 60.577 peatones resultaron heridos, durante el año 2021 [24]. Mientras tanto, en Europa, durante el año 2020, se reportaron 3.608 peatones fallecidos, lo que representa el 19 % del total de las muertes generadas en carretera [25].

La naturaleza de las posibles aplicaciones que involucran la detección de peatones hacen necesario tener un desempeño con alta precisión, y que sea capaz de operar en tiempo real [26]. Algunos de los principales desafíos para los métodos de detección de peatones, es que los individuos en las imágenes presentan diferentes escalas, distintos tipos de oclusión, diversas razones de aspecto, etc. [27], lo que genera una dificultad extra a la hora de aplicar esta clase de métodos.

El problema de detección de peatones puede ser pensado como un subproblema del problema más amplio de detección de objetos genéricos, por lo tanto, muchos de estos métodos se pueden adaptar para detectar peatones en lugar de objetos genéricos [28]. En los enfoques con *Deep Learning*, se han usado dos familias de métodos para la detección de objetos: detectores de una etapa, como SSD [29] y YOLO [30], y detectores de dos etapas, con métodos como Faster R-CNN [14] y Cascade R-CNN [15].

Los detectores de objetos de dos etapas incluyen una tarea intermedia de generar propuestas de región, para luego aplicar una etapa de clasificación de objetos a cada región propuesta [15]. En general, los métodos de una etapa suelen ser más rápidos que los de dos etapas, sin embargo, los métodos de dos etapas logran un rendimiento más robusto [15]. Por otro lado,

la detección de peatones presenta sus propios desafíos, en comparación con la tarea general de detección de objetos. A modo de ejemplo, las muestras negativas difíciles de las regiones pertenecientes al fondo, suelen confundir a los métodos de detección de peatones [31].

La capacidad de los métodos actuales de última generación, es decir, los del estado del arte, para obtener un alto desempeño en pruebas con conjuntos de datos cruzados, es un problema que aún no ha sido resuelto. Hasan et al. [19] mostró que los métodos del estado del arte para detectar peatones no funcionan bien cuando se cambia el dominio, lo que deteriora las métricas de desempeño al evaluar en escenarios de pruebas con bases de datos cruzadas.

En este contexto, el cambio de dominio (en inglés *Domain change* o *Domain shift*) se define como el problema que surge cuando ocurre un cambio en la distribución entre un conjunto de datos de entrenamiento (fuente) y un conjunto de datos de prueba (objetivo). Este problema es causado porque la mayoría de los métodos de aprendizaje estadístico se basan en la suposición de que tanto los datos fuente como los datos objetivo son independientes e idénticamente distribuidos, ignorando los escenarios fuera de la distribución que comúnmente se encuentran en la práctica. Esto conduce a una disminución del rendimiento cuando un algoritmo entrenado solo con datos fuente se prueba en un dominio objetivo fuera de la distribución. Este problema ha limitado la implementación de modelos a gran escala [32]. Por otro lado, la generalización de dominio (*Domain generalization* en inglés) es un problema de aprendizaje automático en el que el modelo aprende a partir de datos de entrenamiento etiquetados en tareas relacionadas, esperando que generalice a una tarea de predicción futura sin acceso a datos etiquetados [33]. Este concepto se introdujo para abordar los desafíos del cambio de dominio y la falta de datos del dominio objetivo. El propósito es entrenar un modelo utilizando datos de uno o más dominios fuente relacionados pero distintos, de modo que pueda generalizar y desempeñarse de manera efectiva en cualquier dominio objetivo fuera de la distribución [32].

El método propuesto en este trabajo tiene como objetivo mejorar algunas de las limitaciones mencionadas, basándose explícitamente en enfoques de adaptación de dominio, en este caso, usando la función *Triplet Loss*. Esta familia de funciones de pérdida ha sido usada exitosamente en otras tareas de visión computacional, por ejemplo, reconocimiento facial [1, 34–38], re-identificación de personas [39–44], generalización de dominio [45–49], entre otros. Aquí, se usa *Triplet Loss* como una función de pérdida adicional para la cabeza de clasificación, después de la extracción de las regiones de interés (ROIs) en un enfoque de detección de peatones de dos etapas. Se utilizó esta función de pérdida, en forma adicional a las funciones de pérdida de clasificación estándar y la función de pérdida de *bounding boxes*.

1.2. Hipótesis

- Es posible mejorar el desempeño de un detector de objetos de dos etapas, mediante la aplicación de técnicas de generalización de dominio en lugares específicos del *pipeline* de detección.
- La adición de la función *Triplet Loss* a una cabeza de clasificación de un detector de objetos de dos etapas, aplicada sobre los *embeddings* generados por la capa de extracción de ROI, permitiría mejorar el desempeño del método.
- Es posible mejorar el desempeño de los detectores de peatones de dos etapas, Faster R-CNN y Cascade R-CNN, reemplazando la(s) cabeza(s) de clasificación estándar, por una nueva cabeza de clasificación, la cual aplica técnicas de generalización de dominio.
- La generalización de dominio de un detector de peatones de dos etapas con la nueva cabeza de clasificación, mejora usando un flujo de entrenamiento progresivo, es decir, entrenar en bases de datos más lejanas del dominio objetivo, e irse acercando a este a medida que se agregan más dominios al flujo de entrenamiento.

1.3. Objetivo General

El objetivo general de esta tesis es desarrollar un método de detección de peatones de dos etapas, basado en redes neuronales convolucionales, aplicando técnicas de generalización de dominio, minimizando la tasa de error de detecciones, y maximizando la tasa de aciertos.

1.4. Objetivos Específicos

- Desarrollar una nueva cabeza de clasificación, para ser usada con detectores de objetos de dos etapas, que utilice la función de costo *Triplet Loss*. Esto con el objetivo de mejorar el rendimiento en la tarea de detección de peatones.
- Diseñar e implementar una nueva cabeza de clasificación para un detector de objetos basado en Faster R-CNN, reemplazando la cabeza estándar de clasificación por la cabeza desarrollada, con el objetivo de mejorar el desempeño en la detección de peatones.
- Diseñar e implementar una nueva cabeza de clasificación para un detector de objetos basado en Cascade R-CNN, reemplazando una o varias de las tres cabezas estándar

de clasificación, por la cabeza desarrollada, de manera que mejore el desempeño en la detección de peatones.

- Medir el impacto de esta nueva cabeza de clasificación, tanto en el detector Faster R-CNN como Cascade R-CNN, usando bases de datos internacionales, y comparar los resultados obtenidos con el estado del arte.
- Medir el impacto en el desempeño de la aplicación de un flujo de entrenamiento progresivo para el detector desarrollado, entrenando al principio en bases de datos más lejanas del dominio objetivo, para luego irse acercando a este a medida que se agregan más dominios al flujo de entrenamiento.

1.5. Principales Contribuciones

La principal contribución de este trabajo es el desarrollo de nuevo enfoque para la detección de peatones, mediante el uso de una nueva cabeza de clasificación para detectores de objetos de dos etapas, que añade una nueva función de pérdida, complementando las funciones de pérdida de clasificación y regresión de *bounding boxes*, usadas como etapa final en este tipo de detectores. Esta nueva cabeza incorpora la función *Triplet Loss*, aplicada a los *embeddings* generados por el extractor ROI, en una red neuronal de dos etapas. El objetivo de incorporar esta nueva cabeza es mejorar la capacidad de generalización de dominio de un sistema de detección de peatones. La adición de *Triplet Loss* resultó en una nueva función de pérdida combinada, que disminuye la varianza interclase de las características, mejorando así el rendimiento de la detección de objetos en comparación con el estado del arte. Según la revisión de la literatura realizada, el enfoque propuesto no se ha utilizado previamente en detección de peatones. El diseño propuesto tiene como objetivo mejorar el desempeño en escenarios de pruebas de bases de datos cruzadas, maximizando explícitamente la distancia interclase y minimizando la distancia intraclase, usando un término de margen para determinar la frontera de decisión entre ejemplos positivos y negativos. De esta manera, los peatones provenientes de diferentes dominios (Bases de datos) se agrupan en el espacio de características.

Los resultados obtenidos superan los publicados en el estado del arte para la base de datos de CityPersons, especificamente en la partición más difícil disponible, utilizando el entrenamiento cruzado en un conjunto diferente, es decir, no entrenado explícitamente en ninguna partición del conjunto objetivo. La cabeza desarrollada podría usarse como una nueva dirección para mejorar el rendimiento en otros detectores de peatones con arquitecturas compatibles, o en otras tareas de detección de objetos, considerando aplicaciones del mundo

real como la conducción autónoma y la vigilancia por video.

Finalmente, se puede mencionar que este trabajo dio como resultado una publicación en la revista Sensors, la cual se encuentra indexada en Web of Science (WoS). Esta publicación destaca la validez y relevancia científica de la investigación desarrollada, contribuyendo al conocimiento en el ámbito de la detección de peatones. La referencia al trabajo es la siguiente: *Schulz, D., Perez, C. A., "Two-stage pedestrian detection model using a new classification head for domain generalization", Sensors, vol. 23, no. 23, p. 9380, 2023.*

1.6. Estructura de la Tesis

El documento está organizado de la siguiente manera: El capítulo 2 muestra el trabajo relacionado, el capítulo 3 muestra el método desarrollado, los detalles de las bases de datos, y los experimentos diseñados, el capítulo 4 muestra los resultados obtenidos, y realiza un análisis de estos, finalmente, el capítulo 5 muestra las conclusiones del trabajo desarrollado, y el trabajo futuro para seguir esta línea de investigación.

Capítulo 2

Estado del Arte

2.1. Detección de peatones

Los primeros enfoques usados para la detección de peatones y, en general, para la detección de objetos, utilizaban el paradigma de ventanas deslizantes, las cuales se aplicaban en todas las ubicaciones y todas las escalas de las imágenes analizadas. Aquí se puede mencionar un hito importante con el trabajo desarrollado por Viola y Jones [50], el cual permitió por primera vez hacer procesamiento en tiempo real, en este caso, aplicado a la detección de rostros. Esto debido al esquema que ellos presentaron, usando el concepto de imagen integral, con el objetivo de acelerar el cálculo de las características de Haar, para luego aplicar una estructura de clasificación en cascada, obteniendo una detección eficiente basada en clasificadores *AdaBoost*. Un enfoque similar fue aplicado por los autores a la detección de peatones [51]. El esquema en cascada se puede observar en la Figura 2.1. Otro de los trabajos pioneros que utilizó este enfoque fue desarrollado por Papageorgiou y Poggio [52], quienes utilizaron una combinación de wavelets de Haar multiescala y máquinas de soporte vectorial (SVM). En el trabajo desarrollado por Dalal y Triggs [53], se utilizaron características basadas en el histograma de gradientes orientados (HOG) y SVM para la detección de personas, superando al momento de su publicación, a las características existentes en el estado del arte, basadas en intensidad. Dollár et al. [54] propusieron características de canal agregado (ACF) para la detección de peatones, mejorando la velocidad sin sacrificar el rendimiento, aproximando características en una pirámide muestreada finamente. Felzenszwalb et al. [55] desarrollaron un método para la detección de objetos, basado en una mezcla de modelos deformables multiescala, utilizando un entrenamiento discriminatorio de clasificadores que utilizan información latente.

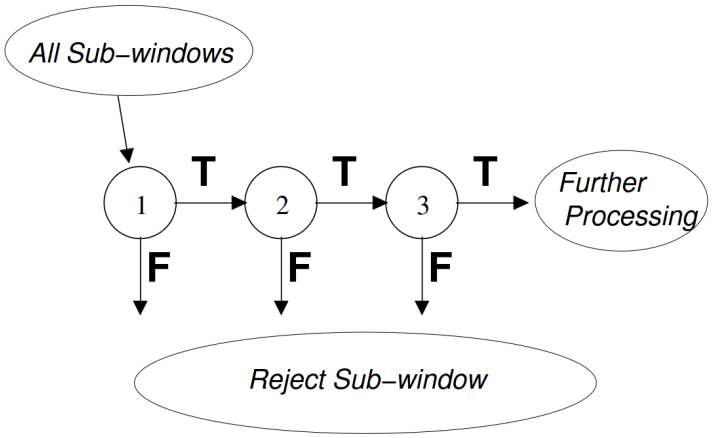


Figura 2.1: Diagrama esquemático del detector en cascada de Viola y Jones. Tomado de [50].

Actualmente, los métodos del estado del arte, usan por lo general *Deep Learning*, principalmente con redes neuronales convolucionales (CNN). Estos métodos mejoraron considerablemente el rendimiento para el problema de detección de objetos [13, 14, 29, 56]. Muchas de las técnicas genéricas de detección de objetos se utilizaron como base para los métodos modernos de detección de peatones. Uno de los primeros métodos basados en CNN fue propuesto por Angelova et al. [57]. Se usó una combinación de clasificadores en cascada y características de redes neuronales profundas, lo que dio como resultado un método rápido y preciso, que se ejecuta en tiempo real, cuyo desempeño fue evaluado en la base de datos de detección de peatones Caltech.

Cai et al. [58] desarrollaron un algoritmo basado en boosting llamado CompACT, utilizando un diseño en cascada, usando optimización Lagrangiana de una función de costo que tiene en cuenta tanto la precisión como la complejidad, y permitiendo el uso de características con diferentes complejidades en un solo detector. Esto incluye una cascada que combina CNN con un mecanismo de propuesta de objetos, obteniendo buenos resultados en las bases de datos Caltech y KITTI.

En el trabajo de Hosang et al. [59], se realizaron varios experimentos con diferentes arquitecturas de redes neuronales convolucionales disponibles en ese momento, evitando diseños personalizados adaptados para el problema de detección de peatones. Los autores muestran resultados competitivos en las bases de datos de referencia Caltech y KITTI. Zhang et al. [31] propusieron un método basado en una Red de Propuesta de Regiones (RPN), seguido por *Boosted Forests* en cascada, para clasificar las propuestas de regiones. De esta manera, las características de resoluciones arbitrarias de cualquier capa se combinan, y se luego se realiza una minería de negativos difíciles, superando las limitaciones del método original Faster

R-CNN. Brazil et al. [60] propusieron un marco de trabajo de infusión multitarea, de manera de realizar en forma simultánea segmentación semántica y detección de peatones, obteniendo resultados comparables al estado del arte en la base de datos Caltech y un rendimiento competitivo en el conjunto de datos KITTI.

Zhou y Yuan [61] desarrollaron un método para detectar peatones, y también estimar el grado de oclusión, usando una red neuronal convolucional con dos ramas. La primera rama era para la estimación del cuerpo completo del peatón, y la segunda rama era para la estimación de la parte visible del cuerpo. Ambas ramas producen salidas que se complementan entre sí, mejorando el rendimiento de la detección. El método se evaluó en las bases de datos Caltech y CityPersons, obteniendo excelentes resultados en la detección de peatones, tanto ocluidos y sin oclusión.

Liu et al. [17] desarrollaron un método de detección de una etapa (SSD), llamado *Asymptotic Localization Fitting* (ALF), que apila una serie de predictores para evolucionar los *bounding boxes* ancla predeterminados de SSD, paso a paso, acercándolos a los *bounding boxes* anotados, y luego utiliza una arquitectura de detección de peatones llamada ALFNet. Este método mejoró la precisión, y al mismo tiempo, mantenía la eficiencia de los detectores de una sola etapa, logrando un rendimiento comparable al estado del arte en las bases de datos CityPersons y Caltech.

Liu et al. [62] propusieron el método CSP (Sigla para *Center-Scale Predictor*), en el que la detección de peatones se considera una detección de características semánticas de alto nivel, prediciendo el centro y la escala del peatón utilizando CNNs. Este método simple alcanzó resultados competitivos tanto en detección como en tiempo de cómputo en varias bases de datos de detección de peatones. El esquema de este método se puede apreciar en la Figura 2.2.

Yin et al. [63] propusieron un método llamado DA-Net, que utiliza un detector de dos etapas basado en *Feature Pyramid Networks* (FPN), e incorpora un Bloque Conectado Densamente (DCB), que comprende un módulo de atención por canal (CWAM) y un módulo de atención global (GAM). Al agregar varios DCBs para profundizar la red, las capas de predicción pueden capturar información semántica más rica sobre los ejemplos objetivo, lo que conduce a una localización más precisa de los mismos. El método fue evaluado en CUHK y CityPersons, en este último se realizó la evaluación en el subconjunto *Heavy*, obteniendo buenos resultados.

En el trabajo de Lin et al. [64] se desarrolló un detector de peatones llamado PedJoint-Net, que ejecuta regresión en forma simultánea para dos tipos de *bounding boxes*, una que

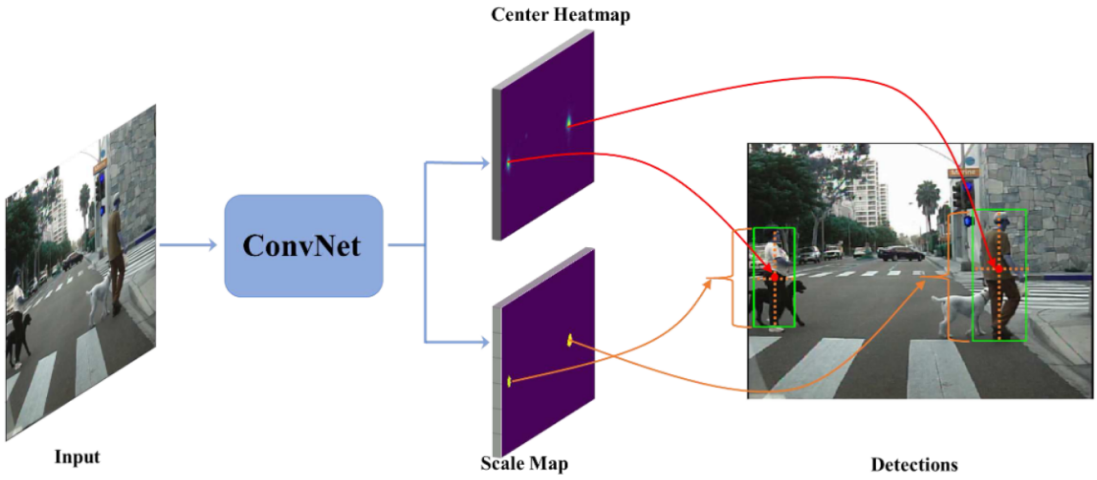


Figura 2.2: Diagrama del método CSP. Se pueden apreciar los canales para localizar los centros (Puntos rojos) y predecir las escalas (Líneas punteadas amarillas). Tomado de [62].

representa las regiones de cabeza-hombros, y otra para representar el cuerpo completo, basados en una estructura de detección de objetos factible, logrando excelente rendimiento en la detección de peatones, tanto con oclusiones como completamente visibles, especialmente en circunstancias que implican oclusión. El método fue evaluado en las bases de datos CUHK-SYSU, TownCentre y CityPersons.

Cai et al. [65] desarrollaron un detector de peatones sin anclas ni propuestas de región, denominado *Pedestrian-as-Points Network* (PP-Net). Este método es capaz de encontrar un mejor equilibrio entre precisión y eficiencia. Para este propósito, los peatones fueron modelados como puntos únicos, es decir, el punto central de la instancia, y luego predijeron la escala del peatón en cada punto central detectado. Para evitar la pérdida de información de alto nivel en el camino descendente de la red, se construyó un módulo de guía profunda (DGM) en la parte superior del *backbone*. Con este enfoque, se obtuvieron resultados de estado del arte en las bases de datos Caltech y CityPersons. Para lograr este objetivo, se captura información de alto nivel en el proceso de construir una red de pirámide de características (FPN). Utilizando la supresión de no máximos (NMS) como post-procesamiento, los resultados obtenidos fueron mejores que muchos métodos del estado del arte en desafiantes bases de datos de detección de peatones.

Li et al. [66] desarrollaron un detector de peatones basado en YOLOv7, con el objetivo de mejorar la detección de peatones ocluidos. Para esto, el *backbone* predeterminado de YOLOv7 fue reemplazado por un *backbone* liviano basado en MobileNetV3. Luego, se utilizó una estructura de pirámide de características de alta resolución para mejorar el desempeño en peatones no detectados con oclusión, empleando un mecanismo de atención para reducir las

bounding boxes redundantes. El método se aplicó a la base de datos de datos CrowdHuman, obteniendo resultados prometedores.

Liu et al. [67] desarrollaron un detector de peatones para entornos de tráfico con niebla, llamado YOLO-GW, utilizando el algoritmo de de-fogging de canal oscuro, en conjunto con un detector YOLOv7. Además, se agregó un módulo de atención eficiente de canal (*Efficient Channel Attention* ECA, en inglés) y una cabeza de detección, con el objetivo de mejorar la clasificación y regresión de objetos. Los resultados mostraron una mejora en la velocidad de procesamiento de un 63,08 %, y una mejora en la detección utilizando la métrica mAP (*mean Average Precision*), aumentando en un 9,06 %.

Zhang et al. [68] evaluaron el desempeño de un detector de peatones, basado en Faster R-CNN. Para esto, usaron una nueva base de datos, llamada CityPersons, que consiste en anotaciones de personas en imágenes provenientes de la base de datos Cityscapes. La diversidad de esta base de datos permitió entrenar un modelo único que generaliza bien sobre diversos conjuntos de pruebas.

Actualmente existen varias bases de datos que se utilizan como *benchmark* para evaluar la detección de peatones. Algunos de ellas son la base de datos Daimler [69], INRIA [53], ETH [70], TUDBrussels [71], y Wider Pedestrian [72]. Sin embargo, muchos de estas bases de datos fueron capturadas en escenarios de vigilancia, por lo que no son adecuadas para aplicaciones de conducción autónoma.

Por otro lado, existen otras bases de datos específicamente diseñadas para la conducción autónoma, como Caltech [20], KITTI [73], CityPersons [68], y EuroCity Persons [74]. Estas bases de datos contienen una amplia variedad de situaciones en las que se pueden encontrar peatones en la vía pública, lo que las hace adecuadas para evaluar la capacidad de los algoritmos de detección de peatones en entornos más complejos.

Es importante tener en cuenta que, aunque estas bases de datos se han utilizado ampliamente en la evaluación de algoritmos de detección de peatones, la elección de la base de datos depende del problema específico que se esté abordando y del contexto en el que se vaya a utilizar el algoritmo. Por lo tanto, es fundamental evaluar el rendimiento de los algoritmos en una amplia variedad de bases de datos para poder determinar su capacidad de generalización y asegurar que sean adecuados para el propósito previsto.

2.2. Generalización de dominio en detectores de peatones

Respecto a la generalización de dominio en detectores de peatones, Braun et al. [74] evaluaron la capacidad de generalización de cuatro detectores de objetos genéricos basados en *Deep Learning*, aplicados a la detección de peatones: Faster R-CNN, R-FCN, SSD y YOLOv3, utilizando una nueva base de datos, de mayor tamaño a las existentes en la literatura, llamada, EuroCity Persons. Estudiaron el efecto en el rendimiento del detector para muchas variables relacionadas con el tamaño del conjunto de entrenamiento, la diversidad y detalle de la base de datos, y la calidad de las anotaciones. Se observó que el pre-entrenamiento con conjuntos muy grandes supera el uso de solo conjuntos de entrenamiento específicos.

Chao et al. [75] también evaluaron la capacidad de generalización de detectores de personas. Para lograr este objetivo, los autores crearon un conjunto de gran tamaño, llamado CrowdHuman, y evaluaron la generalización entre distintas bases de datos, obteniendo nuevos resultados del estado del arte en las bases de datos Caltech, CityPersons y Brainwash. Demostraron que el nuevo conjunto de datos propuesto podría utilizarse para el pre-entrenamiento de detectores de personas.

Hasan et al. [19] llevaron a cabo una evaluación exhaustiva utilizando varios métodos de detección de peatones de última generación. Probaron sus capacidades de generalización de dominio para ciertos métodos populares de detección de objetos generales, no diseñados específicamente para la detección de peatones. En general, los métodos alcanzaron un buen desempeño de detección cuando se entrenaron y probaron en la misma base de datos, pero los resultados empeoraron cuando se evaluaron en bases de datos diferentes. Los autores probaron la generalización de los detectores de peatones en diferentes bases de datos que contienen diferentes características y condiciones de captura. Descubrieron que estos métodos generales tuvieron un mejor desempeño que los detectores de peatones específicos cuando se realizaron experimentos en bases de datos cruzadas, lo que indica que son más adecuados para la detección de peatones en diferentes situaciones. Sin embargo, los autores también encontraron que los métodos generales no siempre superan a los detectores de peatones específicos. Estos resultados sugieren que los métodos generales pueden ser útiles para la detección de peatones en una variedad de situaciones, pero aún pueden requerir ajustes específicos para una base de datos en particular, de manera de obtener un mejor rendimiento de detección.

De acuerdo con la revisión bibliográfica realizada, se observa que muchos métodos alcanzaron un buen rendimiento en la detección de peatones cuando fueron entrenados y probados en

las mismas bases de datos. Sin embargo, los resultados empeoraron significativamente cuando se realizaron experimentos evaluando en una base de datos distinta a la donde el método fue entrenado. La capacidad de desempeñarse bien en escenarios desconocidos es crucial para que los métodos desarrollados en la literatura puedan ser implementados en aplicaciones del mundo real. Por ejemplo, el detector de peatones que posee un vehículo autónomo es efectivamente un escenario cruzado, en el cual la mayoría de los datos observados son nuevos para el detector. De esto se puede inferir que la generalización para los detectores de peatones, evaluando en bases de datos distintas a las de entrenamiento, es un problema que aún no se ha resuelto en el estado del arte actual. Otro problema importante presente en la revisión de la literatura es la capacidad limitada para detectar peatones con un alto grado de oclusión. Esto se puede confirmar al analizar los resultados para la partición *Heavy* del *benchmark* CityPersons, cuyos resultados actuales en el estado del arte aún están lejos de los resultados obtenidos para la partición *Reasonable*, un escenario más fácil en el cual los métodos obtienen resultados que pueden ser útiles para aplicaciones del mundo real.

2.3. Función *Triplet Loss*

Respecto de la función *Triplet Loss*, esta ha sido usada en diversas tareas de aprendizaje automático y visión computacional, comenzando con el trabajo de Schroff et al. [1], que empleó esta función en el contexto de reconocimiento facial, proponiendo un método llamado FaceNet, que obtuvo los mejores resultados hasta el momento de la publicación, al ser evaluado sobre las bases de datos LFW y YouTube Faces. Otros estudios en reconocimiento facial que utilizan *Triplet Loss* fueron desarrollados por Parkhi et al. [34], Trigueros et al. [35], Boutros et al. [36], Yeung et al. [37] y Feng et al. [38].

En los últimos años, la función *Triplet Loss* también ha sido usada exitosamente para resolver el problema de re-identificación de personas. Esta tarea está relacionada con la detección de peatones, pero presenta algunas diferencias importantes, y se define como la tarea de identificar y hacer coincidir a las mismas personas, ya sea a través de varias cámaras o a lo largo del tiempo dentro de una sola cámara [76]. En esta línea, se han propuesto muchos métodos para la re-identificación de personas usando *Triplet Loss*, alcanzando resultados comparables a los del estado del arte en los últimos años, como por ejemplo en los trabajos presentados en [39–44]. Un trabajo interesante fue propuesto por Wang et al. [77], el cual usa *Triplet Loss* para ajustar la distancia de las características entre cada peatón, y distinguir entre diferentes peatones en escenarios con aglomeraciones. Aquí hay una sustancial diferencia con el método desarrollado, ya que este usa la función *Triplet Loss* para agrupar las características de todos los peatones, en lugar de identificar individuos específicos.

Otra tarea donde la función *Triplet Loss* ha sido usada de forma satisfactoria es la resolución del problema de generalización de dominio en distintas tareas. La investigación de Lee [45] aplicó *Triplet Loss* para reconocer emociones en el habla entre diferentes *corpus* para generalizar entre dominios. Yu et al. [46] introdujeron una función *Triplet Loss* adaptada como enfoque novedoso para mitigar el sesgo en la selección de tripletes y abordar el cambio de distribución en los tripletes seleccionados, evaluando diferentes bases de datos de imágenes. Wang et al. [47] presentaron un marco novedoso de generalización de dominio, EISNet, que aprende a generalizar simultáneamente entre dominios diversos mediante la utilización de supervisión de relaciones extrínsecas y auto-supervisión intrínseca, particularmente para imágenes de múltiples dominios fuente. En el trabajo de Dou et al. [48], se usó un paradigma de aprendizaje agnóstico al modelo para exponer la optimización al cambio de dominio, introduciendo dos funciones de pérdida complementarias que regularizan explícitamente la estructura semántica del espacio de características. Deng et al. [49] estudiaron el aprendizaje métrico dentro de la adaptación de dominio, introduciendo una restricción guiada por similitud en forma de *Triplet Loss*, donde cada triplete se forma tomando datos tanto del dominio fuente como del dominio objetivo.

Finalmente, se pueden mencionar otras tareas de aprendizaje automático, no limitadas a la visión computacional, donde se han usado enfoques basados en *Triplet Loss*. Aquí se pueden citar algunas tareas como el seguimiento de objetos [78–80], reconocimiento de hablantes [81, 82], detección de intenciones para la comprensión del lenguaje hablado en sistemas de diálogo [83], recuperación de imágenes de teledetección [84], reconocimiento de gestos 3D [85], detección automática de *covers* musicales [86] y mejora de imágenes en entornos con baja iluminación [87].

Capítulo 3

Metodología

Los detectores de peatones de dos etapas más comunes en la literatura, tienen dos funciones de pérdida en la segunda etapa, es decir, en la cabeza de clasificación: una función de pérdida para la regresión de *bounding boxes*, y otra para la clasificación multiclase propiamente tal. Los métodos de detección de objetos de dos etapas suelen ser preferidos, porque han logrado mejor rendimiento, en términos de menores tasas de error, a costa de una menor velocidad de ejecución, en comparación con los enfoques de una sola etapa. Estos métodos demostraron ser exitosos cuando el conjunto de entrenamiento y prueba provienen del mismo dominio, es decir, pertenecen al mismo conjunto de datos. Sin embargo, cuando estos métodos se evalúan en conjuntos de datos nuevos, que no han sido vistos por el algoritmo, el rendimiento disminuye significativamente, como fue señalado por Hasan et al. en [19]. Para superar este problema, se desarrolló una nueva cabeza de clasificación, para ser usada en la segunda etapa de un detector de dos etapas, de manera que sea posible concentrar los ejemplos de peatones en el espacio de características explícitamente, independiente del conjunto al cual pertenecen. El método propuesto utiliza el concepto de tripletes, que ha sido utilizado con éxito en otras tareas de la visión computacional, especialmente en la de reconocimiento facial [1, 38]. Usamos la función de pérdida *Triplet Loss*, que involucra tres ejemplos para ser calculada: una muestra de anclaje, una muestra de la misma clase que la muestra de anclaje y una muestra de una clase diferente. De esta manera, la red aprende a minimizar la distancia entre muestras de la misma clase, mientras maximiza la distancia de muestras de diferentes clases.

3.1. Detectores de objetos de dos etapas

En este trabajo, los esfuerzos se enfocaron en mejorar la arquitectura de un detector de dos etapas. Se realizaron experimentos utilizando Faster R-CNN [14] y Cascade R-CNN [15], ambos pertenecientes a la familia de métodos R-CNN. Cascade R-CNN supera a Faster R-CNN en muchos *benchmarks* de detección de objetos, pero también se incluyó este último método porque todavía se utiliza, como se aprecia en la literatura, por lo que fue usado como línea base de comparación. Ambos métodos enfrentan la detección como un problema de aprendizaje multitarea, combinando clasificación y regresión de *bounding boxes*.

3.1.1. Fast R-CNN

Para entender estos detectores, primero se debe entender el método Fast R-CNN [13], el cual es un método de detección de objetos de dos etapas, que mejora la eficiencia y precisión en comparación con los métodos anteriormente desarrollados. Consta de una etapa de generación de propuestas de región, utilizando métodos como *Selective Search* o *EdgeBoxes*. Estas regiones son candidatas a contener objetos y se extraen de la imagen original. Luego, se extraen características de cada propuesta de región utilizando una red neuronal convolucional (CNN), para finalmente aplicar una cabeza de clasificación, la que determina a cada región como objeto o fondo, utilizando una capa *softmax* que se entrena con un conjunto de datos de entrenamiento. Esta capa produce una puntuación que indica la probabilidad de que la región contenga un objeto. También se hace la regresión de *bounding boxes*, de forma que se ajuste la región propuesta de la mejor forma al objeto en análisis. En la Figura 3.1 se puede apreciar el esquema de Fast R-CNN.

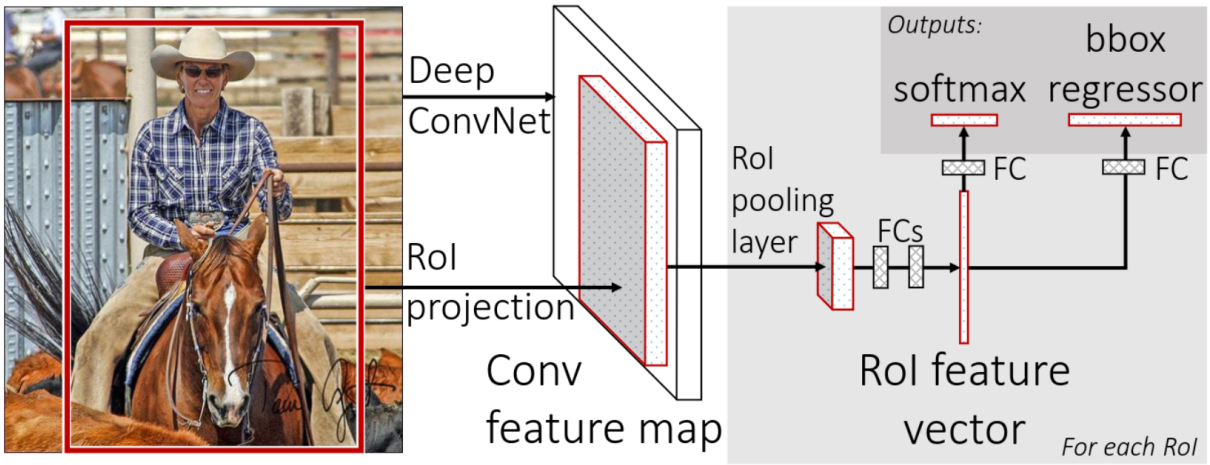


Figura 3.1: Detector Fast R-CNN. Tomado de [13].

En resumen, primero se toma una imagen la cual genera múltiples regiones de interés (ROIs), las cuales se introducen en una red *fully-convolutional*. Luego, cada región es agrupada (*pooled*) en un mapa de características de tamaño fijo, para luego ser mapeado a un vector de características por una red *fully-connected*. Estos vectores entran en una red con dos capas de salidas hermanas, donde la primera entrega una distribución de probabilidad discreta $p = (p_0, \ldots, p_K)$ por cada ROI, sobre las $K + 1$ categorías, y la otra, entrega los *offsets* de regresión $t^k = \left(t^k_{\mathrm{x}}, t^k_{\mathrm{y}}, t^k_{\mathrm{w}}, t^k_{\mathrm{h}}\right)$ para los *bounding boxes*, para cada una de las K clases. Cada ROI se etiqueta con un *ground truth* u y un *bounding box* anotado v. Para entrenar, se usó una función de pérdida multi tarea, dada por la siguiente ecuación:

$$L\left(p, u, t^u, v\right) = L_{\mathrm{cls}}(p, u) + \lambda[u \geq 1] L_{\mathrm{loc}}\left(t^u, v\right) \tag{3.1}$$

donde $L_{\mathrm{cls}}(p, u) = -\log p_u$ se define como la pérdida logarítmica (*log-loss*) para la clase verdadera u, y L_{loc} , se define sobre una tupla de objetivos de regresión del *bounding box* verdadero para las clases $u, v = (v_{\mathrm{x}}, v_{\mathrm{y}}, v_{\mathrm{w}}, v_{\mathrm{h}})$, y una tupla predicha $t^u = \left(t^u_{\mathrm{x}}, t^u_{\mathrm{y}}, t^u_{\mathrm{w}}, t^u_{\mathrm{h}}\right)$, nuevamente para la clase u. Por lo general, se utiliza la función *Smooth* L_1 para L_{loc} , debido a que es una función de pérdida robusta, menos sensible a outliers comparado a una pérdida L_2. La función $[u \geq 1]$ es 1 cuando $u \geq 1$, y es 0 en caso contrario.

3.1.2. Faster R-CNN

El método Faster R-CNN [14], propuesto por Ren et al, se basa en Fast R-CNN [13], pero tiene como principal diferencia la inclusión de una etapa de propuesta de regiones, la cual se realiza mediante una nueva Red de Propuesta de Regiones (RPN), que comparte características convolucionales con la red de detección, lo que permite propuestas de región casi sin costo. Esta RPN es una red totalmente convolucional, la cual se entrena en forma *end-to-end*, tomando como entrada una imagen de cualquier tamaño, y dando como salida un conjunto de propuestas de región rectangulares, donde cada una de estas propuestas se conforma de un *bounding box* y un score de objetividad, es decir, si hay un objeto en dicha región o no. Esta mejora se desarrolló para reducir el tiempo de ejecución de la etapa de propuesta de regiones de Fast R-CNN, que actuaba como cuello de botella en términos de velocidad.

Para generar las propuestas de región, se aplica una ventana deslizante a los mapas de características de la última capa convolucional, para luego mapear cada ventana a un vector de dimensión menor, 256 en este caso. Este vector se usa como entrada de dos redes *fully-connected*, una para hacer regresión de *bounding boxes*, y otra para clasificación. En cada

ubicación, se predicen en forma simultánea k propuestas de región, por lo que la capa de regresión entrega $4k$ salidas (Las 4 coordenadas de cada *bounding box*), y la red de clasificación entrega $2k$ salidas (La probabilidad de ser objeto/no objeto). Cada una de estas k propuestas están parametrizadas respecto a k cajas de referencia, llamadas anclas, las cuales están centradas para cada ubicación de la ventana deslizante. Se usaron 3 escalas y 3 razones de aspecto, lo que genera $k = 9$ anclas en cada posición. Este comportamiento se puede apreciar en la Figura 3.2.

Para poder entrenar la RPN, se asigna una etiqueta binaria a cada ancla, donde dicha etiqueta indica si hay un objeto o no. Dicha etiqueta positiva puede ser asignada a dos tipos de anclas: El o las anclas con el mayor porcentaje de intersección sobre unión (IoU) a una caja del *ground truth*, o un ancla que al menos tenga un IoU de 0,7 con cualquier caja del *ground truth*. Por otro lado, una etiqueta negativa se asigna a cualquier caja no positiva cuyo IoU sea menor a 0,3 para todas las cajas del *ground truth*.

La función de pérdida a minimizar, es del tipo multi tarea, tal como en Fast R-CNN, con dos funciones, L_{cls} que representa la función de clasificación, y L_{reg} representa a la función de regresión. Dicha función total se define, para una imagen, por la siguiente ecuación:

$$L\left(\{p_i\},\{t_i\}\right) = \frac{1}{N_{cls}}\sum_i L_{cls}\left(p_i, p_i^*\right) + \lambda\frac{1}{N_{\text{reg}}}\sum_i p_i^* L_{reg}\left(t_i, t_i^*\right) \tag{3.2}$$

donde i representa a un ancla dentro de un *mini-batch*, p_i es la probabilidad predicha de que i sea un objeto, p_i^* corresponde a si i está anotado como objeto, t_i es un vector de 4 coordenadas con el *bounding box* predicho, t_i^* representa el *bounding box* anotado, asociado a un ancla positiva, y λ representa el peso para balancear ambas funciones, las cuales están normalizadas por N_{cls} y N_{reg} respectivamente.

Por lo general, la función de clasificación L_{cls}, suele ser del tipo *cross-entropy*, mientras que la función de regresión L_{reg}, suele ser la función *Smooth* L_1. Viendo la ecuación 3.2, se puede apreciar que para la función de regresión, solo los ejemplos positivos son considerados, debido a la multiplicación $p_i^* L_{reg}$.

3.1.3. Cascade R-CNN

Cascade R-CNN [15] aborda el problema de las detecciones ruidosas, cuando un detector se entrena con un umbral IoU bajo, el que típicamente suele ser del orden de 0,5 en la mayoría de los casos. Este es un problema difícil de resolver, dado que el rendimiento tiende a empeorar a medida que se aumenta el umbral de IoU. Para superar este problema, se entrena

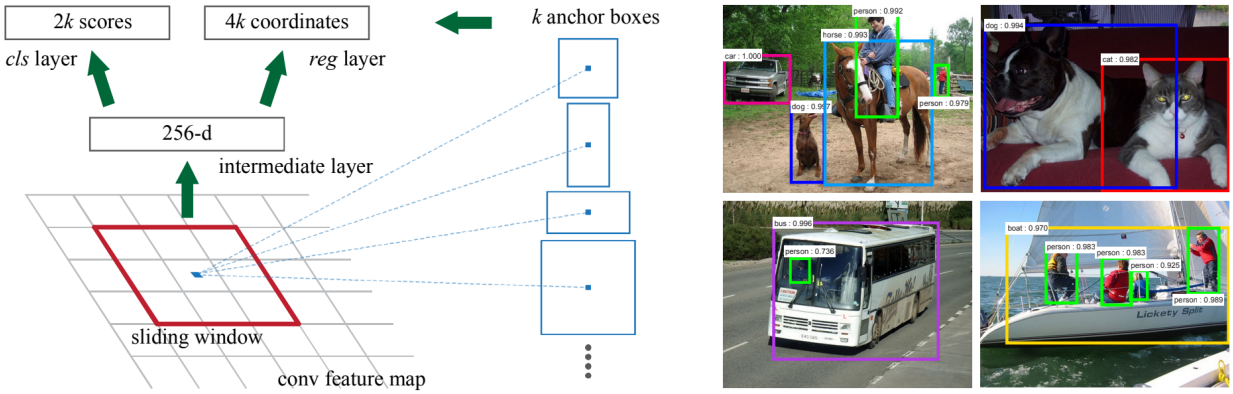


Figura 3.2: Red de propuesta de regiones usada en Faster R-CNN, con detecciones de ejemplo sobre el *benchmark* PASCAL VOC 2007. Tomado de [14].

una secuencia de cabezas de detección con umbrales de IoU crecientes, etapa por etapa, de manera que cada cabeza sea más selectiva secuencialmente contra falsos positivos cercanos. De esta manera, se filtran las muestras ancla falsas positivas, así generando propuestas de región de mejor calidad. En la Figura 3.3 se puede apreciar como la calidad de las detecciones depende del nivel de superposición entre las propuestas y las regiones de interés. Se muestran gráficos de desempeño de tres detectores entrenados con distintos umbrales de IoU, y se observa que cada regresor de *bounding boxes* funciona mejor para ejemplos con IoU cercano al umbral para el que fue entrenado. Además, se concluye que un detector optimizado en un solo nivel de IoU no necesariamente es óptimo en otros niveles, y que una detección de alta calidad requiere propuestas de alta calidad.

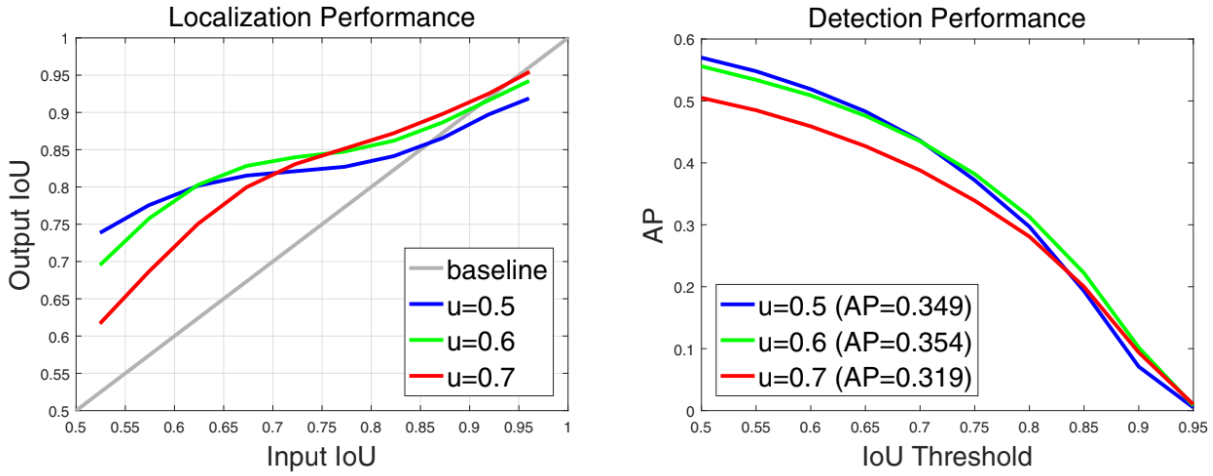


Figura 3.3: Desempeño de un detector de objetos a medida que aumenta el umbral IoU. Tomado de [15].

El diseño de Cascade R-CNN puede ser visto como un problema de regresión en casca-

da, con la arquitectura mostrada en la Figura 3.4 derecha, basándose en una secuencia de regresores especializados, definidos por la siguiente ecuación:

$$f(x, \mathbf{b}) = f_T \circ f_{T-1} \circ \cdots \circ f_1(x, \mathbf{b}) \tag{3.3}$$

donde T representa al número total de etapas de la cascada, y f_t representa a cada regresor en la cascada, el cual está optimizado con respecto a la distribución de muestras $\{\mathbf{b}^t\}$, en vez de la distribución inicial $\{\mathbf{b}^1\}$.

Para cada etapa t de la cascada, R-CNN incluye un clasificador h_t y un regresor f_t, el cual está optimizado para el umbral de IoU u^t, donde se debe cumplir la condición $u^t > u^{t-1}$. Para lograr este objetivo, se debe minimizar la siguiente función de costo:

$$L\left(x^t, g\right) = L_{cls}\left(h_t\left(x^t\right), y^t\right) + \lambda\left[y^t \geq 1\right] L_{loc}\left(f_t\left(x^t, \mathbf{b}^t\right), \mathbf{g}\right) \tag{3.4}$$

donde $\mathbf{b}^t$ se define como $\mathbf{b}^t = f_{t-1}\left(x^{t-1}, \mathbf{b}^{t-1}\right)$, g representa el objeto anotado $x^t, \lambda = 1$ representa el coeficiente de balance entre ambas funciones, [.] representa la función indicatriz, y finalmente y^t es la etiqueta para el ejemplo x^t dado un umbral u^t, es decir, se considera positiva si se solo se supera el umbral de IoU dado.

En tiempo de inferencia, el método en cascada descrito genera hipótesis cuya calidad se va mejorando en forma secuencial, lo que implica que detectores de mejor calidad son solo requeridos cuando las hipótesis que ingresan son de alta calidad. De esta manera, se pueden obtener una detección de objetos de alta calidad, en el sentido de los umbrales de IoU aplicados.

Una comparación de las arquitecturas genéricas de Faster R-CNN y Cascade R-CNN se muestra en la Figura 3.4. En Faster R-CNN, la primera etapa es una subred de propuestas de región, H0, que opera sobre la imagen completa, generando hipótesis de detección preliminares, conocidas como propuestas de objetos, y denotadas como B0. En la segunda etapa, las hipótesis son procesadas por una subred de detección de regiones de interés, H1, denominada cabeza de detección, asignando un puntaje de clasificación final, C1, y un *bounding box* refinado, B1. Esto es análogo para el caso de Cascade R-CNN.

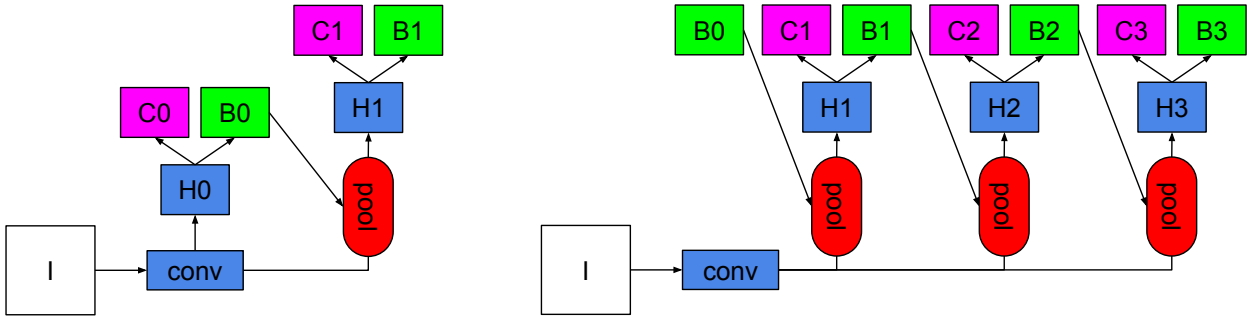


Figura 3.4: Comparación de arquitecturas genéricas para Faster R-CNN en la izquierda, y Cascade R-CNN en la derecha. I es la imagen de entrada, conv corresponde a las capas de una red neuronal convolucional, pool es el extractor de características para las regiones de interés, H son las cabezas, donde C es una cabeza de clasificación y B una de regresión de *bounding boxes*.

3.2. Función *Triplet Loss*

La función de pérdida *Triplet Loss*, se encarga de entrenar una red neuronal, de manera que características de la misma clase queden cerca, y al mismo tiempo maximiza la distancia entre características de diferentes clases. Para calcular esta función, se elige una muestra de anclaje, junto con una muestra negativa y una positiva. Como resultado, la función *Triplet Loss* maximiza la distancia inter-clase explícitamente, mientras minimiza la distancia intra-clase, utilizando un término de margen, el cual se usa para determinar la frontera de decisión entre pares positivos y negativos.

Esta función ha sido utilizada con éxito en muchas aplicaciones de aprendizaje automático y visión computacional, como por ejemplo, reconocimiento facial [1, 34, 36], re-identificación de personas [41, 42, 76], seguimiento de objetos [78–80], y reconocimiento de locutores [81, 82], entre otros.

Esta familia de funciones se aplica generalmente a la proyección de muestras (*Embeddings*), realizadas por una red neuronal. La figura 3.5 muestra el comportamiento de este tipo de función, donde inicialmente se tiene que la distancia entre el ancla y el negativo es menor a la distancia entre el ancla y el positivo, y luego, a través de la optimización de la función, se logra acercar el ejemplo positivo al ancla, mientras que el ejemplo negativo se aleja del ancla.

Formalmente, se define una función de pérdida *Triplet Loss* mediante tripletes de *embeddings*, definiendo los siguientes conceptos:

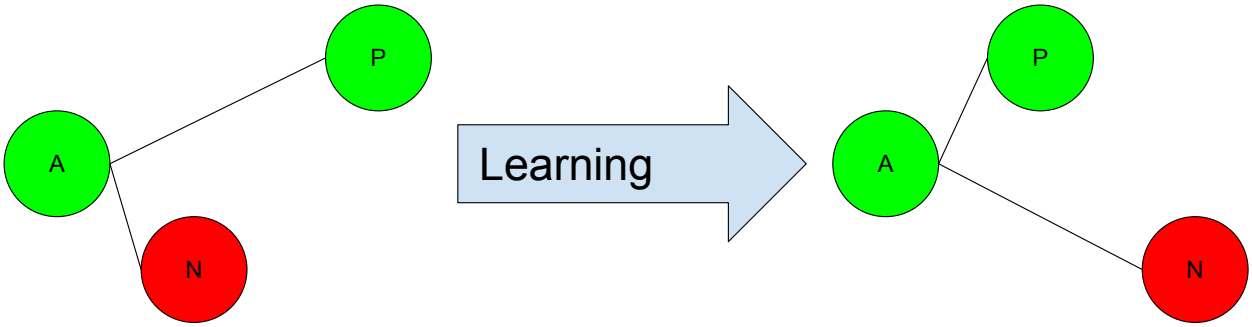


Figura 3.5: Función *Triplet Loss*. La red aprende a minimizar la distancia entre ejemplos de la misma clase (A y P en este caso), mientras maximiza la distancia entre ejemplos de clases distintas (A y N).

- Una muestra ancla a, por ejemplo, un peatón.
- Una muestra positiva p, con la misma clase que el ancla.
- Una muestra negativa n de una clase diferente, por ejemplo, el fondo.

Para alguna métrica de distancia d en el espacio embebido (típicamente la distancia euclidiana), el valor de la función para un triplete (a, p, n) se define como:

$$L_t = \max(d(a,p) - d(a,n) + margen, 0) \tag{3.5}$$

El objetivo de esta función es generar *embeddings* que permitan diferenciar claramente entre ejemplos de diferentes clases, mientras que los *embeddings* de los ejemplos de una misma clase, queden agrupados de manera efectiva. El margen actúa como una medida de cuánto los *embeddings* de distintas clases deben estar separados en el espacio. En consecuencia, se espera que los *embeddings* de ejemplos de la misma clase estén más cerca entre sí que los *embeddings* de diferentes clases, y que los *embeddings* de diferentes clases estén separadas por una distancia al menos igual al margen.

Basándose en la definición dada en la ecuación 3.5, se pueden obtener tres categorías de tripletes:

- Tripletes fáciles: tripletes que tienen un valor de la función de pérdida de 0, porque $d(a,p) + margin < d(a,n)$.
- Tripletes difíciles: tripletes donde el negativo está más cerca del ancla que el positivo, es decir, $d(a,n) < d(a,p)$.

- Tripletes semi-difíciles: tripletes donde el negativo no está más cerca del ancla que el positivo, pero aún tienen un valor de pérdida positiva: $d(a,p) < d(a,n) < d(a,p) + margin$.

Según la definición anterior, se pueden categorizar las muestras negativas en negativos difíciles, negativos semi-difíciles y negativos fáciles. Esto está relacionado con la ubicación de la muestra negativa en relación con el ancla y las muestras positivas dentro del espacio embebido. Esto se puede observar con mayor detalle en la Figura 3.6. En este gráfico, se puede tomar un ejemplo negativo que se sitúe en cualquier lugar del espacio bidimensional mostrado, lo que generará un ejemplo negativo fácil, semi-difícil o difícil, de acuerdo al lugar donde dicho ejemplo negativo se encuentre.

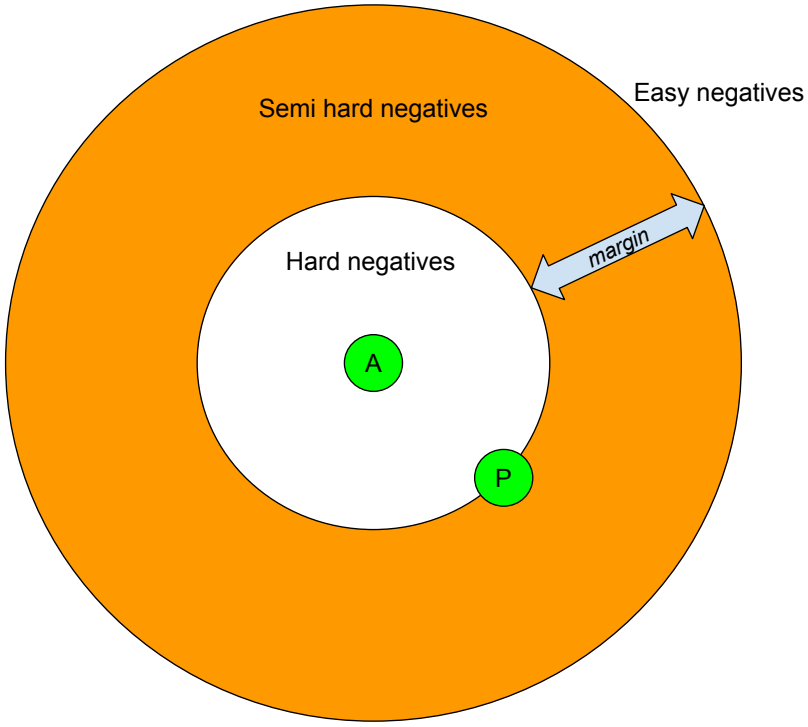


Figura 3.6: Categorización de ejemplos negativos, en función de las distancias relativas a los positivos y el ancla.

La estrategia de selección de tripletes es un paso crucial para lograr un buen rendimiento de detección, como se indica en la literatura [44, 78]. Es necesario seleccionar qué tripletes serán procesados por la red, ya que si se generaran todos los tripletes posibles, muchos de ellos serían fáciles y no contribuirían al entrenamiento. Esto resultaría en una convergencia más lenta, ya que todos los tripletes deben ser procesados a través de las capas de la red neuronal profunda usada. Por lo tanto, es crucial seleccionar tripletes activos, es decir, los que se según la definición anterior se clasifican como difíciles y semi-difíciles, de manera que puedan contribuir a mejorar el modelo durante el entrenamiento [1]. En este caso, se siguió la estrategia utilizada

en [1], utilizando el muestreo en línea de ejemplos negativos, que asegura que la dificultad de los tripletes aumente a medida que avanza el entrenamiento de la red. Con este fin, se generaron los tripletes en línea, seleccionando los ejemplos positivos/negativos difíciles dentro de un *mini-batch*. También es importante evitar el problema del entrenamiento inestable. Con este propósito, se necesita garantizar una representación significativa de las distancias entre el ancla y los positivos. Es necesario asegurar que un número mínimo de ejemplos de cada clase esté presente en cada *mini-batch*. Se adaptó el procedimiento utilizado en [1] al caso particular en desarrollo, asegurando que para cada imagen, durante el entrenamiento, haya al menos un peatón que genere ejemplos positivos, después de la etapa de generación de regiones de interés (ROI *pooling*), la cual se usa para crear los tripletes.

3.3. Nueva cabeza de clasificación

El método propuesto se enfoca en la cabeza de clasificación, que está inmediatamente después de la etapa de ROI *pooling*, con las características calculadas a partir de las regiones generadas por la Red de propuestas de región (RPN). Como se mencionó anteriormente, los detectores de objetos de dos etapas actuales están diseñados para aprender en forma multitarea, haciendo la clasificación y la regresión de *bounding boxes* de forma simultánea. Se modificó la cabeza final de clasificación actual, agregando una tercera función de pérdida, de manera que las distancias entre *embeddings* se optimicen de acuerdo con la función *Triplet Loss* definida anteriormente.

En este trabajo, las muestras de entrada a la cabeza de clasificación desarrollada son las regiones de interés (ROIs) proyectadas en el espacio embebido, generadas por el extractor de ROIs. Por lo tanto, dichas muestran son comparables, y se les puede aplicar una métrica, dado que los vectores son del mismo tamaño. El método propuesto, aplicado a un detector Faster R-CNN, se puede observar en la Figura 3.7. Se aplicó la función *Triplet Loss* utilizando las características descritas previamente, de manera que emule en cierta medida el comportamiento de un espacio embebido en, por ejemplo, una tarea de reconocimiento facial, donde las caras del mismo sujeto deben estar más cercanas, concentradas en el espacio, y más lejos de las caras de otros sujetos. En este caso, se requiere que las características de los peatones sean más cercanas entre sí, y que estén alejadas de las características que representan al fondo.

La función de pérdida de la nueva cabeza de clasificación, queda definida de la siguiente manera:

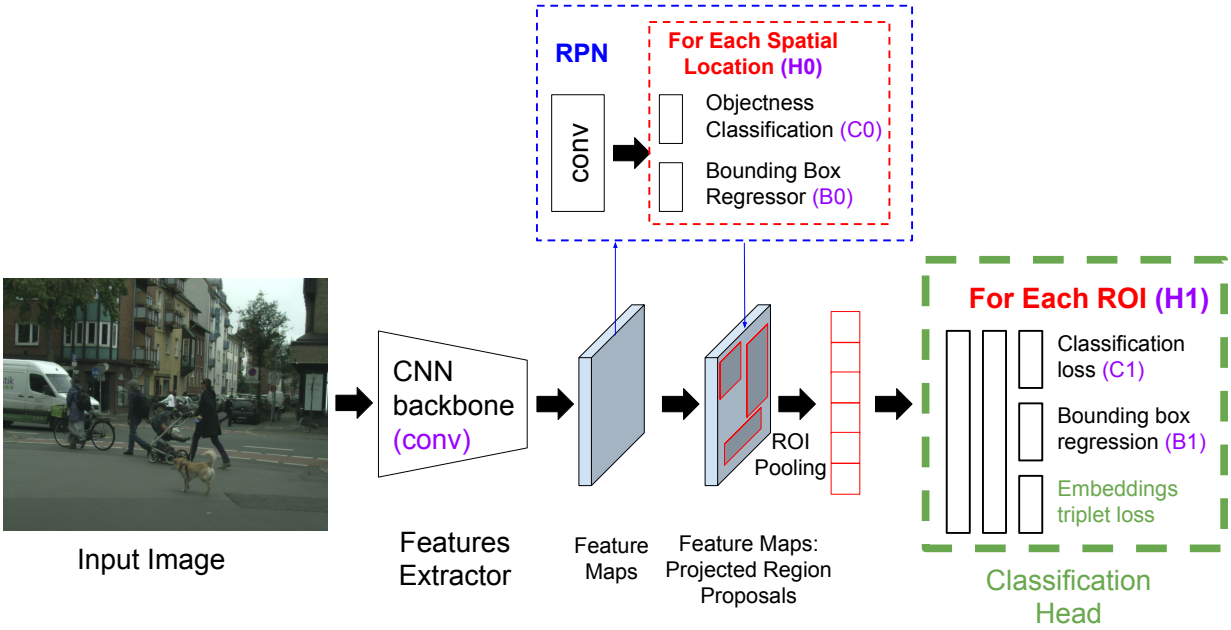


Figura 3.7: Detector Faster R-CNN, con la nueva cabeza de clasificación aplicada en la segunda etapa del sistema. Las contribuciones de este trabajo se muestran en verde, con la adición de la función *Triplet Loss* en la cabeza de clasificación. En morado se muestran los bloques correspondientes a la Figura 3.4.

$$L\left(p, u, t^{u}, v, f\right) = L_{\text{cls}}(p, u) + \lambda_1[u \geq 1]L_{\text{loc}}\left(t^{u}, v\right) + \lambda_2 L_{\text{tri}}\left(f, u\right) \tag{3.6}$$

En la ecuación anterior, L_{cls} representa la función de pérdida de clasificación, la cual típicamente se implementa usando *cross-entropy*, L_{loc} representa la función de pérdida de regresión, típicamente implementada usando *Smooth* L_1, y finalmente L_{tri} representa la función aplicada sobre los *embeddings*, en este caso, *Triplet Loss*, la cual actúa sobre las características extraídas f luego de la extracción de regiones de interés. Por otro lado, λ_1 representa el peso asignado a la función de regresión, y λ_2 representa al peso asignado a la función aplicada sobre los *embeddings*. Análogamente a la ecuación 3.1, p es la distribución de probabilidad discreta para cada ROI, sobre las $K + 1$ categorías, t^u son los *bounding box* predichos, u la etiqueta del *ground truth*, y v representa un *bounding box* anotado.

En el caso de un detector Faster R-CNN, la cabeza de clasificación fue modificada agregando la función de pérdida *Triplet Loss* directamente como una tercera función, de esta manera se complementa a las funciones de pérdida de clasificación y regresión de *bounding boxes*, presentes en la versión estándar de la cabeza de clasificación. Para el caso de un detector Cascade R-CNN, dado que este posee varias cabezas de clasificación, tal como se puede observar a la derecha en la Figura 3.4, entonces fue necesario decidir que cabeza de clasifi-

cación puede ser reemplazada. En este caso, podemos usar esta nueva cabeza, y reemplazar cualquiera de las tres cabezas existentes, H1, H2 o H3, pudiendo ser solo una, dos o las tres en forma simultánea. Para este objetivo, se realizaron varios experimentos para encontrar la ubicación óptima de esta nueva cabeza en el *pipeline* de detección Cascade R-CNN. En todos los experimentos realizados, el margen utilizado fue de 1,0, y la selección de tripletes para *Triplet Loss* se realizó dentro del *batch*, de manera *online*, de esta forma, se obtiene un entrenamiento mucho más eficiente.

3.4. Experimentos

Para los experimentos dentro del mismo dominio, se utilizó la base de datos CityPersons para entrenar Faster R-CNN y Cascade R-CNN con pérdida *Triplet Loss* de la siguiente manera: Se usó la partición de entrenamiento disponible en CityPersons para el entrenamiento. Para los experimentos de generalización de dominio, se utilizó la partición de validación de CityPersons para la evaluación, pero no fue empleada la partición de entrenamiento. Se usó la partición de validación en vez de la partición de prueba, debido a que esta última está destinada a ser usada en *challenges*, donde se deben enviar los resultados a un servidor para ser procesados, por lo que no posee anotaciones disponibles para ser usadas, a diferencia de la partición de validación, que sí tiene anotaciones disponibles. Este es el procedimiento estándar utilizado en la literatura [18, 19, 62, 88]. En cuanto a la aumentación de datos, solo se usó la operación de espejado vertical. Para un trabajo futuro, se pueden aplicar otras operaciones de aumentación de datos con el objetivo de mejorar las capacidades de generalización del método desarrollado. En la Figura 3.8 se muestra un resumen de las bases de datos usadas en los experimentos. En la Figura 3.9, se muestran los diagramas de bloque genéricos en las etapas de entrenamiento y validación (Inferencia). En la etapa de entrenamiento del detector, se puede usar cualquier detector de peatones/objetos disponible en la literatura.

La segunda parte de los experimentos consistió en el uso del protocolo de *pipeline* de entrenamiento progresivo descrito en [19], lo que permitió tener una ventaja al aumentar el desempeño en la detección de peatones sobre conjuntos de datos obtenidos de múltiples fuentes. Este *pipeline* entrena detectores utilizando un conjunto de datos diverso, pero diferente al del dominio objetivo, lo que permite mejorar en forma incremental el desempeño. Posteriormente, el *pipeline* realiza re-entrenamientos en conjuntos que se asemejan más al dominio objetivo. Se debe aclarar que se utilizó solo el subconjunto de entrenamiento de cada conjunto para el entrenamiento, y el subconjunto de evaluación para medir el desempeño del sistema desarrollado. En la Figura 3.10, se observa, en términos genéricos, el esquema de entrenamiento progresivo propuesto. En la parte superior de la Figura, se observa la primera

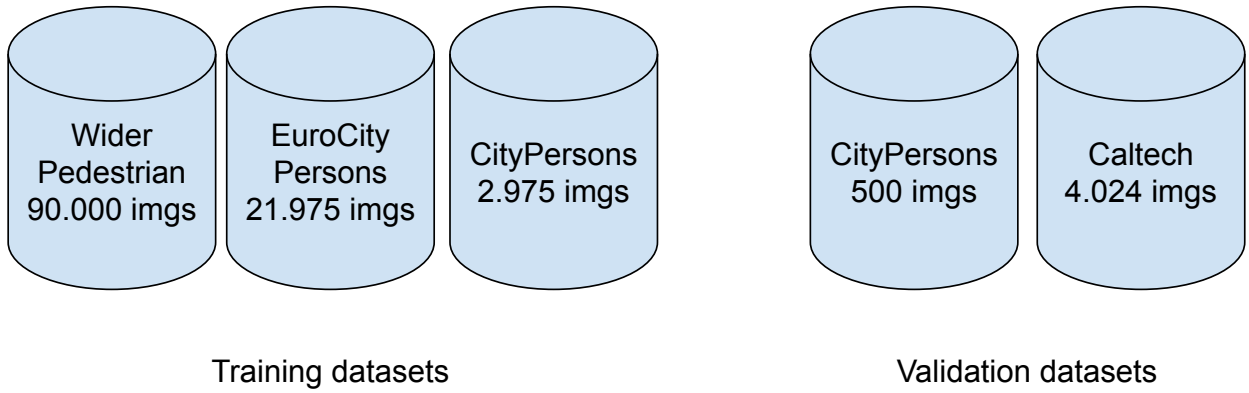


Figura 3.8: Bases de datos usadas en los experimentos.

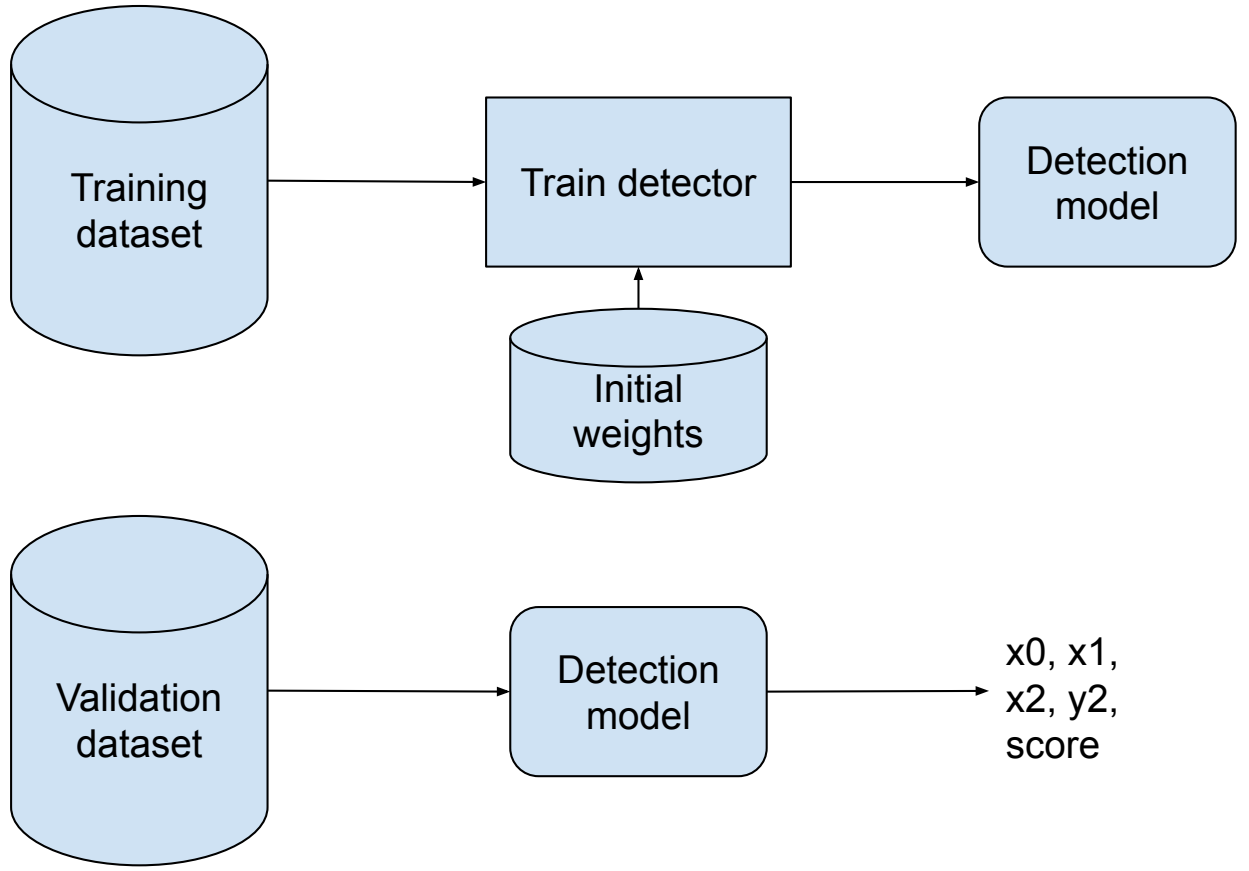


Figura 3.9: Diagramas de bloques en las etapas de entrenamiento (Parte superior) y validación (Parte inferior) del sistema.

parte del *pipeline*, donde se entrena un detector de objetos sobre una base de datos mas alejada del dominio objetivo, en este caso, Wider Pedestrian. Una vez entrenado, se puede observar en la parte inferior, el *fine tuning* del modelo previamente entrenado, sobre una base de datos mas cercana al dominio objetivo, en este caso, EuroCity Persons.

Se realizaron experimentos usando distintos modelos de detección, comparando su desem-

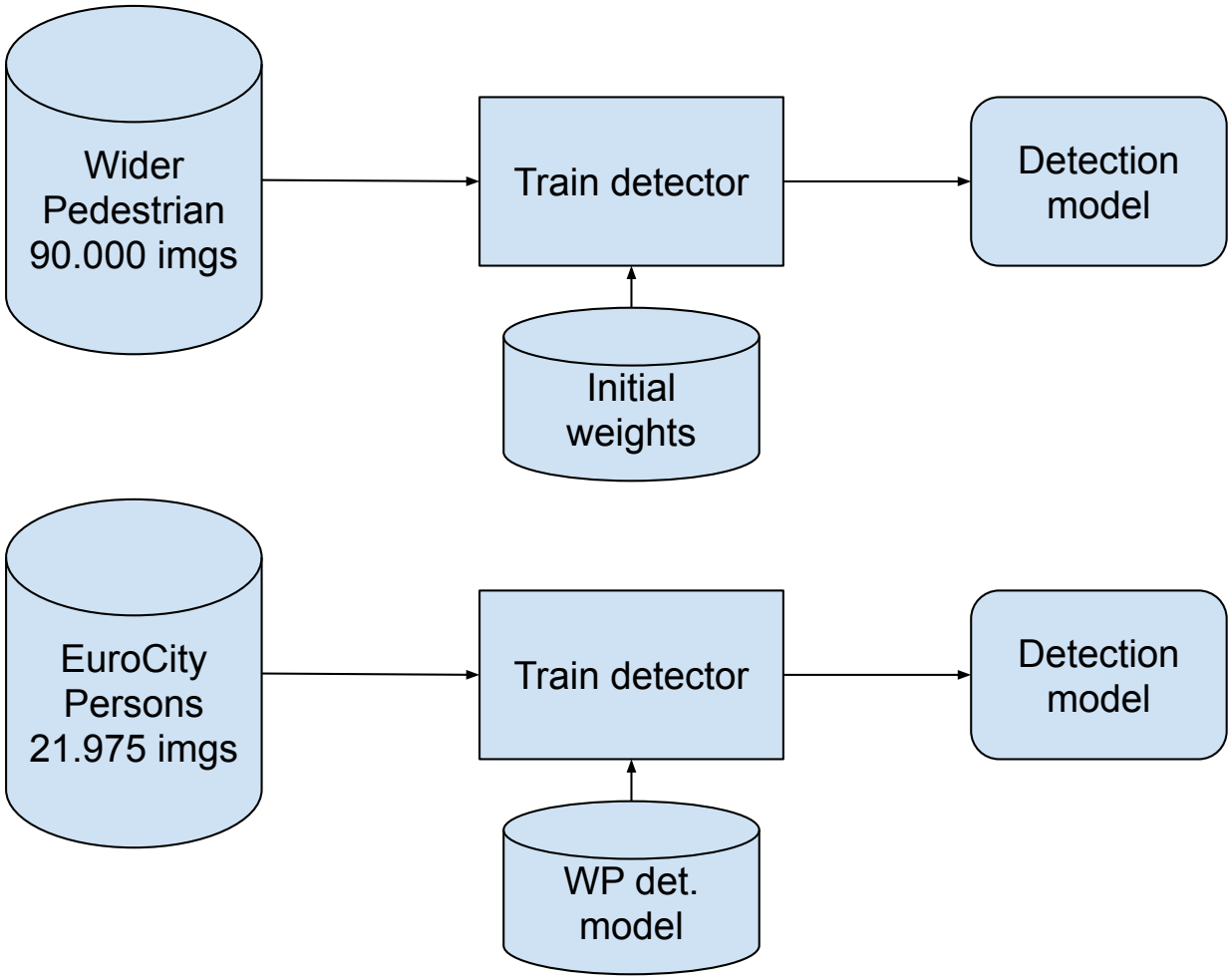


Figura 3.10: Diagramas de bloques del *pipeline* de entrenamiento progresivo. En la parte superior, se observa la primera parte, el entrenamiento sobre Wider Pedestrian. En la parte inferior, se observa el *fine tuning* del modelo entrenado sobre Wider Pedestrian, usando EuroCity Persons.

peño en *benchmarks* de detección de peatones. Los detectores de objetos base usados, fueron Faster R-CNN [14] y Cascade R-CNN [15]. El *backbone* del sistema, es decir, la red neuronal convolucional que calcula los mapas de características, fue HRNet [89], debido a que en el estado del arte, esta red ha mostrado desempeño superior a otras redes usadas como *backbone*, como por ejemplo, ResNet50 o ResNeXt [19].

La métrica de desempeño usada para medir los detectores desarrollados, fue la tasa de error logarítmica media sobre los Falsos Positivos por Imagen (FPPI), calculada como el promedio de la tasa de error para nueve tasas de FPPI, distribuidas uniformemente en el espacio logarítmico, usando el intervalo $[10^{-2}, 10^{0}]$ [20]. Cabe señalar que esta es la métrica que se usa actualmente en los detectores del estado del arte, tal como se muestra en numerosos trabajos [17–19, 62, 88, 90]. En lo sucesivo, esta métrica será referida como MR^{-2}. Esta métrica es, en cierto sentido, similar a la métrica *Average Precision* (AP), usada ampliamente para

cuantificar el desempeño en muchas aplicaciones de visión computacional. Una de las ventajas es que esta métrica provee de un solo valor, el cual representa toda la curva, de manera que se proporciona una evaluación más estable e informativa. Como la métrica MR^{-2} mide el error, un valor menor representa un mejor rendimiento para el algoritmo evaluado. Tal como se observa en el estado del arte, los métodos fueron evaluados para diferentes tamaños y niveles de oclusión de peatones, tal como se aprecia en la Tabla 3.1.

Tabla 3.1: Distintos niveles de tamaño y oclusión, usados para la evaluación de los métodos desarrollados.

Subconjunto	**Altura**	**Visibilidad**
Reasonable	[50, ∞]	[0,65, ∞]
Reasonable small	[50, 75]	[0,65, ∞]
Heavy	[50, ∞]	[0,2, 0,65]
All	[20, ∞]	[0,2, ∞]

3.5. Bases de datos

Para poder entrenar y evaluar los métodos desarrollados, es necesario contar con bases de datos internacionales, de manera que dichos resultados puedan ser comparados con los del estado del arte. En este trabajo, las bases de datos utilizadas, tanto para entrenamiento como evaluación, fueron las siguientes: Caltech [20], CityPersons [68], EuroCity Persons [74] y Wider Pedestrian [72].

3.5.1. CityPersons

Esta base de datos, creada por Zhang et al. [68], es un subconjunto de la base de datos Cityscapes [91], pero solo contiene anotaciones de personas. Las imágenes fueron capturadas en diferentes ciudades de Alemania y países cercanos. La base de datos posee 2.975 imágenes para entrenamiento, 500 para validación y 1.575 para prueba. En promedio, la base de datos tiene 6,47 peatones por imagen. La resolución de las imágenes es de 2.048 x 1.024 píxeles (ancho x alto). Se entregan anotaciones para la región visible de la persona y el cuerpo completo. Para poder comparar los resultados generados en este trabajo con los del estado del arte, solo se usaron los subconjuntos de entrenamiento y validación. En la Figura 3.11 se muestran algunos ejemplos de esta base de datos, donde se puede apreciar el distinto nivel de dificultad presente en las imágenes, dadas las distintas condiciones de iluminación, escala de los peatones, etc.

3.5.2. EuroCity Persons

Esta base de datos, creada por Braun et al. [74], es de gran escala, y fue capturada en 31 ciudades europeas, presentando una gran variedad de escenarios diferentes. Dependiendo del momento de la grabación, la base de datos se puede dividir en dos subconjuntos: diurno y nocturno. Hay 21.975 imágenes para entrenamiento, con un promedio de 9,2 peatones por imagen. La resolución de las imágenes es de 1.920 x 1.024 píxeles (ancho x alto). Para poder comparar los resultados obtenidos con los del estado del arte, en este trabajo solo se usó el subconjunto de entrenamiento diurno. En la Figura 3.12 se pueden observar algunos ejemplos de esta base de datos, donde se puede apreciar el distinto nivel de dificultad presente en las imágenes, dadas las distintas condiciones de iluminación, escala de los peatones, etc.

3.5.3. Wider Pedestrian

Esta base de datos, desarrollada por Loy et al. [72], fue construida para abordar el problema de detección de peatones en entornos no controlados. Las imágenes fueron capturadas en aplicaciones de conducción autónoma y de video vigilancia. La base de datos contiene 90.000 imágenes para entrenamiento, con un promedio de 3,2 peatones por imagen. Esta base de datos contiene imágenes de peatones recolectadas de diversas fuentes en internet, por lo que la resolución varía dependiendo del origen de los datos. En este caso, también se utilizó solo el subconjunto de entrenamiento para poder compararlo con otros resultados del estado del arte. En la Figura 3.13 se pueden observar algunos ejemplos de esta base de datos.

3.5.4. Caltech

Esta base de datos, desarrollada por Dollar et al. [20], es una de las más populares en la literatura de detección de peatones. Consiste en 10 horas de video grabados en Los Ángeles, EE.UU., mediante una cámara frontal de un vehículo, y contiene aproximadamente 43.000 imágenes y 13.000 personas extraídas del video. El conjunto de validación posee 4.024 frames. En promedio, esta base de datos posee 0,32 peatones por imagen. La resolución de las imágenes es de 640 x 480 píxeles (ancho x alto). Para evaluar, se usaron las anotaciones refinadas, las cuales se presentan en el trabajo de Zhang et al. [28].

Analizando las bases de datos usadas, podemos notar que Caltech es la que menos dificultad presenta, debido a que las escenas son obtenidas en la misma ciudad, en la misma estación, y solo de día, con un número bajo de peatones por imagen. Por otro lado, la base de datos CityPersons, presenta mayor dificultad, ya que está grabada en varias ciudades de

Alemania y países vecinos, en tres estaciones, lo que implica distintos niveles de iluminación y distintos estilos de vestimenta. Al igual que Caltech, CityPersons solo presenta imágenes de día, aunque con un número mayor de peatones por imagen. En tercer lugar, la base de datos EuroCity Persons, posee aún mas variabilidad, dado que está capturada en 31 ciudades de 12 países europeos, durante las cuatro estaciones del año, lo que implica aún mas variedad tanto en la iluminación, estilos de ropa, y condiciones climáticas, secas, húmedas, nieve, etc. EuroCity Persons posee un número aún mayor de peatones por imagen, comparado a Caltech y CityPersons. Finalmente, WiderPedestrian está compuesta por imágenes capturadas en escenarios de vigilancia y de conducción de automóviles, por lo que posee ángulos de cámara, escala de objetos e iluminación muy diferentes, donde incluso existen algunas imágenes capturadas de noche. Dado que esta base de datos fue confeccionada mediante recolección de imágenes en internet, y con escenarios de vigilancia y conducción, es la que mas variabilidad posee. Es importante notar que que las personas en las bases de datos Caltech, CityPersons y EuroCity Persons, dada la naturaleza de las capturas, están distribuidas en una banda estrecha a lo largo del centro de la imagen. Específicamente, las personas se concentran en ambos lados de la carretera y aparecen principalmente en el lado derecho, debido al método de recolección de datos sesgado donde las imágenes son capturadas desde un automóvil que circula bajo condiciones de tráfico diestro. En contraste, WiderPedestrian posee una distribución uniforme de ubicaciones, con personas apareciendo en varias posiciones excepto en la parte superior de la imagen.

Este trabajo se centra en obtener los mejores resultados de detección de peatones, usando la partición de validación de la base de datos CityPersons, y adicionalmente, se realizaron evaluaciones en la base de datos Caltech, de manera de poder comparar con los resultados publicados en el estado del arte.

Respecto a la implementación, todos los experimentos se llevaron a cabo utilizando el *framework* de detección de objetos de código abierto MMDetection [92], que se basa en el *framework* de *Deep Learning* PyTorch, como parte del proyecto OpenMMLab. Las GPUs utilizadas para ejecutar los experimentos fueron una NVIDIA GeForce RTX 2080Ti con 11 GB de RAM, y una NVIDIA GeForce GTX 1080Ti, también con 11 GB de RAM. De manera similar a otros trabajos del estado del arte, se usó solo espejado vertical para la aumentación de datos [19]. El optimizador usado fue SGD (Del *inglés Stochastic Gradient Descent*), con distintas tasas de aprendizaje, *momentum* de 0,9 y *weight decay* de 0,0001.

Figura 3.11: Ejemplos de la base de datos CityPersons, tomadas en distintas ciudades de Alemania y países vecinos.

Figura 3.12: Ejemplos de la base de datos EuroCity Persons, capturadas en distintas ciudades europeas.

Figura 3.13: Ejemplos de la base de datos Wider Pedestrian.

Figura 3.14: Ejemplos de la base de datos Caltech, tomadas en Los Angeles.

Capítulo 4

Resultados, análisis y discusión

En esta sección, se explican los experimentos realizados, aplicando la cabeza de clasificación desarrollada con *Triplet Loss* a los detectores de peatones basados en Faster R-CNN y Cascade R-CNN. Luego, se llevó a cabo un estudio de ablación para evaluar el impacto de aplicar la cabeza desarrollada a los detectores descritos anteriormente, y poder comparar los resultados contra las versiones regulares de estos. Finalmente, se ejecutó un estudio de comparación con el estado del arte, para luego mostrar algunos resultados cualitativos, cotejando el método propuesto contra uno del estado del arte. Cabe señalar que todos los experimentos realizados consistieron en entrenar los detectores usando varias épocas de entrenamiento, y los resultados presentados en las tablas corresponden a los obtenidos en la mejor época para cada entrenamiento realizado.

4.1. Resultados para Faster R-CNN con *Triplet Loss*

En esta serie de experimentos, se utilizó la nueva cabeza propuesta, la cual incorpora la función *Triplet Loss*, reemplazando la cabeza estándar de clasificación para un detector de objetos Faster R-CNN. Para entrenar el detector propuesto, se utilizó un *backbone* HRNet pre-entrenado sobre la base de datos ImageNet. Los resultados obtenidos se pueden observar en la Tabla 4.1.

De la Tabla 4.1 se puede observar que una pequeña contribución de *Triplet Loss* a la cabeza de clasificación, reflejado en un valor bajo del peso, logra obtener los mejores resultados. Por otro lado, al incrementar el valor del peso de *Triplet Loss*, se observa un aumento significativo del error.

Tabla 4.1: Resultados MR^{-2} para cada partición de la base de datos CityPersons, usando un detector Faster R-CNN con *Triplet Loss*. El detector fue entrenado usando CityPersons. En negrita se destacan los mejores resultados.

Peso Triplet Loss	Part. Reasonable	Part. Small	Part. Heavy	All
0,025	**13,7**	18,3	40,2	**37,6**
0,050	14,8	**17,6**	**39,7**	38,1
0,010	14,0	18,6	40,9	37,7
0,150	14,2	18,2	40,4	37,8
0,200	14,6	18,7	41,5	39,1
0,250	15,7	19,1	42,1	39,5
0,500	16,5	20,2	47,2	41,5
0,750	17,6	20,3	48,8	43,0
1,000	17,9	21,3	48,5	42,7

Una vez finalizados los experimentos en el mismo dominio, se realizaron experimentos de generalización de dominio. Para ello, se entrenó un detector Faster R-CNN usando la base de datos WiderPedestrian, para luego evaluar dicho detector sobre la base de datos CityPersons, tal como se puede observar en la Tabla 4.2. Al comparar los resultados con los de la Tabla 4.1, se puede observar que usar pesos en el intervalo $[0, 0, 25]$ resulta en el menor error. Dado esto, solo se realizaron experimentos de generalización de dominio enfocándonos en valores bajos del peso de *Triplet Loss*, en el rango $[0, 0, 25]$.

Tabla 4.2: Resultados MR^{-2} de generalización, usando un detector Faster R-CNN con *Triplet Loss*, entrenando en WiderPedestrian y evaluando en cada partición de CityPersons. En negrita se destacan los mejores resultados.

Peso Triplet Loss	Part. Reasonable	Part. Small	Part. Heavy	All
0,025	16,1	20,7	55,3	44,8
0,050	15,2	21,2	53,5	43,2
0,100	15,7	19,6	54,8	44,3
0,150	**15,1**	**18,7**	**52,9**	42,9
0,200	15,6	20,4	53,5	**42,7**
0,250	16,1	20,0	53,2	43,2

En la Tabla 4.2, se puede apreciar que los resultados obtenidos son peores en comparación al entrenar y evaluar en el mismo dominio, cuyos resultados se muestran en la Tabla 4.1. Esto se explica porque la base de datos fuente en la Tabla 4.2 es diferente a la base de datos objetivo. Este efecto se amplifica en el subconjunto *Heavy*, el más difícil en el conjunto de evaluación, donde el error aumentó en cerca de un 13 %, comparado con el subconjunto *Reasonable*, donde el aumento del error fue mucho menor, cercano al 2 %. Al observar los

resultados en la Tabla 4.2, el desempeño usando *Triplet Loss* es el mejor en términos generales para todos los subconjuntos cuando el peso es 0,15, pero aún se mantienen alejados a los resultados que se presentan en la Tabla 4.1. No obstante, en los experimentos siguientes, los resultados mejoraron significativamente cuando el modelo se ajustó finamente utilizando *Triplet Loss* en bases de datos más cercanas al dominio objetivo, en este caso, EuroCity Persons.

4.2. Resultados para Cascade R-CNN con *Triplet Loss*

El siguiente paso fue realizar una serie de experimentos basándose en un detector Cascade R-CNN. Este método posee múltiples cabezas de clasificación y, por lo tanto, se tuvo que elegir el lugar donde se debió aplicar la función *Triplet Loss*. Con este propósito, se ejecutaron varios experimentos cambiando la cabeza estándar del detector por la cabeza desarrollada, ajustando el peso que se le da a la función *Triplet Loss* dentro de la cabeza seleccionada. En la Tabla 4.3 se muestra un resumen con los resultados de los experimentos realizados, donde, en términos generales, aplicar *Triplet Loss* en la primera cabeza de clasificación parece generar los mejores resultados, ya que muestra buen desempeño en los subconjuntos *Reasonable*, *Small* y *Heavy* al mismo tiempo. El desempeño de detección aumentó en comparación con los resultados obtenidos utilizando Faster R-CNN, como se muestra en la Tabla 4.1. En este caso, el mejor resultado para el subconjunto *Reasonable* se obtuvo modificando la primera cabeza de clasificación, asignando un peso bastante bajo a la función *Triplet Loss*, con un valor de 0,05.

Procediendo de manera similar a Faster R-CNN, se realizaron experimentos entre bases de datos para Cascade R-CNN, En este caso, se entrenaron los modelos desarrollados usando la base de datos WiderPedestrian, para luego ser evaluados en CityPersons. Dado que los resultados muestran ser más consistentes para todas las particiones de CityPersons aplicando la nueva cabeza de clasificación de la primera posición, se eligió esta cabeza para los siguientes experimentos. También se debe señalar que valores pequeños del peso para *Triplet Loss* fueron usados, porque, como se mencionó anteriormente, parece que un peso pequeño tiene un mayor impacto para mejorar los resultados en el detector. Los resultados para esta serie de experimentos se pueden observar en la Tabla 4.4.

De la Tabla 4.4, se puede apreciar que ocurrió el mismo efecto al usar Faster R-CNN, donde los resultados de generalización de dominio, son peores que los resultados en el mismo dominio, mostrados en la Tabla 4.3, Esto también se debe a que la base de datos fuente en la Tabla 4.4 es diferente a la base de datos objetivo. Nuevamente, este efecto es más notorio

Tabla 4.3: Resultados para Cascade R-CNN agregando *Triplet Loss* a cada cabeza de clasificación (H1, H2, y H3 en la Figura 3.4), entrenado y evaluado en CityPersons. En negrita se destacan los mejores resultados.

Cabeza	**Peso Triplet Loss**	**Reasonable**	**Small**	**Heavy**	**All**
H1	0,025	13,4	17,3	38,9	36,5
H1	0,050	**12,7**	16,1	39,7	35,9
H1	0,100	12,8	15,5	38,8	35,8
H1	0,150	13,4	16,8	41,7	36,9
H1	0,200	13,4	16,0	40,1	36,7
H1	0,250	13,8	17,1	40,5	36,4
H1	0,500	13,9	15,5	41,8	37,1
H1	0,750	14,8	17,5	44,4	38,6
H1	1,000	14,9	19,0	44,4	38,9
H2	0,025	13,7	16,4	40,1	37,2
H2	0,050	13,7	16,1	40,7	36,4
H2	0,100	13,5	16,8	38,5	36,0
H2	0,150	13,0	16,0	39,8	35,9
H2	0,200	15,9	20,1	48,1	40,4
H2	0,250	15,3	19,6	46,5	39,3
H2	0,500	13,0	16,1	40,2	36,1
H2	0,750	16,7	22,1	50,1	41,2
H2	1,000	18,6	24,7	54,6	43,9
H3	0,025	13,5	16,6	41,5	36,7
H3	0,050	**12,7**	15,2	40,5	35,7
H3	0,100	13,3	**14,7**	40,5	36,4
H3	0,150	13,6	15,5	**38,4**	**35,7**
H3	0,200	13,5	17,4	39,6	36,5
H3	0,250	13,1	16,1	40,0	**35,7**
H3	0,500	13,9	16,7	38,8	36,6
H3	0,750	13,0	14,8	40,3	36,0
H3	1,000	13,2	17,2	41,3	36,3

en la partición *Heavy*. El mejor resultado global para Cascade R-CNN se obtuvo aplicando *Triplet Loss* con un peso de 0,15 en la primera cabeza de clasificación, H1. Este peso funcionó bien en todas las particiones de CityPersons.

Tabla 4.4: Resultados MR^{-2} de generalización, usando un detector Cascade R-CNN con *Triplet Loss* en la primera cabeza de clasificación (H1 en la Figura 3.4), entrenando en WiderPedestrian y evaluando en cada partición de CityPersons. En negrita se destacan los mejores resultados.

Peso Triplet Loss	**Reasonable**	**Small**	**Heavy**	**All**
0,025	**13,5**	**18,1**	53,0	41,3
0,050	14,4	19,1	52,9	40,9
0,100	14,1	20,3	52,5	40,7
0,150	13,8	**18,1**	50,9	**39,7**
0,200	14,0	18,7	**50,7**	40,7
0,250	14,8	19,5	52,8	41,8

4.3. Estudio de ablación

En esta sección, se procede a comparar los resultados para los detectores de peatones basados en Faster R-CNN y Cascade R-CNN, tanto con la nueva cabeza desarrollada como sin ella.

En la Tabla 4.5 se observan resultados para Faster R-CNN y Cascade R-CNN, entrenados y evaluados sobre la base de datos CityPersons, usando la nueva cabeza propuesta y comparando al caso sin usarla.

Tabla 4.5: Valores de MR^{-2} para Faster R-CNN y Cascade R-CNN, comparando el mejor resultado obtenido con *Triplet Loss* contra el método regular, entrenado y evaluado sobre CityPersons. En negrita se destacan los mejores resultados.

Método	**Reasonable**	**Small**	**Heavy**	**All**
Faster R-CNN (sin *Triplet Loss*)	13,8	**16,2**	47,6	39,7
Faster R-CNN (*Triplet Loss* $\lambda = 0,025$)	**13,7**	18,3	**40,2**	**37,6**
Cascade R-CNN (sin *Triplet Loss*)	13,4	16,7	40,5	36,6
Cascade R-CNN (*Triplet Loss* H1 $\lambda = 0,050$)	**12,7**	**16,1**	**39,7**	**35,9**

De la Tabla 4.5, se puede observar que, tanto para Faster R-CNN como para Cascade R-CNN, una pequeña contribución de la función *Triplet Loss* a la cabeza de clasificación mejora el rendimiento en casi todos los subconjuntos en CityPersons. Para el método Faster R-CNN, los resultados en el subconjunto *Reasonable* mejoraron moderadamente, pero la mejora fue significativa en el subconjunto *Heavy*. En el caso de Cascade R-CNN, los resultados en todos los subconjuntos de CityPersons mostraron una mejora, aunque no fue tan significativa como en el caso de Faster R-CNN.

En la Tabla 4.6 se muestran los resultados para Faster R-CNN y Cascade R-CNN, entrenados sobre la base de datos WiderPedestrian y evaluados sobre CityPersons, usando la nueva cabeza modificada versus la versión estándar de cada detector.

Tabla 4.6: Valores de MR^{-2} para Faster R-CNN y Cascade R-CNN, comparando el mejor resultado obtenido con *Triplet Loss* contra el método regular, entrenado en WiderPedestrian y evaluado en CityPersons. En negrita se destacan los mejores resultados.

Método	**Reasonable**	**Small**	**Heavy**	**All**
Faster R-CNN (sin *Triplet Loss*)	15,9	20,9	54,9	44,6
Faster R-CNN (*Triplet Loss* $\lambda = 0, 150$)	**15,1**	**18,7**	**52,9**	**42,9**
Cascade R-CNN (sin *Triplet Loss*)	16,0	21,6	57,4	-
Cascade R-CNN (*Triplet Loss* H1 $\lambda = 0, 150$)	**13,8**	**18,1**	**50,9**	**39,7**

Se puede observar en la Tabla 4.6 que el uso de *Triplet Loss* nuevamente mejoró los resultados en comparación a los obtenidos sin ella. Este comportamiento es evidente tanto para Faster R-CNN como para Cascade R-CNN. La mejora en el rendimiento es especialmente notable para Cascade R-CNN, donde el error en el subconjunto *Reasonable* disminuyó en un 2,2 %, en un 3,5 % para el subconjunto *Small* y en un impresionante 6,5 % para el subconjunto *Heavy*. Esto demuestra el impacto de la cabeza desarrollada con *Triplet Loss* en un escenario entre bases de datos, para los ejemplos de mayor dificultad en la base de datos CityPersons, en este caso, el subconjunto *Heavy*.

4.4. Estudio de comparación

En la Tabla 4.7, se muestran resultados obtenidos por distintos métodos del estado del arte. En todos estos resultados se usó el mismo conjunto, en sus distintas particiones, para entrenar y evaluar, por lo que no se evaluó la capacidad de generalización de dominio de los métodos presentados. Cabe señalar que los algoritmos presentados en esta Tabla, a veces no muestran resultados para todas las particiones de CityPersons, esto se debe a que así aparecen en los artículos donde se presentan dichos métodos.

Se observa que al comparar los mejores resultados de Faster R-CNN en la Tabla 4.7 (15,4 % en el subconjunto *Reasonable*) con el mejor resultado en la Tabla 4.1 (13,7 % en el subconjunto *Reasonable*), se aprecia una disminución significativa del error MR^{-2}, en torno al 1,7 % para el subconjunto *Reasonable*. Además, los resultados en el subconjunto *Small* mejoraron en un 7,3 %, pasando de 25,6 % al utilizar Faster R-CNN con un *backbone* VGG16 y sin *Triplet Loss*, a un 18,3 % utilizando *Triplet Loss* y un backbome HRNet. Esto es un

Tabla 4.7: Valores de MR^{-2} para diferentes métodos del estado del arte, entrenados y evaluados en la base de datos CityPersons. En negrita se destacan los mejores resultados.

Método	**Reasonable**	**Small**	**Heavy**
Repulsion Loss [18]	13,2	-	-
ALFNet [17]	12,0	19,0	48,1
CSP (ResNet50) [62]	11,0	16,0	39,4
CSP (HRNet) [19]	9,4	**11,4**	**36,7**
Faster R-CNN (VGG16) [68]	15,4	25,6	-
PRNet (ResNet50) [88]	10,8	-	-
BGCNet (HRNet) [90]	**8,8**	-	-
Faster R-CNN (ResNeXt101) [19]	16,4	-	-
Cascade R-CNN (HRNet) [19]	11,2	14,0	37,1
DA-Net [63]	-	-	51,6
PedJointNet [64]	13,5	-	52,2
PP-Net [65]	12,1	-	53,0

efecto combinado de la adición de *Triplet Loss* y el *backbone* HRNet, ya que en [68], se utilizó un *backbone* VGG16 [93], el cual es más antiguo.

En la Tabla 4.8, se presenta una comparación que incluye los mejores resultados del método desarrollado, utilizando los detectores Faster R-CNN y Cascade R-CNN, en comparación con los del estado del arte para generalización de dominio utilizando WiderPedestrian como conjunto de entrenamiento, y CityPersons como conjunto de prueba, tomados de Hasan et al. [19]. Se puede observar en la tercera fila de la Tabla 4.8 que el método Faster R-CNN con *Triplet Loss* (w=0,15) supera al algoritmo Cascade R-CNN regular, el cual se observa en la primera fila, siendo este último un método moderno que, en la mayoría de las tareas de detección de objetos, supera a Faster R-CNN [15, 19]. La cuarta fila de la Tabla 4.8 muestra los resultados del método desarrollado aplicado a un detector Cascade R-CNN (H1 w=0,15). Se puede observar que los resultados del método desarrollado superan a los del Cascade R-CNN regular mostrado en la primera fila, obteniendo una mejora del 2,2 % para el subconjunto *Reasonable* y del 6,5 % en el subconjunto *Heavy*.

Finalmente, se llevaron a cabo una serie de experimentos para evaluar las capacidades de generalización del método propuesto, aplicando el enfoque de entrenamiento progresivo presentado por Hasan [19], y utilizando el conjunto de evaluación CityPersons como conjunto de destino, de manera de comparar los resultados obtenidos con los del estado del arte. Se siguió la misma secuencia de entrenamiento en cascada presentada en el estado del arte para poder comparar los resultados obtenidos con los publicados. Se hizo el primer entrenamiento

Tabla 4.8: Resumen de resultados para evaluación cruzada sobre CityPersons. Todos los detectores usaron HRNet como *backbone*. Las 2 últimas filas muestran el método desarrollado (Faster R-CNN TL y Cascade R-CNN TL). En negrita se destacan los mejores resultados.

Método	**Conj. Entr.**	**Reasonable**	**Small**	**Heavy**	**All**
Cascade R-CNN	WiderPedestrian	16,0	21,6	57,4	-
CSP	WiderPedestrian	17,0	22,4	58,2	-
Faster R-CNN TL	WiderPedestrian	15,1	18,7	52,9	42,9
Cascade R-CNN TL	WiderPedestrian	**13,8**	**18,1**	**50,9**	**39,7**

en la base de datos WiderPedestrian y luego, con el mejor resultado obtenido (última fila de la Tabla 4.8), se hizo un ajuste fino usando el subconjunto diurno de EuroCity Persons. En la Tabla 4.9, se observan los resultados de hacer *fine tuning* sobre EuroCity Persons, para distintos pesos de *Triplet Loss*.

Tabla 4.9: Resultados MR^{-2} de generalización, usando un detector Cascade R-CNN con *Triplet Loss* en la primera cabeza de clasificación (H1 en la Figura 3.4), entrenado en WiderPedestrian, luego *fine tuned* en EuroCity persons, y evaluado en cada partición de CityPersons. En negrita se destacan los mejores resultados.

Peso Triplet Loss	**Reasonable**	**Small**	**Heavy**	**All**
0,025	10,2	12,2	**35,9**	**30,4**
0,050	10,3	11,6	39,7	32,3
0,100	**9,9**	**11,0**	36,2	30,8
0,150	10,4	11,5	36,4	30,8
0,200	10,4	11,9	37,2	31,5

Los mejores resultados obtenidos para esta serie de experimentos se muestran en la Tabla 4.10. Se obtuvo el mejor resultado con un peso de 0,1 para la primera cabeza de clasificación (H1) en el detector Cascade R-CNN. En resumen, se entrenó un detector en WiderPedestrian, utilizando un peso de 0,15 en la primera cabeza H1, y luego se hizo ajuste fino en la base de datos EuroCity Persons, utilizando un peso de 0,1 en la cabeza H1. Como se muestra en la Tabla 4.10, los resultados obtenidos superan al estado del arte actual en dos de los tres subconjuntos de CityPersons, es decir, los subconjuntos *Small* y *Heavy*, y se obtuvo un desempeño comparable en el subconjunto *Reasonable*. El incremento en desempeño es aproximadamente un 1,5 % para el subconjunto *Heavy*, que ha demostrado ser el subconjunto más difícil en CityPersons en la literatura [19, 62, 64, 65]. Se puede hipotetizar que los resultados en el conjunto de destino mejorarán con el método propuesto al utilizar bases de datos adicionales en el proceso de entrenamiento, que estén mas cerca del dominio objetivo, ya que el rendimiento base es mejor.

Tabla 4.10: Resumen de resultados para evaluación sobre CityPersons, realizando pruebas cruzadas, usando el flujo de entrenamiento progresivo. Todos los detectores usan HRNet como *backbone*. Los resultados de la última fila muestran el método propuesto (Cascade R-CNN TL). En negrita se destacan los mejores resultados.

Método	**Conj. Entr.**	**Reasonable**	**Small**	**Heavy**	**All**
Cascade R-CNN	CH → ECP	10,3	12,6	40,7	-
Cascade R-CNN	WP → ECP	**9,7**	11,8	37,7	-
Cascade R-CNN TL	WP → ECP	9,9	**11,0**	**36,2**	30,8

4.5. Evaluación en el *benchmark* Caltech

Una vez finalizadas las pruebas sobre la base de datos CityPersons, si hicieron algunos experimentos sobre el *benchmark* Caltech, de manera de comparar los resultados obtenidos con los publicados en el estado del arte. Se entrenó un detector Cascade R-CNN, y se agregó *Triplet Loss* solo en la primera cabeza, donde se obtuvieron los mejores resultados en los experimentos anteriores. Los resultados del modelo, evaluados en Caltech, se observan el la Tabla 4.11.

Tabla 4.11: Resultados MR^{-2} de generalización, usando un detector Cascade R-CNN con *Triplet Loss* en la primera cabeza de clasificación (H1 en la Figura 3.4), entrenado en WiderPedestrian, luego *fine tuned* en EuroCity persons, y evaluado en cada partición de Caltech. En negrita se destacan los mejores resultados.

Peso Triplet Loss	**Reasonable**	**Heavy**	**All**
0,025	4,4	40,5	46,5
0,050	3,7	35,5	47,9
0,100	3,5	**34,5**	**45,0**
0,150	3,6	38,2	45,7
0,200	**3,4**	38,0	46,0

En la Tabla 4.12, se observan los mejores resultados del estado del arte, comparados con el método desarrollado. Se puede observar que el método desarrollado obtuvo buenos resultados, solo obteniendo resultados peores que el mejor método del estado del arte. Aquí, el método desarrollado que mejor resultado obtuvo, es el que tiene un peso para *Triplet Loss* en la primera cabeza de $0,1$.

Tabla 4.12: Valores de MR^{-2} para diferentes métodos del estado del arte, evaluados en la base de datos Caltech. Los mejores resultados se muestran en negrita.

Método	**Conj. Entr.**	**Reasonable**	**Small**	**Heavy**
Repulsion Loss [18]	Caltech	5,0	5,2	47,9
ALFNet [17]	Caltech	6,1	7,9	51,0
CSP (ResNet50) [62]	Caltech	5,0	6,8	46,6
Cascade R-CNN [19]	WP $\rightarrow$ ECP	**2,5**	**9,9**	**31,0**
Cascade R-CNN TL	WP $\rightarrow$ ECP	3,5	-	34,5

4.6. Comparación de resultados cualitativos y discusión

En los siguientes ejemplos, se pueden observar resultados cualitativos del modelo propuesto, aplicado a la base de datos CityPersons, los cuales se comparan al método CSP [62], donde las imágenes en la parte superior, (a), muestran los resultados del método propuesto, mientras que las imágenes en la parte inferior, (b), muestran los resultados de CSP. Los falsos negativos se muestran como *bounding boxes* blancos, los falsos positivos en rojo y los verdaderos positivos en verde. Los casos de ejemplos fueron escogidos de manera que muestren situaciones donde es complejo detectar peatones, ya sea porque existen muchos peatones a distinta escala, cuando existen muchas oclusiones, donde otros métodos del estado del arte fallan en forma significativa, y donde las anotaciones poseen pequeños errores, donde no hay peatones anotados, pero nuestro método es capaz de detectarlos.

En las Figuras 4.1a, 4.2a y 4.3a, se muestran algunos falsos negativos al observar los resultados obtenidos por el método propuesto, pero si se observa detenidamente, se aprecia que la mayoría de las detecciones omitidas ocurren en áreas donde otra persona genera la oclusión. Cuando hay otros tipos de oclusión, por ejemplo, generada por automóviles u otros objetos, el método desarrollado funcionó bien, incluso en peatones de baja visibilidad y tamaño pequeño. Específicamente, en las Figuras 4.1 y 4.2, se puede observar una diferencia significativa en el rendimiento entre ambos métodos, donde el método propuesto es capaz de detectar en un número mayor a los peatones con mayor grado de dificultad, al comparar con CSP, especialmente aquellos que son pequeños o presentan un alto grado de oclusión.

En la Figura 4.3a, el método propuesto muestra mejores resultados cuando se aplica a peatones con oclusión. En términos generales, el método desarrollado muestra algunos errores que son debidos principalmente a oclusiones causadas por otros peatones, a la vez que se muestra que maneja las oclusiones mejor que CSP, como se aprecia en la Figura 4.3b. Al

realizar un análisis cualitativo de todo el conjunto de pruebas, la razón antes señalada parece ser la principal causa de que el método desarrollado presente algunos problemas al momento de enfrentar peatones con un alto grado de oclusión. Cabe destacar que tanto ciclistas como automovilistas no se detectan, esto es de manera intencional, ya que pertenecen a otra clase en la base de datos Cityscapes. En la Figura 4.2a, también se puede observar un falso positivo en rojo, pero al observarlo detenidamente, parece ser un peatón altamente ocluido, el que no está anotado en esa ubicación. La Figura 4.4a muestra muchas instancias de falsos positivos, y también un falso negativo, que parece ser causado por otro peatón, como en las Figuras 4.1a, 4.2a y 4.3a. Observando detenidamente, se puede ver que todos los *bounding boxes* en rojo son efectivamente peatones, algunos de los cuales están altamente ocluidos, pero son contados como falsos positivos de acuerdo al procedimiento de evaluación estándar presente en la literatura. Al comparar los resultados obtenidos con los de CSP, mostrados en la Figura 4.4b, se aprecia que solo uno de estos errores etiquetados como falsos positivos es detectado por CSP, lo que en conclusión presenta un rendimiento inferior, como tampoco detecta un peatón pequeño que está presente al medio de la escena. Finalmente, en la Figura 4.5a, se puede observar una única detección, señalada como un falso positivo, en rojo, pero al observar cuidadosamente, se aprecia que es un peatón con un alto grado de oclusión. Este peatón no es detectado por CSP, como se puede ver en la Figura 4.5b.

De los resultados cualitativos obtenidos, se puede observar que el método propuesto funciona bien en imágenes difíciles, manejando las oclusiones y distintas escalas de mejor manera que CSP [62], incluso es capaz de detectar peatones de alta dificultad, por ejemplo, los que no están anotados pero el método los detecta (*bounding boxes* rojos), pero que en realidad son peatones, donde algunos presentan un alto grado de oclusión. El método propuesto es capaz de detectar peatones en distintas escalas, con distinta iluminación y diferentes grados de oclusión, e incluso es capaz de evitar la detección de ciclistas, que tienen un alto grado de similitud visual a los peatones, lo que crea un nivel de dificultad adicional. Estos resultados se pueden explicar por el diseño de *Triplet Loss*, el cual trata en forma explícita de juntar todos los tipos de peatones en el espacio de características, independiente de sus tamaños o grados de oclusión, lo que obliga a la red a aprender que dichos peatones se mantengan cerca. En las Figuras 4.1–4.5, se observan varios ejemplos de detecciones de buen desempeño, las cuales poseen peatones con un alto grado de oclusión.

Una limitación que se observa es cuando las oclusiones son generadas por otros peatones, en lugar de objetos, por ejemplo, vehículos o árboles, por lo que en un trabajo futuro, se pueden enfocar los esfuerzos en tratar de resolver este tipo de oclusiones. Esto se podría explicar por la poca densidad de oclusiones de las bases de datos existentes en el estado del arte, lo que repercute en que el método presente algunas fallas en este escenario. Se

debe mencionar que este problema no ha sido bien estudiado hasta el momento, debido a la inherente complejidad del mismo. Si bien hay bases de datos que aportan con oclusiones intraclases, aún no es suficiente, y estas oclusiones siguen estando subrepresentadas. Como se puede observar en las Figuras 4.1 a 4.5, las oclusiones intraclase donde falla el método, poseen un alto valor de IoU, en general superior a 0,5. Otro factor importante a mencionar es que el modelo es capaz de detectar un peatón en cierta área, pero el que está ocluido no tiene suficiente información para detectarlo. Una posible solución al problema de las oclusiones generadas por peatones que están muy cerca, sería sobre muestrear ejemplos de bases de datos donde el IoU entre peatones sea muy alto (lo que implica que son cercanos), de manera que este tipo de oclusión posea una representación más alta de la que tendría al usar la base de datos sin aplicar este sobre muestreo. Una vez obtenido esto, se podría aplicar un *data augmentation* agresivo para minimizar la redundancia generada por el sobre muestreo.

Los resultados obtenidos también mostraron que el método desarrollado generaliza bien a partir de bases de datos con diferentes escenarios. La base de datos en la que se evaluó el método, CityPersons, posee imágenes capturadas en diferentes ciudades de Alemania y países vecinos, durante tres estaciones del año, y presentando diversas condiciones climáticas. Por otro lado, los datos de entrenamiento provienen de WiderPedestrian y EuroCity Persons, donde WiderPedestrian está compuesto por imágenes capturadas en escenarios de vigilancia y de conducción de automóviles, con ángulos de cámara, escala de objetos e iluminación muy diferentes, donde incluso existen algunas imágenes capturadas de noche. La base de datos EuroCity Persons fue capturada en 31 ciudades de 12 países europeos, abarcando una gran área geográfica, durante las cuatro estaciones del año, lo que implica una variedad de estilos de ropa, es decir, ligera/corta para el verano y gruesa/larga para el invierno, y condiciones climáticas secas o húmedas. Tener todas estas diferentes condiciones en el conjunto de entrenamiento, y la capacidad de *Triplet Loss* para agrupar ejemplos de la misma clase, en este caso, peatones, permite que el detector propuesto generalice bien, porque puede hacer que las proyecciones de los ejemplos de peatones queden más cerca en el espacio de características, incluso si se capturan bajo diferentes condiciones, por ejemplo, en invierno y verano, con lluvia, nieve, diferentes grados de oclusión, etc. Incluso si hay menos ejemplos de un escenario comparado con otros, estos se ven obligados a estar cerca en el espacio de características.

Es importante señalar que los resultados de nuestro método al entrenar solo con CityPersons, son peores que al entrenar usando el flujo de entrenamiento progresivo, lo que se puede ver comparando los resultados de la Tabla 4.5 con la Tabla 4.9, pasando de un MR^{-2} de 12,7 a 9,9 en la partición *Reasonable* usando Cascade R-CNN. Esto se debe a la gran cantidad de parámetros que poseen las CNN modernas, por lo que el conjunto de entrenamiento de

CityPersons no tiene suficientes ejemplos para poder obtener los mejores resultados. Algunos métodos logran resultados mejores, pero haciendo ajustes específicos a los datos usados (Modificando las anclas de la RPN por ejemplo) de manera que funcionen bien en la misma base de datos, pero sin probar la capacidad de generalización de dichos métodos. Por otro lado, en nuestro trabajo, se usan detectores de objetos genéricos, a los cuales se les agrega la cabeza desarrollada, y luego se aplica el esquema de entrenamiento progresivo, mostrando mejores resultados.

Finalmente, se debe señalar que se hizo una prueba para calcular la varianza intraclases, tanto para la clase peatón como fondo, usando la salida de la RPN. Para esto, se usó el mejor modelo obtenido con la cabeza nueva, entrenado en WiderPedestrian, y ajustado sobre EuroCity Persons, y se comparó con un modelo con el mismo flujo de entrenamiento, pero sin la cabeza nueva. La prueba se realizó sobre un subconjunto de la partición de validación de CityPersons, y se obtuvieron los resultados que se muestran en la Tabla 4.13. Debido a que los ejemplos de la clase fondo son muchos mas que la clase peatón, se puede afirmar que en este caso, la varianza total interclase disminuyó.

Tabla 4.13: Comparación de varianzas intraclase para dos modelos, uno con la cabeza nueva, y el otro sin, ambos entrenados en WiderPedestrian y ajustados en EuroCity Persons. La varianza se calculó en un subconjunto de la partición de validación de CityPersons.

Modelo	**Var. Peatones**	**Var. Fondo**
Cascade R-CNN TL	0,8959	0,7209
Cascade R-CNN	0,8821	0,9190

Como podemos observar en la Tabla 4.13, la varianza para la clase peatones, en ambos modelos es similar, pero en la clase fondo, se produce una gran diferencia, donde el modelo con *Triplet Loss* disminuye en forma significativa la varianza para esta clase. Se deben hacer mas experimentos para confirmar el real efecto de *Triplet Loss* en las varianzas, dado que el peso empleado para esta función de costo es pequeño comparado a los de las funciones de regresión y clasificación.

4.7. Análisis del costo computacional

El costo computacional, en términos del tiempo de ejecución de agregar la función *Triplet Loss* a las funciones de pérdida de clasificación y regresión existentes, es pequeño según mediciones realizadas. Durante el entrenamiento, el tiempo computacional aumenta en aproximadamente un 1 %, con un tiempo promedio de iteración de 0,6849s con *Triplet Loss*, en

comparación con 0,6799s sin ella, usando una GPU NVIDIA GeForce GTX 1080Ti. Esto podría explicarse por el hecho de que todos los cálculos realizados para la nueva función de pérdida requieren un cálculo de distancia dentro de un mini-bacth que consume la mayor parte del tiempo. Durante la inferencia, el tiempo computacional es el mismo para los casos con y sin *Triplet Loss*, porque esta función solo se usa durante el tiempo de entrenamiento. Por lo tanto, la función *Triplet Loss* no tiene un impacto negativo al usar el modelo para la detección de peatones, una vez que la red está entrenada.

En términos generales, al hacer el análisis de complejidad computacional de *Triplet Loss*, se obtiene una complejidad computacional de $O(N^3/C)$ por época, donde N es el número de muestras y C es el número de clases, con $C < N$. Debido a que las bases de datos son cada vez más grandes, el entrenamiento basado en *Triplet Loss* resulta computacionalmente desafiante. Se han hecho muchos esfuerzos en reducir esta complejidad sin afectar significativamente la efectividad del entrenamiento, donde una de las principales ideas exploradas es el diseño de mecanismos que seleccionen subconjuntos representativos de las muestras de tripletes con orden $O(N^3)$, como la minería de tripletes difíciles o semi-difíciles, o la minería inteligente de tripletes. Sin embargo, estos métodos todavía presentan una alta complejidad computacional, con un peor caso de complejidad de entrenamiento de $O(N^2)$ [94].

Los tiempos de ejecución, cuando se entrenó el sistema con distintas bases de datos, se pueden observar en la Tabla 4.15 para el caso de Faster R-CNN, y en la Tabla 4.14 para el caso de Cascade R-CNN. Todas las pruebas de cálculo de tiempos de ejecución fueron realizadas usando una GPU NVIDIA GeForce GTX 1080Ti.

Tabla 4.14: Tiempos de ejecución en entrenamiento, para Faster R-CNN, usando la nueva cabeza propuesta. Se entrenó por 30 épocas, y el *batch size* fue de tamaño 1.

Base de datos	**Nro. ejemplos**	**Tpo. iteración (s)**	**Tpo. ejecución (h)**
CityPersons	2.975	0,53209	13,2
Wider Pedestrian	90.000	0,46905	351,8

Tabla 4.15: Tiempos de ejecución en entrenamiento, para Cascade R-CNN, usando la nueva cabeza propuesta. Se entrenó por 30 épocas, y el *batch size* fue de tamaño 1.

Base de datos	**Nro. ejemplos**	**Tpo. iteración (s)**	**Tpo. ejecución (h)**
CityPersons	2.975	0,68561	17,0
EuroCity Persons	21.975	0,68488	125,4
Wider Pedestrian	90.000	0,63636	477,3

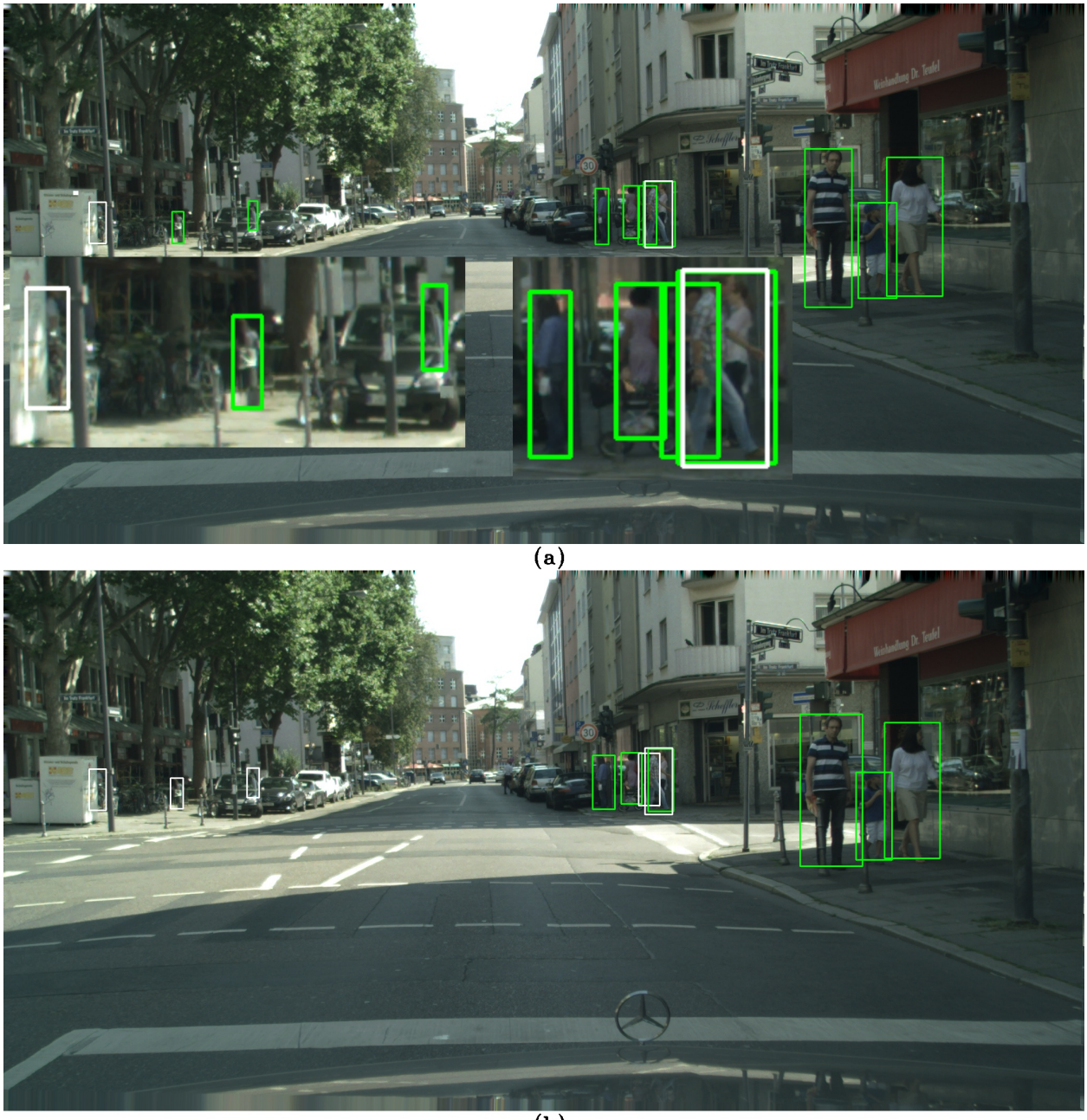
De las tablas 4.15 y 4.14, podemos observar que el entrenamiento de los modelos es bastante

lento, esto considerando que en una etapa inicial del *pipeline* de entrenamiento progresivo, primero se entrena en la base de datos Wider Pedestrian, lo que toma 20 días para el caso de Cascade R-CNN, para luego hacer *fine tuning* sobre EuroCity Persons, lo que toma mas de 5 días. Esto implica que obtener el modelo completo, tarda alrededor de 25 días en ser entrenado.

Por otro lado, en la Tabla 4.16, se pueden observar los tiempos de ejecución y la velocidad de procesamiento, para ambos métodos, Faster R-CNN y Cascade R-CNN, evaluados sobre las 500 imágenes de la base de datos CityPersons.

Tabla 4.16: Velocidad de ejecución en inferencia, sobre la base de datos CityPersons (500 ejemplos), para Faster R-CNN y Cascade R-CNN, usando la nueva cabeza propuesta.

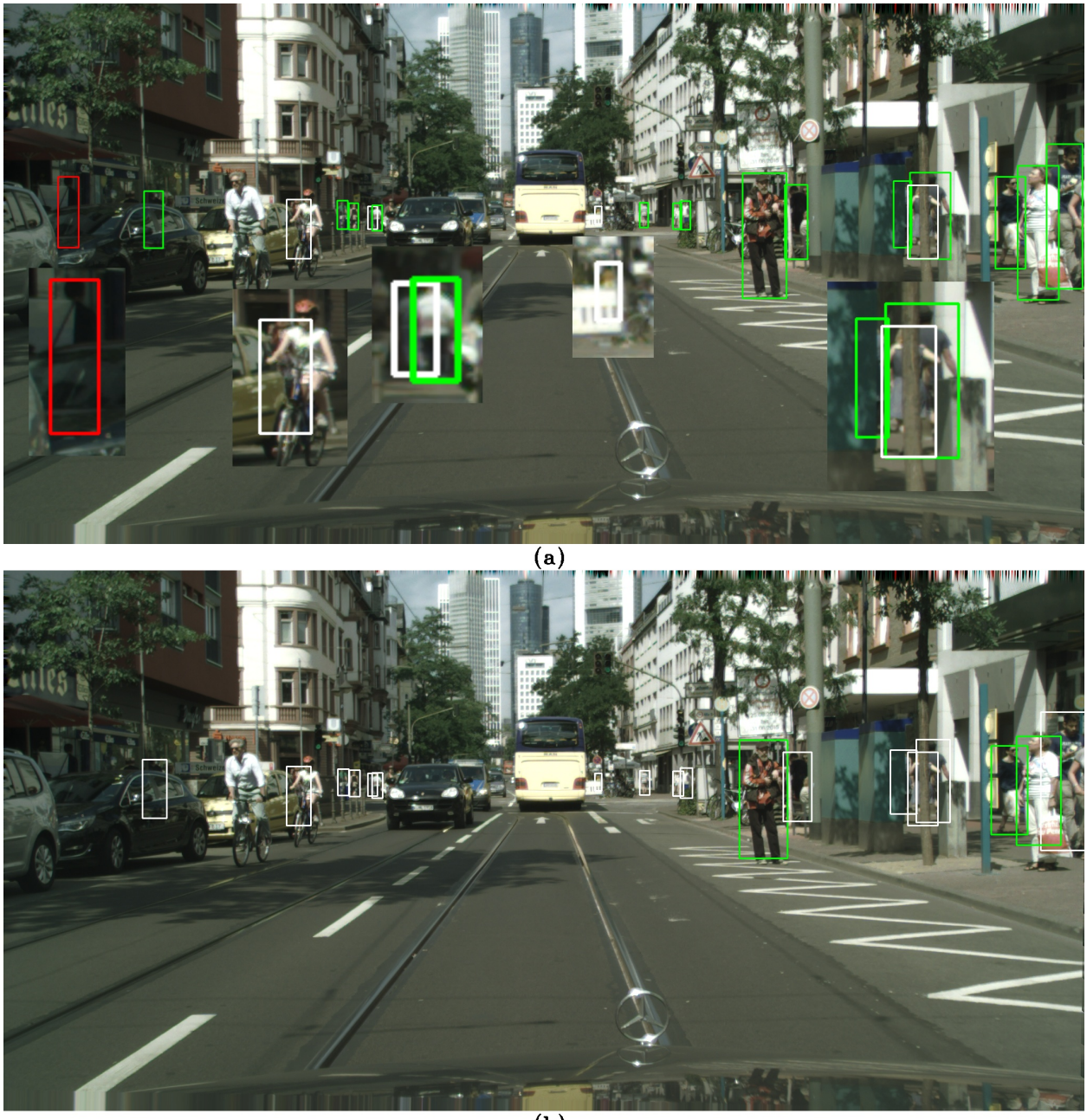
Método	**Tiempo total (s)**	**Imgs./seg.**
Faster R-CNN	97	5,2
Cascade R-CNN	107	4.7



(a)

(b)

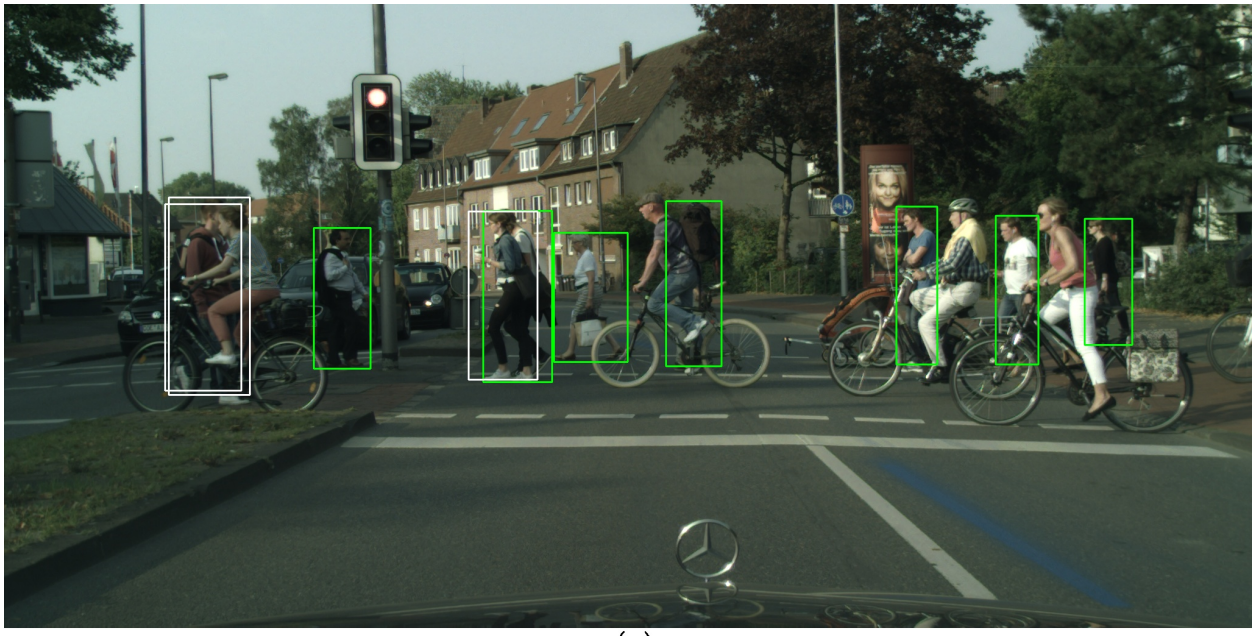
Figura 4.1: En (**a**), se muestran los resultados en CityPersons utilizando el método desarrollado. En (**b**), se muestran los resultados en CityPersons utilizando CSP [62]. Los falsos negativos se muestran en blanco, y los verdaderos positivos en verde. Los casos mas significativos se muestran con zoom para las detecciones del método propuesto. En (**a**), a la derecha, se aprecia un falso negativo causado por otro peatón. A la izquierda, un peatón de tamaño pequeño no es detectado, dado el alto grado de oclusión que presenta, causado por un objeto. Se aprecia que el método presenta un buen desempeño en peatones pequeños, como los presentes a la izquierda de la imagen. Finalmente, se observa que el método desarrollado obtiene mejores resultados en comparación a CSP, el cual se muestra en (**b**), especialmente en los peatones que están al lado izquierdo de la imagen, donde CSP no es capaz de detectar ningún peatón.



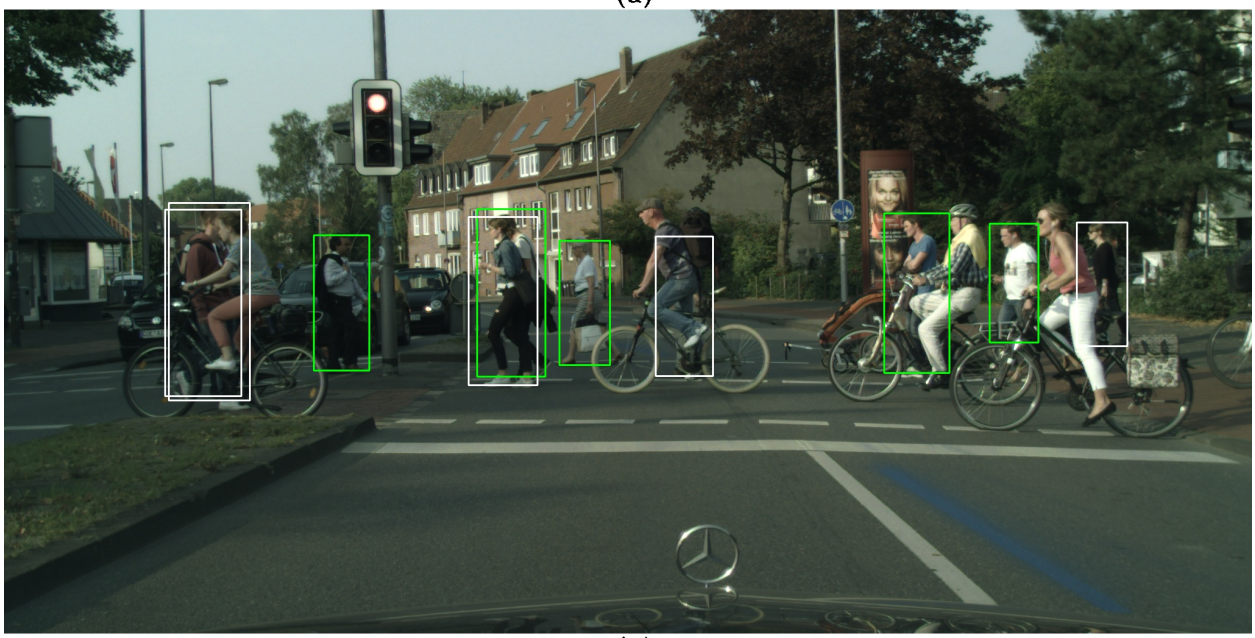
(a)

(b)

Figura 4.2: En (**a**), se muestran los resultados en CityPersons utilizando el método desarrollado. En (**b**), se muestran los resultados en CityPersons utilizando CSP [62]. Los falsos negativos se muestran en blanco, los falsos positivos en rojo, y los verdaderos positivos en verde. Los casos mas significativos se muestran con zoom para las detecciones del método propuesto. En (**a**), a la izquierda, se observan dos detecciones perdidas y un falso positivo, pero el falso positivo es efectivamente un peatón. A la derecha, se aprecian dos detecciones perdidas. Este ejemplo muestra un buen rendimiento en peatones de altura promedio y baja. También se observa que el método desarrollado presenta un mejor desempeño en la porción izquierda de la imagen, en comparación con CSP, que se muestra en (**b**), ya que CSP no detecta peatones pequeños. Además, a la derecha, se muestran varios falsos negativos, en blanco, en comparación con el método desarrollado, que detecta a la mayoría de los peatones.

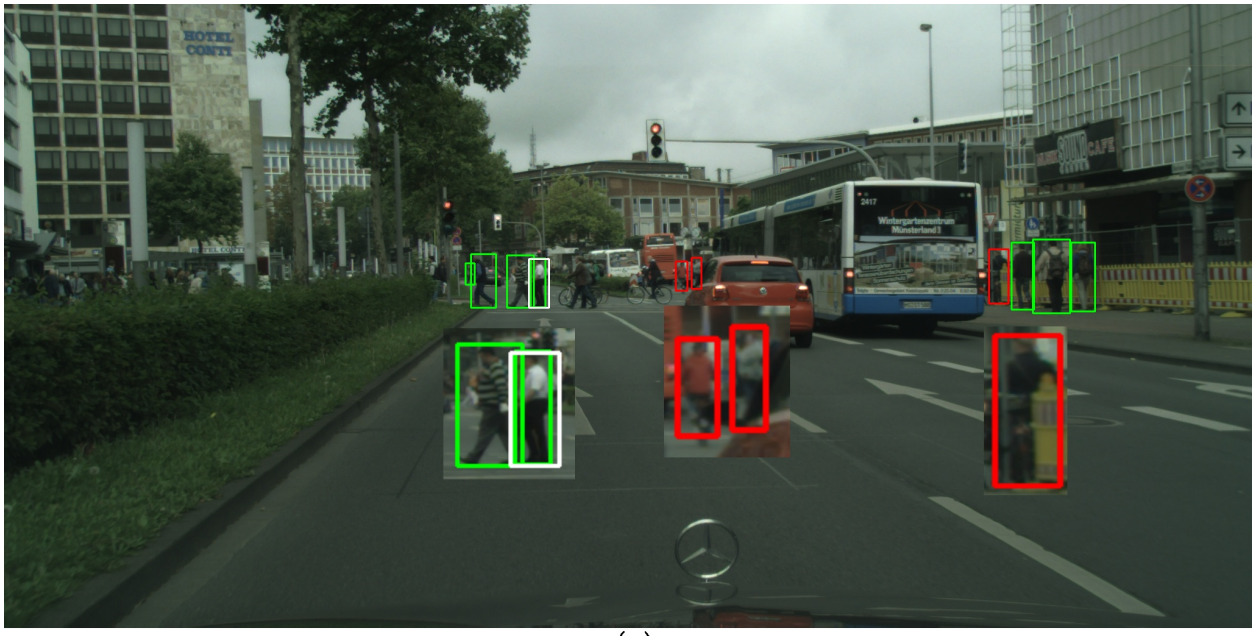


(a)



(b)

Figura 4.3: En (**a**), se muestran los resultados en CityPersons utilizando el método desarrollado. En la figura (**b**), se muestran los resultados en CityPersons utilizando CSP [62]. Los falsos negativos se muestran en blanco y los verdaderos positivos en verde. Con el método desarrollado, a la izquierda, se muestran dos falsos negativos, los cuales fueron generados por otro peatón. También hay otro falso negativo en el medio, también causado por otro peatón. Cabe señalar que los ciclistas se ignoran intencionalmente en las anotaciones, porque no pertenecen a la clase de peatones. El método CSP, que se muestra en la figura (**b**), genera dos falsos negativos adicionales, en el medio y en la parte derecha de la imagen.



(a)

(b)

Figura 4.4: En (**a**), se muestran los resultados en CityPersons utilizando el método desarrollado. En (**b**), se muestran los resultados en CityPersons utilizando CSP [62]. Los falsos negativos se muestran en blanco, los falsos positivos en rojo y los verdaderos positivos en verde. Los casos mas significativos se muestran con zoom para las detecciones del método propuesto. En (**a**) se observa un falso negativo causado por otro peatón, en el medio. Además, en el medio y a la derecha, se observan cajas rojas que se informan como falsos positivos, pero si observamos cuidadosamente, son peatones que no están anotados en la base de datos CityPersons. En los resultados del método CSP, en (**b**), podemos ver un peatón de baja estatura que no es detectado en el medio, y los dos peatones de baja estatura que el método desarrollado detecta, pero que no están anotados, CSP no es capaz de detectarlos.

(a)

(b)

Figura 4.5: En (**a**), se muestran los resultados en CityPersons utilizando el método desarrollado. En (**b**), se muestran los resultados en CityPersons utilizando CSP [62]. En (**a**), el método desarrollado muestra, en rojo, un falso positivo, el cual está en la parte derecha de la imagen. Al observar en forma cuidadosa (Ver zoom), se aprecia que efectivamente hay un peatón en ese lugar, el cual no está anotado en la base de datos CityPersons. Dicho peatón no es detectado por CSP, tal como se observa en (**b**).

Capítulo 5

Conclusiones

La detección de peatones es una de las tareas clave en el área de visión computacional, para la cual se han desarrollado varios modelos en los últimos años, que han mostrado una mejora constante con los métodos basados en *Deep Learning*. Muchas aplicaciones del mundo real requieren un alto rendimiento en la detección de peatones como, por ejemplo, los vehículos autónomos, la navegación robótica, la vigilancia de video, el reconocimiento de acciones y el *tracking*, lo que hace que este tema esté siendo activamente investigado en la comunidad científica. En este trabajo, se desarrolló un nuevo método de detección de peatones, que utiliza una nueva cabeza de clasificación, diseñada para ser usada en detectores de objetos de dos etapas. Este método tiene como objetivo mejorar las capacidades de generalización de dominio de los detectores de objetos existentes aplicados a peatones. Se agregó una tercera función de pérdida, basada en *Triplet Loss*, a las funciones de pérdida de clasificación y de regresión de *bounding boxes*, y se aplicó a los *embeddings* generados para las regiones de interés, que a su vez fueron generados por la red de propuestas de regiones, RPN. Por otro lado, se pudo observar que el método disminuye la varianza interclase en general para las muestras en el espacio de características, lo que tiene como efecto que los ejemplos tiendan a agruparse en dicho espacio, aunque deben realizarse mas experimentos para confirmarlo.

Al comparar los resultados del método desarrollado frente a los del estado del arte, se puede observar que el método desarrollado obtuvo resultados similares cuando fue entrenado y evaluado con CityPersons, tanto para Faster R-CNN como Cascade R-CNN. Por otra parte, se obtuvieron resultados mejores a los del estado del arte para el *benchmark* CityPersons, al entrenar con una base de datos distintas a la base objetivo, en este caso, usando la base de datos WiderPedestrian. En otras palabras, se hizo sin entrenar el método explícitamente dentro del conjunto objetivo que, en este caso, sería la partición de entrenamiento de CityPersons.

Donde se mostró realmente el poder de generalización del método, fue mediante el uso del esquema de entrenamiento de *pipeline* progresivo, el cual consiste en entrenar en bases de datos más lejanas del dominio objetivo, e irse acercando a este a medida que se agregan más dominios al flujo de entrenamiento. En este caso, se comenzó entrenando en WiderPedestrian, la cual está más alejada de CityPersons, dado que está compuesta por imágenes capturadas en escenarios de vigilancia y conducción de automóviles. Luego, se hizo ajuste fino en una base de datos más cercana, EuroCity Persons, la cual está compuesta de imágenes capturadas en un escenario de conducción de automóviles en Europa. Con esto, se logró una mejora significativa en la partición *Heavy*, la cual según se ha demostrado en los resultados del estado del arte, resulta ser la más compleja de las particiones disponibles en CityPersons. En la base de datos CityPersons, se obtuvo una métrica MR^{-2} de 9,9 para el subconjunto de datos *Reasonable*, 11,0 para el subconjunto *Small* y 36,2 para el subconjunto *Heavy*, lo que supera los resultados del estado del arte actuales para los subconjuntos *Small* y *Heavy*, y es altamente competitivo para el subconjunto *Reasonable*.

Los resultados obtenidos muestran que el método desarrollado generaliza bien a partir de bases de datos con diferentes escenarios, esto gracias a la capacidad de *Triplet Loss* para agrupar ejemplos de la misma clase, permitiendo que el detector propuesto generalice, debido a que las proyecciones de los ejemplos de peatones quedan más cerca en el espacio de características. Este efecto se aprecia incluso cuando las condiciones de captura son radicalmente distintas, por ejemplo, en invierno y verano, con lluvia, nieve, diferentes grados de oclusión, escala, etc., incluso si hay menos ejemplos de un escenario comparado a otros, estos se ven obligados a estar cerca en el espacio de características.

Por otro lado, observando los resultados cualitativos presentados, se observa que el método propuesto funciona bien en imágenes difíciles, manejando las oclusiones y distintas escalas de mejor manera que el método del estado del arte CSP [62]. Este método es incluso capaz de detectar peatones de alta dificultad, algunos de ellos no estando anotados en las anotaciones oficiales de CityPersons, lo que se puede apreciar claramente en los *bounding boxes* rojos de las Figuras 4.4 y 4.5. También se observa que el método propuesto detecta peatones en distintas escalas, distinta iluminación y oclusión, e incluso es capaz de evitar la detección de ciclistas, que tienen un alto grado de similitud visual a los peatones, lo que crea un nivel de dificultad adicional.

5.1. Trabajo Futuro

A modo de trabajo futuro, y en función de los resultados expuestos en este trabajo, el método desarrollado se beneficiaría del entrenamiento con bases de datos adicionales en el *pipeline* de entrenamiento progresivo, ya que el rendimiento base del método desarrollado es significativamente más alto que los presentes en el estado del arte. Por otro lado, dado que todas las imágenes de CityPersons están capturadas de día, en un trabajo futuro se podrían explorar estrategias para entrenar y evaluar el método desarrollado usando bases de datos que contengan imágenes capturadas de noche, como la partición nocturna de la base de datos EuroCity Persons.

Como se observó en los resultados cualitativos, existe una limitación que ocurre cuando las oclusiones son generadas por otros peatones, en lugar de objetos, por ejemplo, vehículos o árboles, por lo que en un trabajo futuro, se pueden enfocar los esfuerzos en tratar de resolver este tipo de oclusiones en forma explícita, por ejemplo, reforzando este tipo de ejemplos al momento de entrenar.

Algunas otras posibles líneas de investigación relacionadas al trabajo desarrollado, pueden ser el uso de otros *backbones*, que han mostrado buen desempeño en otros problemas, como por ejemplo, los de la familia EfficientNet [95, 96]. Otra línea de investigación sería el uso de métodos basados en *Transformers* [97–99], que también han logrado resultados sorprendentes en otras tareas de visión computacional, pero se debe estudiar bien la factibilidad de la nueva cabeza con *Triplet Loss*, dado que la arquitectura de los detectores de objetos basados en *Transformers* difiere de los detectores CNN de dos etapas.

Bibliografía

[1] Schroff, F., Kalenichenko, D., y Philbin, J., "Facenet: A unified embedding for face recognition and clustering", en Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7-12 June 2015, pp. 815–823, 2015.

[2] Tapia, J. E. y Perez, C. A., "Clusters of features using complementary information applied to gender classification from face images", IEEE Access, vol. 7, pp. 79374–79387, 2019.

[3] Tapia, J. E. y Perez, C. A., "Gender classification from nir images by using quadrature encoding filters of the most relevant features", IEEE Access, vol. 7, pp. 29114–29127, 2019.

[4] Zambrano, J. E., Benalcazar, D. P., Perez, C. A., y Bowyer, K. W., "Iris recognition using low-level cnn layers without training and single matching", IEEE Access, vol. 10, pp. 41276–41286, 2022.

[5] Galdames, F. J., Perez, C. A., Estevez, P. A., y Adams, M., "Rock lithological instance classification by hyperspectral images using dimensionality reduction and deep learning", Chemometrics and Intelligent Laboratory Systems, vol. 224, p. 104538, 2022.

[6] Perez, C. A., Estévez, P. A., Galdames, F. J., Schulz, D. A., Perez, J. P., Bastías, D., y Vilar, D. R., "Trademark image retrieval using a combination of deep convolutional neural networks", en Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8-13 July 2018, pp. 1–7, IEEE, 2018.

[7] Montecino, D. A., Perez, C. A., y Bowyer, K. W., "Two-level genetic algorithm for evolving convolutional neural networks for pattern recognition", IEEE Access, vol. 9, pp. 126856–126872, 2021.

[8] Perez, J. P. y Perez, C. A., "Face patches designed through neuroevolution for face recognition with large pose variation", IEEE Access, vol. 11, pp. 72861–72873, 2023, doi:10.1109/ACCESS.2023.3295330.

[9] Vilar, D. R. y Perez, C. A., "Extracting structured supervision from captions for weakly

supervised semantic segmentation", IEEE Access, vol. 9, pp. 65702–65720, 2021.

[10] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., y Zisserman, A., "The pascal visual object classes (voc) challenge", International journal of computer vision, vol. 88, pp. 303–308, 2009.

[11] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., y Zitnick, C. L., "Microsoft coco: Common objects in context", en Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pp. 740–755, Springer, 2014.

[12] Girshick, R., Donahue, J., Darrell, T., y Malik, J., "Rich feature hierarchies for accurate object detection and semantic segmentation", en Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014, pp. 580–587, 2014.

[13] Girshick, R., "Fast r-cnn", en Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7-13 December 2015, pp. 1440–1448, 2015.

[14] Ren, S., He, K., Girshick, R., y Sun, J., "Faster r-cnn: Towards real-time object detection with region proposal networks", en Advances in Neural Information Processing Systems (Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., y Garnett, R., eds.), vol. 28, Curran Associates, Inc., 2015, https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.

[15] Cai, Z. y Vasconcelos, N., "Cascade r-cnn: Delving into high quality object detection", en Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18-22 June 2018, pp. 6154–6162, 2018.

[16] Cao, J., Song, C., Peng, S., Song, S., Zhang, X., Shao, Y., y Xiao, F., "Pedestrian detection algorithm for intelligent vehicles in complex scenarios", Sensors, vol. 20, no. 13, p. 3646, 2020.

[17] Liu, W., Liao, S., Hu, W., Liang, X., y Chen, X., "Learning efficient single-stage pedestrian detectors by asymptotic localization fitting", en Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8-14 September 2018, pp. 618–634, 2018.

[18] Wang, X., Xiao, T., Jiang, Y., Shao, S., Sun, J., y Shen, C., "Repulsion loss: Detecting pedestrians in a crowd", en Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18-22 June 2018, pp. 7774–7783, 2018.

[19] Hasan, I., Liao, S., Li, J., Akram, S. U., y Shao, L., "Generalizable pedestrian detection: The elephant in the room", en Proceedings of the IEEE/CVF Conference on Computer

Vision and Pattern Recognition, virtual, 19-25 June 2021, pp. 11328–11337, 2021.

[20] Dollar, P., Wojek, C., Schiele, B., y Perona, P., "Pedestrian detection: An evaluation of the state of the art", IEEE transactions on pattern analysis and machine intelligence, vol. 34, no. 4, pp. 743–761, 2011.

[21] Hbaieb, A., Rezgui, J., y Chaari, L., "Pedestrian detection for autonomous driving within cooperative communication system", en 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 15-18 April 2019, pp. 1–6, IEEE, 2019.

[22] Hattori, H., Naresh Boddeti, V., Kitani, K. M., y Kanade, T., "Learning scene-specific pedestrian detectors without real data", en Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7-12 June 2015, pp. 3819–3827, 2015.

[23] Huang, L., Zhao, X., y Huang, K., "Bridging the gap between detection and tracking: A unified approach", en Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea (South), October 27 - November 2 2019, pp. 3999–4009, 2019.

[24] National Highway Traffic Safety Administration (NHTSA), "Overview of motor vehicle traffic crashes in 2021", 2023, https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/813435 (visitado el 2023-10-10).

[25] European Road Safety Observatory, "Facts and figures – pedestrians - 2023", 2023, https://road-safety.transport.ec.europa.eu/system/files/2023-02/ff_pedestrians_20230213.pdf (visitado el 2023-10-10).

[26] Dollár, P., Belongie, S. J., y Perona, P., "The fastest pedestrian detector in the west", en Proceedings of the British Machine Vision Conference, Aberystwyth, UK, 31 August - 3 September 2010, 2010, https://api.semanticscholar.org/CorpusID:7540163.

[27] Li, F., Li, X., Liu, Q., y Li, Z., "Occlusion handling and multi-scale pedestrian detection based on deep learning: a review", IEEE Access, 2022.

[28] Zhang, S., Benenson, R., Omran, M., Hosang, J., y Schiele, B., "How far are we from solving pedestrian detection?", en Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27-30 June 2016, pp. 1259–1267, 2016.

[29] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., y Berg, A. C., "Ssd: Single shot multibox detector", en Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11-14 October 2016, pp. 21–37, Springer, 2016.

[30] Redmon, J., Divvala, S., Girshick, R., y Farhadi, A., "You only look once: Unified, real-

time object detection", en Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27-30 June 2016, pp. 779–788, 2016.

[31] Zhang, L., Lin, L., Liang, X., y He, K., "Is faster r-cnn doing well for pedestrian detection?", en Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11-14 October 2016, pp. 443–457, Springer, 2016.

[32] Zhou, K., Liu, Z., Qiao, Y., Xiang, T., y Loy, C. C., "Domain generalization: A survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 4, pp. 4396–4415, 2023.

[33] Blanchard, G., Deshmukh, A. A., Dogan, Ü., Lee, G., y Scott, C., "Domain generalization by marginal transfer learning", The Journal of Machine Learning Research, vol. 22, no. 1, pp. 46–100, 2021.

[34] Parkhi, O., Vedaldi, A., y Zisserman, A., "Deep face recognition", en Proceedings of the British Machine Vision Conference,Swansea, UK, 7-10 September 2015, British Machine Vision Association, 2015.

[35] Trigueros, D. S., Meng, L., y Hartnett, M., "Enhancing convolutional neural networks for face recognition with occlusion maps and batch triplet loss", Image and Vision Computing, vol. 79, pp. 99–108, 2018.

[36] Boutros, F., Damer, N., Kirchbuchner, F., y Kuijper, A., "Self-restrained triplet loss for accurate masked face recognition", Pattern Recognition, vol. 124, p. 108473, 2022.

[37] Yeung, H. W. F., Li, J., y Chung, Y. Y., "Improved performance of face recognition using cnn with constrained triplet loss layer", en Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14-19 May 2017, pp. 1948–1955, IEEE, 2017.

[38] Feng, Y., Wang, H., Hu, H. R., Yu, L., Wang, W., y Wang, S., "Triplet distillation for deep face recognition", en Proceedings of the 2020 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 25-28 October 2020, pp. 808–812, IEEE, 2020.

[39] Si, T., Zhang, Z., y Liu, S., "Compact triplet loss for person re-identification in camera sensor networks", Ad Hoc Networks, vol. 95, p. 101984, 2019.

[40] Fan, X., Jiang, W., Luo, H., Mao, W., y Yu, H., "Instance hard triplet loss for in-video person re-identification", Applied Sciences, vol. 10, no. 6, p. 2198, 2020.

[41] Zhou, Q., Zhong, B., Lan, X., Sun, G., Zhang, Y., Zhang, B., y Ji, R., "Fine-grained spatial alignment model for person re-identification with focal triplet loss", IEEE Transactions on Image Processing, vol. 29, pp. 7578–7589, 2020.

[42] Yin, Q., Wang, G., Wu, J., Luo, H., y Tang, Z., "Dynamic re-weighting and cross-camera learning for unsupervised person re-identification", Mathematics, vol. 10, no. 10, p. 1654, 2022.

[43] Mihaescu, R.-E., Chindea, M., Paleologu, C., Carata, S., y Ghenescu, M., "Person re-identification across data distributions based on general purpose dnn object detector", Algorithms, vol. 13, no. 12, p. 343, 2020.

[44] Chen, W., Chen, X., Zhang, J., y Huang, K., "Beyond triplet loss: a deep quadruplet network for person re-identification", en Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21-26 July 2017, pp. 403–412, 2017.

[45] Lee, S.-w., "Domain generalization with triplet network for cross-corpus speech emotion recognition", en Proceedings of the 2021 IEEE Spoken Language Technology Workshop (SLT), Shenzhen, China, 19-22 January 2021, pp. 389–396, IEEE, 2021.

[46] Yu, B., Liu, T., Gong, M., Ding, C., y Tao, D., "Correcting the triplet selection bias for triplet loss", en Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8-14 September 2018, pp. 71–87, 2018.

[47] Wang, S., Yu, L., Li, C., Fu, C.-W., y Heng, P.-A., "Learning from extrinsic and intrinsic supervisions for domain generalization", en Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23-28 August 2020, pp. 159–176, Springer, 2020.

[48] Dou, Q., Coelho de Castro, D., Kamnitsas, K., y Glocker, B., "Domain generalization via model-agnostic learning of semantic features", Advances in neural information processing systems, vol. 32, 2019.

[49] Deng, W., Zheng, L., Sun, Y., y Jiao, J., "Rethinking triplet loss for domain adaptation", IEEE Transactions on Circuits and Systems for Video Technology, vol. 31, no. 1, pp. 29–37, 2020.

[50] Viola, P. y Jones, M. J., "Robust real-time face detection", International journal of computer vision, vol. 57, no. 2, pp. 137–154, 2004.

[51] Viola, P., Jones, M. J., y Snow, D., "Detecting pedestrians using patterns of motion and appearance", International journal of computer vision, vol. 63, pp. 153–161, 2005.

[52] Papageorgiou, C. y Poggio, T., "A trainable system for object detection", International journal of computer vision, vol. 38, no. 1, pp. 15–33, 2000.

[53] Dalal, N. y Triggs, B., "Histograms of oriented gradients for human detection", en Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20-26 June 2005, vol. 1, pp. 886–893, IEEE, 2005.

[54] Dollár, P., Appel, R., Belongie, S., y Perona, P., "Fast feature pyramids for object detection", IEEE transactions on pattern analysis and machine intelligence, vol. 36, no. 8, pp. 1532–1545, 2014.

[55] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., y Ramanan, D., "Object detection with discriminatively trained part-based models", IEEE transactions on pattern analysis and machine intelligence, vol. 32, no. 9, pp. 1627–1645, 2010.

[56] He, K., Gkioxari, G., Dollár, P., y Girshick, R., "Mask r-cnn", en Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22-29 October 2017, pp. 2961–2969, 2017.

[57] Angelova, A., Krizhevsky, A., Vanhoucke, V., Ogale, A. S., y Ferguson, D., "Real-time pedestrian detection with deep network cascades", en Proceedings of the British Machine Vision Conference, Swansea, UK, 7-10 September 2015, 2015, https://api.semanticscholar.org/CorpusID:15230091.

[58] Cai, Z., Saberian, M., y Vasconcelos, N., "Learning complexity-aware cascades for deep pedestrian detection", en Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7-13 December 2015, pp. 3361–3369, 2015.

[59] Hosang, J., Omran, M., Benenson, R., y Schiele, B., "Taking a deeper look at pedestrians", en Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7-12 June 2015, pp. 4073–4082, 2015.

[60] Brazil, G., Yin, X., y Liu, X., "Illuminating pedestrians via simultaneous detection & segmentation", en Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22-29 October 2017, pp. 4950–4959, 2017.

[61] Zhou, C. y Yuan, J., "Bi-box regression for pedestrian detection and occlusion estimation", en Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8-14 September 2018, pp. 135–151, 2018.

[62] Liu, W., Liao, S., Ren, W., Hu, W., y Yu, Y., "High-level semantic feature detection: A new perspective for pedestrian detection", en Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16-20 June 2019, pp. 5187–5196, 2019.

[63] Yin, R., Zhang, R., Zhao, W., y Jiang, F., "Da-net: pedestrian detection using dense connected block and attention modules", IEEE Access, vol. 8, pp. 153929–153940, 2020.

[64] Lin, C.-Y., Xie, H.-X., y Zheng, H., "Pedjointnet: Joint head-shoulder and full body deep network for pedestrian detection", IEEE Access, vol. 7, pp. 47687–47697, 2019.

[65] Cai, J., Lee, F., Yang, S., Lin, C., Chen, H., Kotani, K., y Chen, Q., "Pedestrian as points: An improved anchor-free method for center-based pedestrian detection", IEEE

Access, vol. 8, pp. 179666–179677, 2020.

[66] Li, C., Wang, Y., y Liu, X., "An improved yolov7 lightweight detection algorithm for obscured pedestrians", Sensors, vol. 23, no. 13, p. 5912, 2023.

[67] Liu, X. y Lin, Y., "Yolo-gw: Quickly and accurately detecting pedestrians in a foggy traffic environment", Sensors, vol. 23, no. 12, p. 5539, 2023.

[68] Zhang, S., Benenson, R., y Schiele, B., "Citypersons: A diverse dataset for pedestrian detection", en Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21-26 July 2017, pp. 3213–3221, 2017.

[69] Munder, S. y Gavrila, D. M., "An experimental study on pedestrian classification", IEEE transactions on pattern analysis and machine intelligence, vol. 28, no. 11, pp. 1863–1868, 2006.

[70] Ess, A., Leibe, B., y Van Gool, L., "Depth and appearance for mobile scene analysis", en Proceedings of the 2007 IEEE 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 14-20 October 2007, pp. 1–8, IEEE, 2007.

[71] Wojek, C., Walk, S., y Schiele, B., "Multi-cue onboard pedestrian detection", en Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20-25 June 2009, pp. 794–801, IEEE, 2009.

[72] Loy, C. C., Lin, D., Ouyang, W., Xiong, Y., Yang, S., Huang, Q., Zhou, D., Xia, W., Li, Q., Luo, P., *et al.*, "Wider face and pedestrian challenge 2018: Methods and results", arXiv preprint arXiv:1902.06854, 2019.

[73] Geiger, A., Lenz, P., y Urtasun, R., "Are we ready for autonomous driving? the kitti vision benchmark suite", en Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16-21 June 2012, pp. 3354–3361, IEEE, 2012.

[74] Braun, M., Krebs, S., Flohr, F., y Gavrila, D. M., "Eurocity persons: A novel benchmark for person detection in traffic scenes", IEEE transactions on pattern analysis and machine intelligence, vol. 41, no. 8, pp. 1844–1861, 2019.

[75] Shao, S., Zhao, Z., Li, B., Xiao, T., Yu, G., Zhang, X., y Sun, J., "Crowdhuman: A benchmark for detecting human in a crowd", arXiv preprint arXiv:1805.00123, 2018.

[76] Cheng, D., Gong, Y., Zhou, S., Wang, J., y Zheng, N., "Person re-identification by multi-channel parts-based cnn with improved triplet loss function", en Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27-30 June 2016, pp. 1335–1344, 2016.

[77] Wang, Y., Han, C., Yao, G., y Zhou, W., "Mapd: An improved multi-attribute pedestrian

detection in a crowd”, Neurocomputing, vol. 432, pp. 101–110, 2021.

[78] Dong, X. y Shen, J., “Triplet loss in siamese network for object tracking”, en Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8-14 September 2018, pp. 459–474, 2018.

[79] Unde, A. S. y Rameshan, R. M., “Mots r-cnn: Cosine-margin-triplet loss for multi-object tracking”, arXiv preprint arXiv:2102.03512, 2021.

[80] Yin, J., Wang, W., Meng, Q., Yang, R., y Shen, J., “A unified object motion and affinity model for online multi-object tracking”, en Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13-19 June 2020, pp. 6768–6777, 2020.

[81] Bredin, H., “Tristounet: triplet loss for speaker turn embedding”, en Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5-9 March 2017, pp. 5430–5434, IEEE, 2017.

[82] Li, C., Ma, X., Jiang, B., Li, X., Zhang, X., Liu, X., Cao, Y., Kannan, A., y Zhu, Z., “Deep speaker: an end-to-end neural speaker embedding system”, arXiv preprint arXiv:1705.02304, 2017.

[83] Ren, F. y Xue, S., “Intention detection based on siamese neural network with triplet loss”, IEEE Access, vol. 8, pp. 82242–82254, 2020.

[84] Zhang, M., Cheng, Q., Luo, F., y Ye, L., “A triplet nonlocal neural network with dual-anchor triplet loss for high-resolution remote sensing image retrieval”, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 2711–2723, 2021.

[85] Hazra, S. y Santra, A., “Short-range radar-based gesture recognition system using 3d cnn with triplet loss”, IEEE Access, vol. 7, pp. 125623–125633, 2019.

[86] Doras, G. y Peeters, G., “A prototypical triplet loss for cover detection”, en Proceedings of the ICASSP 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4-8 May 2020, pp. 3797–3801, IEEE, 2020.

[87] Sun, Z., Hu, S., Song, H., y Liang, P., “Learning wasserstein contrastive color histogram representation for low-light image enhancement”, Mathematics, vol. 11, no. 19, p. 4194, 2023.

[88] Song, X., Zhao, K., Chu, W.-S., Zhang, H., y Guo, J., “Progressive refinement network for occluded pedestrian detection”, en Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16, pp. 32–48, Springer, 2020.

[89] Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., *et al.*, "Deep high-resolution representation learning for visual recognition", IEEE transactions on pattern analysis and machine intelligence, vol. 43, no. 10, pp. 3349–3364, 2020.

[90] Li, J., Liao, S., Jiang, H., y Shao, L., "Box guided convolution for pedestrian detection", en Proceedings of the 28th ACM International Conference on Multimedia, Virtual Event / Seattle, WA, USA, 12-16 October 2020, pp. 1615–1624, 2020.

[91] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., y Schiele, B., "The cityscapes dataset for semantic urban scene understanding", en Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27-30 June 2016, pp. 3213–3223, 2016.

[92] Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C. C., y Lin, D., "MMDetection: Open mmlab detection toolbox and benchmark", arXiv preprint arXiv:1906.07155, 2019.

[93] Simonyan, K. y Zisserman, A., "Very deep convolutional networks for large-scale image recognition", en 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[94] Do, T.-T., Tran, T., Reid, I., Kumar, V., Hoang, T., y Carneiro, G., "A theoretically sound upper bound on the triplet loss for improving the efficiency of deep distance metric learning", en Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16-20 June 2019, pp. 10404–10413, 2019.

[95] Tan, M. y Le, Q., "Efficientnet: Rethinking model scaling for convolutional neural networks", en Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, California, USA, June 9-15 2019, pp. 6105–6114, PMLR, 2019.

[96] Tan, M. y Le, Q., "Efficientnetv2: Smaller models and faster training", en Proceedings of the 38th International Conference on Machine Learning, ICML 2021, July 18-24 2021, Virtual Event, pp. 10096–10106, PMLR, 2021.

[97] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale", arXiv preprint arXiv:2010.11929, 2020.

[98] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., y Guo, B., "Swin transformer: Hierarchical vision transformer using shifted windows", en Proceedings of the

2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021, pp. 10012–10022, 2021.

[99] Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., *et al.*, "Swin transformer v2: Scaling up capacity and resolution", en Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022, pp. 12009–12019, 2022.

Anexos

Anexo A. Lista de acrónimos

Backbone: Red de extracción de características, que transforma los datos de entrada en una representación en un espacio de características, los cuales se pueden usar en modelos mas complejos. Ejemplos de backbones populares son las familias VGG, ResNet, Inception, etc.

Benchmark: Bases de datos para problemas específicos, las cuales se usan para medir el desempeño y poder comparar los resultados con otros trabajos en la literatura.

Bounding box: Región rectangular que suele delimitar un objeto de interés, ampliamente usado en tareas de visión computacional, especialmente en detección de objetos. En tiempo de entrenamiento, estas regiones son usadas para aprender acerca de los contenidos de una imagen.

CNN: Una red neuronal convolucional (CNN) es una arquitectura de red de aprendizaje profundo, que aprende directamente a partir de los datos, aplicando filtros a los datos de entrenamiento a distinta resolución, y cuya salida se usa como entrada en la capa siguiente.

Deep Learning: También llamado aprendizaje profundo, es un subconjunto del aprendizaje automático, que usa redes neuronales de múltiples capas, llamadas redes neuronales profundas, para simular la complejidad del funcionamiento del cerebro humano.

Embeddings: Representación en un espacio de dimensión relativamente baja de un vector de dimensionalidad mayor. Estos facilitan el aprendizaje automático, capturando parte de la semántica de la muestra al colocar muestras semánticamente similares cerca en el espacio de embeddings. Un embedding puede ser aprendido y reutilizado entre distintos modelos.

Ground truth: Información que se sabe que es real o verdadera, la cual ha sido obtenida mediante observación y medición directa. En el caso de detección de objetos, suele corres-

ponder a los bounding boxes y sus respectivas etiquetas, las cuales fueron demarcadas por un humano.

ImageNet: Base de datos que contiene 14.197.122 imágenes anotadas de acuerdo a la WordNet hierarchy, de las cuales 1.034.908 tienen anotadas bounding boxes. Ha sido usada ampliamente en tareas de visión computacional, especialmente para entrenar distintos backbones, que sirven de base para resolver otros problemas.

Mini-batch: Subconjuntos de la base de datos, con un número igual de ejemplos de entrenamiento, sobre los cuales se calcula el gradiente y se actualizan los pesos.

ROI: Una región de interés (ROI) es una porción de una imagen sobre la cual se desea realizar algún tipo de operación.

Softmax: La función softmax, convierte un vector de K números reales en una distribución de probabilidad de K resultados posibles. Es una generalización de la función logística a múltiples dimensiones, y se usa frecuentemente como la última función de activación de una red neuronal para normalizar la salida de la red a una distribución de probabilidad sobre clases de salida predichas.