

TRABAJO DE TITULO

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

VIÑA DEL MAR – CHILE



**INTELIGENCIA ARTIFICIAL PARA LA DETECCIÓN DE PERSONAS CON
MOVILIDAD REDUCIDA**

DIEGO IGNACIO VASQUEZ VALDIVIA

TRABAJO DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO EN INFORMÁTICA

Profesora Guía: Heilym Ramírez

Septiembre – 2026

TRABAJO DE TITULO



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: Inteligencia artificial para la detección de personas con movilidad reducida

Nombre del candidato(a): Diego Ignacio Vasquez Valdivia

Carrera / Grado: Ingeniería en informática

Campus: Viña del mar Departamento: electrotecnica e informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Heilym Ramirez _____, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 15 de octubre de 2026.

Firma:

Estudiante o Candidato(a): Diego Ignacio Vasquez Valdivia

Fecha: 15/10/2026

Firma:

DEDICATORIA

Dedico este trabajo a toda mi familia, por su apoyo constante, paciencia y confianza durante mi formación profesional. La culminación de esta investigación representa también el resultado del acompañamiento y del esfuerzo compartido a lo largo de este proceso.

Nada es imposible

AGRADECIMIENTOS

Agradezco a mi profesora guía, Heilym Ramírez, por la orientación proporcionada durante el desarrollo de esta investigación. Agradezco también a las personas que colaboraron en la recopilación, preparación, anotación y revisión de los datos, así como a quienes aportaron apoyo técnico durante los experimentos. agradezco a mi familia, papa, mama, y a mis hermanos, sin ustedes no estaría donde estoy ahora, gracias por todo el cariño, los malos y buenos ratos, todas esas veces que me dijeron que estudiara, aunque no quisiera, gracias por perseverar conmigo y nunca rendirse y siempre apoyarme. Por último, quería agradecer a mi pareja Emilia, gracias por todas esas horas a mi lado estudiando, cuidándome y brindándome apoyo, este logro sin duda es nuestro, gracias por todo el apoyo y la paciencia para permanecer a mi lado

RESUMEN

Esta investigación estudia la detección automática de personas con movilidad reducida mediante visión por computador y aprendizaje profundo, utilizando la familia de detectores YOLO11. La primera etapa compara YOLO11n, YOLO11s y YOLO11m sobre Mobility Aids y considera simultáneamente desempeño de detección y complejidad computacional. A partir de esa comparación se selecciona YOLO11s como arquitectura de referencia para los experimentos posteriores.

Posteriormente, se utiliza YOLO11s para estudiar el cambio de dominio entre Mobility Aids y un conjunto propio denominado Dataset USM. Se estudian cinco configuraciones: un modelo base entrenado con Mobility Aids (E1), fine-tuning desde Mobility Aids hacia USM (E2), entrenamiento exclusivo sobre USM (E3), entrenamiento extendido sobre USM (E4) y entrenamiento conjunto sobre Mobility Aids y USM desde los pesos preentrenados de YOLO11s (E5). Para las comparaciones interdominio se mantienen los mismos conjuntos de prueba de cada dominio.

YOLO11m presentó el mayor mAP50-95 en el benchmark de Mobility Aids realizado con la partición temporal corregida, con 0,631, mientras YOLO11s alcanzó 0,628 y requirió 21,3 GFLOPs frente a 67,7 GFLOPs de YOLO11m. Por ello, YOLO11s fue seleccionada por su compromiso entre desempeño y complejidad. En las corridas corregidas, E1 alcanzó 0,625 en Mobility Aids y 0,534 en USM; E2 alcanzó 0,549 y 0,852, respectivamente; y E5 obtuvo 0,633 y 0,851. Bajo un criterio interdominio basado exclusivamente en las tres clases comunes, E5 presentó el mayor promedio y el mayor valor mínimo entre las configuraciones comparables.

Palabras clave: detección de objetos; YOLO11; movilidad reducida; aprendizaje por transferencia; *fine-tuning*.

ABSTRACT

This thesis studies the automatic detection of people with reduced mobility using computer vision and deep learning, with a focus on the YOLO11 object detector family. The first stage compares YOLO11n, YOLO11s, and YOLO11m on Mobility Aids, considering both detection performance and computational complexity. YOLO11s is selected as the reference architecture for the subsequent experiments.

The second stage analyzes the domain shift between Mobility Aids and a self-built dataset, referred to as the USM Dataset. Five training configurations are studied: a Mobility Aids baseline (E1), fine-tuning from Mobility Aids to USM (E2), USM-only training (E3), extended USM training (E4), and joint training on Mobility Aids and USM from YOLO11s pretrained weights (E5). The same domain-specific test sets are used throughout the comparative evaluation.

YOLO11m achieved the highest mAP50-95 in the Mobility Aids architecture benchmark using the corrected temporal partition, with 0.631, while YOLO11s reached 0.628 with substantially lower computational complexity. In the corrected experiments, E1 reached 0.625 on Mobility Aids and 0.534 on USM; E2 reached 0.549 and 0.852; and E5 reached 0.633 and 0.851. Under a cross-domain criterion based only on the three shared classes, E5 provided the highest observed mean and minimum performance among the directly comparable configurations.

Keywords: object detection; YOLO11; reduced mobility; transfer learning; fine-tuning.

GLOSARIO

AI / IA: Artificial Intelligence / Inteligencia Artificial.

AP: Average Precision; resumen del comportamiento de precision y recall para una clase.

BCE: Binary Cross-Entropy; función de pérdida asociada a clasificación.

CNN: Convolutional Neural Network; red neuronal convolucional.

DFL: Distribution Focal Loss; componente de pérdida utilizado en la regresión distribucional de las cajas.

F1: Media armónica entre precision y recall.

FN: False Negative; instancia real que no es detectada.

FP: False Positive; detección que no corresponde a una instancia válida.

FPN: Feature Pyramid Network; estructura de fusión de características entre escalas.

GFLOPs: Giga Floating Point Operations; estimación del costo computacional en operaciones de punto flotante.

HMM: Hidden Markov Model; modelo utilizado en el trabajo antecedente para estimación probabilística de clase a través del tiempo.

IoU: Intersection over Union; medida de solapamiento entre dos cajas delimitadoras.

mAP: mean Average Precision; promedio de AP entre clases.

NMS: Non-Maximum Suppression; procedimiento para eliminar detecciones redundantes.

PAN: Path Aggregation Network; estructura de agregación de características en el detector.

PMR: Personas con movilidad reducida.

RPN: Region Proposal Network; componente de Faster R-CNN utilizado para generar regiones candidatas.

SGD: Stochastic Gradient Descent; método de optimización por descenso de gradiente estocástico.

TP: True Positive; detección correcta de una instancia.

YOLO: You Only Look Once; familia de detectores de objetos de una etapa.

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN	11
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	12
1.1 Contexto de la problemática	12
1.2 Personas con movilidad reducida y ayudas técnicas	12
1.3 Situación problemática	12
1.4 Justificación de la investigación	13
1.5 Estado del arte y antecedentes	13
1.6 Brecha identificada	14
1.7 Alcance	14
1.8 Hipótesis de investigación	14
1.9 Objetivos	15
CAPÍTULO 2: MARCO CONCEPTUAL	16
2.1 Inteligencia artificial, aprendizaje automático y aprendizaje profundo ...	16
2.2 Visión por computador	16
2.3 Detección de objetos	16
2.4 Redes neuronales convolucionales	16
2.5 Detectores de una y dos etapas	17
2.6 Familia YOLO	17
2.7 YOLO11	17
2.7.1 Backbone	17
2.7.2 Neck y fusión multiescala	18
2.7.3 Head de detección	18
2.7.4 Task-Aligned Assigner	18
2.8 Bounding boxes y ground truth	18
2.9 Intersection over Union (IoU)	18
2.10 Precision	18
2.11 Recall	19
2.12 F1-score	19
2.13 Average Precision y mAP	19
2.14 mAP50 y mAP50-95	19
2.15 Funciones de pérdida	19
2.16 Épocas, batch y learning rate	20

TRABAJO DE TITULO

2.17 Descenso de gradiente estocástico (SGD)	20
2.18 Data augmentation	20
2.19 Transfer learning	20
2.20 Fine-tuning	21
2.21 Domain shift y generalización	21
2.22 Data leakage temporal	21
2.23 GFLOPs y latencia	22
2.24 Síntesis del marco conceptual	22
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN	23
3.1 Enfoque general	23
3.2 Mobility Aids Dataset	23
3.3 Preparación de Mobility Aids	23
3.4 Partición temporal y distribución de clases	24
3.5 Dataset USM	25
3.6 Configuración experimental	26
3.7 Diseño de los cinco experimentos	27
3.8 Evaluación y criterio de selección final	28
3.9 Reproducibilidad y gestión experimental	28
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN	30
4.1 Consideraciones generales	30
4.2 Validación de la partición temporal final	30
4.3 Comparación entre YOLO11n, YOLO11s y YOLO11m	31
4.4 Selección de YOLO11s para los experimentos posteriores	33
4.5 Modelo base YOLO11s y cambio de dominio	34
4.6 Experimentos de adaptación y entrenamiento	35
4.6.1 Resultados sobre Mobility Aids	35
4.6.2 Resultados sobre USM	36
4.6.3 Análisis por clase	37
4.6.4 Análisis visual de errores	40
4.7 Fine-tuning Mobility Aids -> USM	40
4.8 Comparación con Kollmitz et al. (2019)	40
4.9 Discusión integrada	41
CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES	44
5.1 Síntesis de la investigación	44
5.2 Conclusiones metodológicas	45
5.3 Cumplimiento de objetivos	45
5.4 Evaluación de hipótesis	46
5.5 Limitaciones	46
5.6 Recomendaciones y trabajo futuro	47
5.7 Conclusión final	48
REFERENCIAS BIBLIOGRÁFICAS	50
ANEXOS	51

ÍNDICE DE FIGURAS

Figura 1. Arquitectura conceptual de YOLO11 utilizada en la investigación.	17
Figura 2. Distribución de objetos por clase en Mobility Aids.	25
Figura 3. Flujo de los cinco experimentos con YOLO11s.	27
Figura 4. Partición temporal final de Mobility Aids y criterios de auditoría.	31
Figura 5. Comparación de mAP50-95 entre YOLO11n, YOLO11s y YOLO11m.	32
Figura 6. Relación entre complejidad computacional y mAP50-95.	33
Figura 7. Desempeño del modelo base E1 en ambos dominios.	34
Figura 8. mAP50-95 de configuraciones comparables en Mobility Aids.	36
Figura 9. mAP50-95 de los cinco experimentos en USM.	37
Figura 10. AP@50-95 por clase en Mobility Aids.	38
Figura 11. AP@50-95 por clase en USM.	39
Figura 12. Promedio y mínimo de mAP50-95 entre ambos dominios.	42

ÍNDICE DE TABLAS

Tabla 1. Comparación de antecedentes relacionados.	14
Tabla 2. Composición de Mobility Aids por partición.	24
Tabla 3. Distribución de objetos por clase en Mobility Aids.	24
Tabla 4. Composición de Dataset USM por partición.	26
Tabla 5. Distribución de objetos por clase en Dataset USM.	26
Tabla 6. Configuración experimental principal de YOLO11s.	26
Tabla 7. Resumen de los cinco experimentos.	27
Tabla 8. Auditoría de la partición temporal final de Mobility Aids.	30
Tabla 9. Complejidad y desempeño del benchmark de variantes YOLO11.	31
Tabla 10. AP@50-95 por clase en test de Mobility Aids.	32
Tabla 11. Resultados globales del modelo base YOLO11s.	34
Tabla 12. Resultados globales de las configuraciones semánticamente comparables en test de Mobility Aids. ...	35
Tabla 13. Resultados globales en test de USM.	36
Tabla 14. AP@50-95 por clase en Mobility Aids para E1, E2 y E5.	37
Tabla 15. AP@50-95 por clase en USM.	38
Tabla 16. Comparación directa de las principales estrategias de adaptación.	40
Tabla 17. Comparación descriptiva con Kollmitz et al. (2019).	40
Tabla 18. Criterio de generalización equilibrada entre ambos dominios.	41
Tabla 19. Resumen metodológico de resultados corregidos.	43

INTRODUCCIÓN

La movilidad urbana constituye una dimensión relevante de la vida cotidiana, debido a que la posibilidad de desplazarse de forma autónoma y segura condiciona el acceso a espacios de estudio, trabajo, servicios y actividades sociales. Dentro de este contexto, las personas con movilidad reducida pueden encontrar dificultades asociadas tanto a sus condiciones individuales como a las características del entorno en el que se desplazan.

Las ayudas técnicas de movilidad, entre ellas las sillas de ruedas, las muletas y los andadores, permiten observar visualmente determinadas situaciones asociadas a movilidad reducida. La detección automática de estas ayudas mediante imágenes y videos puede constituir una fuente de información cuantificable para complementar procesos de observación y análisis de entornos urbanos. Este trabajo no pretende determinar clínicamente la condición de una persona; su objetivo técnico consiste en detectar visualmente las categorías definidas en los conjuntos de datos utilizados.

La visión por computador y el aprendizaje profundo han permitido desarrollar detectores capaces de localizar múltiples objetos en una imagen mediante cajas delimitadoras y etiquetas de clase. Dentro de este ámbito, la familia YOLO se caracteriza por utilizar un enfoque de detección de una etapa, lo que la hace pertinente cuando se requiere una relación entre desempeño y velocidad de inferencia. YOLO11 corresponde a una generación desarrollada por Ultralytics y utiliza una arquitectura compuesta por *backbone*, *neck* y *head*, con componentes orientados a la extracción y fusión multiescala de características [7], [8].

El principal antecedente utilizado en esta investigación es el trabajo de Kollmitz et al. de 2019, que estudia la percepción de personas según las ayudas técnicas de movilidad mediante Faster R-CNN, estimación de posición tridimensional y seguimiento probabilístico sobre el conjunto Mobility Aids [2]. La presente investigación toma ese conjunto como dominio inicial, pero delimita su alcance a la detección de objetos en imágenes.

La investigación se organiza en cuatro etapas. Primero, se estudia la partición de Mobility Aids y se comparan las variantes YOLO11n, YOLO11s y YOLO11m. Segundo, se selecciona YOLO11s para estudiar la generalización entre Mobility Aids y un segundo dominio, denominado Dataset USM. Tercero, se realizan cinco experimentos de entrenamiento y adaptación. Finalmente, se integran los resultados para determinar qué estrategia presenta un comportamiento más equilibrado entre ambos dominios y se establecen limitaciones y líneas de trabajo futuro.

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

1.1 Contexto de la problemática

Las personas con movilidad reducida pueden requerir condiciones específicas para desplazarse de manera autónoma en espacios públicos. En el contexto de esta investigación, se utilizan como señales visuales el uso de ayudas técnicas de movilidad, ya que estas pueden ser identificadas mediante técnicas de detección de objetos. La información obtenida por un detector no resuelve por sí misma las barreras de accesibilidad, pero puede constituir un insumo cuantificable para estudios posteriores.

Desde el punto de vista informático, la detección automática permite transformar imágenes en información estructurada. En particular, un detector puede determinar la ubicación aproximada de una instancia y asociarla a una clase. Esta capacidad permite estudiar, de manera automatizada, la presencia de categorías relacionadas con movilidad reducida en secuencias de video.

1.2 Personas con movilidad reducida y ayudas técnicas

En esta tesis se utiliza el término personas con movilidad reducida (PMR) para referirse a personas que presentan restricciones permanentes o temporales para desplazarse de forma autónoma. La definición utilizada debe distinguirse de las clases concretas del modelo. En Mobility Aids las categorías empleadas son person, crutches, wheelchair, walking_frame y push_wheelchair. Dataset USM, en cambio, contiene únicamente crutches, wheelchair y walking_frame. Por lo tanto, las categorías de los datasets deben entenderse como clases observables de detección y no como una clasificación clínica exhaustiva.

1.3 Situación problemática

La primera dificultad corresponde a la diferencia entre el dominio en que un detector aprende y aquel en que posteriormente es evaluado. Un modelo entrenado sobre imágenes capturadas en un contexto particular puede aprender características relacionadas con fondos, iluminación, distancia de cámara, perspectiva y composición de escena. Cuando estas condiciones cambian, la distribución visual de los datos también puede cambiar, fenómeno conocido como *domain shift* [5].

La segunda dificultad corresponde a la naturaleza secuencial de Mobility Aids. El conjunto contiene frames provenientes de videos, por lo que imágenes registradas en instantes cercanos pueden ser muy similares. Durante una etapa inicial de preparación se consideró una selección aleatoria de imágenes para entrenamiento y validación. Esta estrategia fue posteriormente descartada como procedimiento principal debido al riesgo de que frames visualmente similares

quedaran en ambos subconjuntos, generando una dependencia temporal que podía inflar las métricas de validación.

La tercera dificultad corresponde a la distribución desigual de las clases. En Mobility Aids la clase wheelchair representa 22,06 % de los objetos de entrenamiento, pero solo 7,60 % del conjunto de test. En Dataset USM, wheelchair alcanza 55,46 % de los objetos de entrenamiento, mientras que walking_frame representa 13,02 %. Estas diferencias hacen necesario complementar las métricas globales con análisis por clase.

1.4 Justificación de la investigación

La investigación se justifica en la necesidad de estudiar de manera reproducible el comportamiento de un detector moderno frente a diferencias de partición y de dominio visual. La comparación considera precision, recall, F1, mAP50 y mAP50-95, además del desempeño por clase. Los conjuntos de prueba de Mobility Aids y Dataset USM se mantienen constantes entre experimentos para facilitar la comparación interna, aunque sus resultados también fueron observados durante el proceso de selección y comparación de configuraciones; esta decisión se reconoce explícitamente como una limitación metodológica.

El trabajo también busca establecer estrategias diferenciadas de adaptación. E2 representa una adaptación secuencial desde un dominio inicial hacia USM, mientras E3 y E4 permiten estudiar el efecto de entrenar exclusivamente en USM y de prolongar ese entrenamiento. E5 evalúa un entrenamiento conjunto desde pesos preentrenados, incorporando simultáneamente información de Mobility Aids y USM. Esta organización permite distinguir especialización, adaptación secuencial y aprendizaje conjunto.

1.5 Estado del arte y antecedentes

Vasquez, Kollmitz, Eitel y Burgard desarrollaron en 2017 un sistema para detectar personas según las ayudas de movilidad utilizadas en un escenario de robótica de servicio. El método generaba regiones de interés a partir de información de profundidad y posteriormente las procesaba mediante Fast R-CNN. El trabajo reportó un mAP de 64,98 % y redujo el tiempo de procesamiento informado de 297 ms a 43 ms por frame [1].

Kollmitz et al. ampliaron este enfoque en 2019. El sistema utilizó Faster R-CNN, incorporó estimación de profundidad del centroide y añadió seguimiento mediante un filtro de Kalman extendido y un modelo oculto de Markov. El conjunto extendido contenía más de 17.000 imágenes RGB-D de 960 x 540 píxeles y consideraba cinco categorías relacionadas con movilidad [2]. Para la variante RGB se reportó un mAP de 0,87 y un tiempo de ejecución de 53 ms por imagen [2].

Dumitrescu, Boiangiu y Voncilă estudiaron en 2022 un enfoque de detección de personas basado en YOLOv4 complementado con una cabeza de clasificación y regresión inspirada en Faster R-CNN. El trabajo también evaluó el conjunto MobilityAids y discutió explícitamente la necesidad de evitar que una misma secuencia contenga imágenes distribuidas entre entrenamiento y validación, lo que constituye un antecedente metodológico relevante para la partición temporal utilizada en esta tesis [3].

En el ámbito de la adaptación de dominio aplicada a la detección, Chen et al. propusieron Domain Adaptive Faster R-CNN, señalando que diferencias entre las distribuciones de

entrenamiento y evaluación pueden producir pérdidas de desempeño. Su propuesta aborda el cambio de dominio a nivel de imagen y de instancia y constituye un antecedente relevante para interpretar los experimentos de cambio de dominio de esta tesis [13]. A diferencia de ese enfoque, la presente investigación no implementa un mecanismo adversarial de adaptación; compara empíricamente estrategias de entrenamiento secuencial y conjunto sobre dos dominios de detección.

Tabla 1. Comparación de antecedentes relacionados.

Trabajo	Modelo / enfoque	Datos o contexto	Resultado principal	Observación
Vasquez et al. (2017)	Fast R-CNN + generación de ROIs	Mobility Aids, profundidad	mAP 64,98	Enfoque para robótica de servicio.
Kollmitz et al. (2019)	Faster R-CNN + estimación 3D + seguimiento	Mobility Aids, RGB/depth	mAP 0,87 RGB	Incluye componentes 3D y temporales.
Presente investigación	YOLO11n/s/m y luego YOLO11s	Mobility Aids + Dataset USM	Evaluación multimétrica y entre dominios	Alcance centrado en detección 2D.

Fuente: Elaboración propia a partir de [1]-[3].

La Tabla 1 se utiliza para contextualizar las diferencias de enfoque. Las cifras no constituyen una comparación directa de superioridad, debido a que los trabajos utilizan arquitecturas, sensores, definiciones de métricas, hardware y objetivos diferentes.

1.6 Brecha identificada

La brecha abordada en esta tesis consiste en estudiar el comportamiento de un detector de una etapa sobre un conjunto de datos de movilidad utilizado como antecedente, respetando la estructura secuencial de sus imágenes y auditando la independencia entre las particiones de aprendizaje. Posteriormente, se analiza el cambio de dominio hacia un segundo conjunto construido de manera independiente. La contribución experimental se centra en comparar adaptación secuencial, especialización en un segundo dominio, extensión del entrenamiento y entrenamiento conjunto.

1.7 Alcance

- La investigación se limita a la detección de objetos en imágenes y secuencias de video.
- No se desarrolla seguimiento de individuos, reidentificación, estimación clínica de discapacidad ni estimación de profundidad tridimensional.
- Mobility Aids se utiliza como dominio de referencia y Dataset USM como segundo dominio experimental.
- El conjunto adicional obtenido mediante colaboración con el Metro de Valparaíso no se incorpora cuantitativamente al conjunto principal de resultados por restricciones de tiempo y se considera trabajo futuro.
- La evaluación se centra en las métricas de detección entregadas por el framework utilizado y en F1 calculado a partir de precision y recall.

1.8 Hipótesis de investigación

Hipótesis principal. YOLO11 puede detectar las categorías observables asociadas a movilidad reducida en los dos dominios estudiados, y las estrategias de adaptación entre dominios producen cambios medibles en el desempeño, particularmente cuando el modelo pasa de un dominio de origen a un dominio visual diferente.

Hipótesis secundaria. Considerando únicamente las tres clases compartidas por Mobility Aids y Dataset USM, el entrenamiento conjunto de ambos dominios puede producir un comportamiento más equilibrado entre ellos que el modelo base entrenado solo con Mobility Aids o que la adaptación secuencial desde Mobility Aids hacia USM.

1.9 Objetivos

Objetivo general. Desarrollar y validar un modelo de detección automática capaz de identificar personas con movilidad reducida en imágenes, evaluando su comportamiento sobre Mobility Aids y Dataset USM y estudiando estrategias de adaptación entre ambos dominios.

1. Comparar las variantes YOLO11n, YOLO11s y YOLO11m sobre Mobility Aids bajo un protocolo experimental común.
2. Diseñar y auditar una partición de Mobility Aids que considere la dependencia temporal entre frames y que permita demostrar la separación entre las unidades de entrenamiento y validación.
3. Construir y documentar Dataset USM, incluyendo su preparación, clases y particiones.
4. Evaluar el desempeño mediante precision, recall, mAP50 y mAP50-95.
5. Estudiar el efecto del *fine-tuning* desde Mobility Aids hacia USM.
6. Comparar entrenamiento exclusivo con USM, extensión por más épocas y entrenamiento con datos combinados.
7. Determinar una configuración de referencia considerando la generalización equilibrada entre ambos dominios y utilizando únicamente clases semánticamente comparables en el cálculo interdominio.

CAPÍTULO 2: MARCO CONCEPTUAL

2.1 Inteligencia artificial, aprendizaje automático y aprendizaje profundo

La inteligencia artificial agrupa métodos computacionales orientados a resolver tareas que requieren capacidades como reconocimiento de patrones, percepción o inferencia. Dentro de este campo, el aprendizaje automático se caracteriza por ajustar los parámetros de un modelo a partir de datos de entrenamiento. El aprendizaje profundo utiliza redes neuronales con múltiples capas capaces de aprender representaciones jerárquicas de la información [4].

En visión por computador, el aprendizaje profundo permite aprender representaciones directamente desde las imágenes. Esto resulta relevante para la detección de objetos porque el modelo debe aprender características que permitan distinguir clases y, simultáneamente, localizar sus instancias.

2.2 Visión por computador

La visión por computador comprende técnicas destinadas a obtener información a partir de imágenes o videos. Entre sus tareas se encuentran clasificación, segmentación, estimación de profundidad, seguimiento y detección de objetos. En la presente investigación se estudia específicamente la detección de objetos, ya que el resultado requerido combina una categoría semántica con una localización espacial.

2.3 Detección de objetos

Un problema de detección de objetos consiste en identificar las instancias de determinadas clases presentes en una imagen y estimar la ubicación de cada una. La salida de un detector suele representarse mediante una caja delimitadora, una clase y una puntuación de confianza. Una imagen puede contener varias instancias de una misma clase o de clases diferentes.

La calidad de una detección depende de dos componentes: la correcta identificación de la clase y la correcta localización espacial. Por este motivo, las métricas de evaluación utilizan tanto la correspondencia entre clases como el solapamiento de las cajas predichas con sus anotaciones de referencia.

2.4 Redes neuronales convolucionales

Las redes neuronales convolucionales (CNN) utilizan operaciones de convolución para transformar la imagen en mapas de características. En las primeras etapas pueden capturar patrones locales, mientras que las representaciones más profundas integran información sobre formas y estructuras de mayor nivel. Estas representaciones constituyen la base de numerosos detectores de objetos modernos [4].

2.5 Detectores de una y dos etapas

Los detectores de dos etapas separan conceptualmente la generación de regiones candidatas de la clasificación y regresión de cajas. Fast R-CNN utiliza regiones de interés previamente propuestas, mientras Faster R-CNN incorpora una Region Proposal Network (RPN) para producir regiones candidatas de manera integrada [2], [11], [12].

Los detectores de una etapa, como la familia YOLO, realizan la predicción de clases y cajas dentro de un flujo de detección único. Esta formulación busca reducir la complejidad temporal del proceso de detección y resulta pertinente cuando la latencia constituye una restricción del sistema [10].

2.6 Familia YOLO

YOLO, sigla de You Only Look Once, identifica una familia de detectores de objetos que evolucionó desde una formulación de una etapa hacia arquitecturas cada vez más complejas. El trabajo original planteó la detección como un problema de regresión desde la imagen completa hacia cajas y probabilidades de clase [10]. Versiones posteriores incorporaron mejoras en extracción de características, predicción multiescala y funciones de pérdida.

2.7 YOLO11

YOLO11 fue presentado por Ultralytics en 2024. La documentación de la arquitectura describe el modelo mediante tres componentes generales: *backbone*, *neck* y *head*. YOLO11 utiliza bloques C3k2 en la extracción de características, SPPF para ampliar el campo receptivo y C2PSA para introducir atención espacial. El cuello integra características a distintas escalas y la cabeza de detección es desacoplada y sin anclajes, utilizando DFL para la regresión distribucional [7], [8].

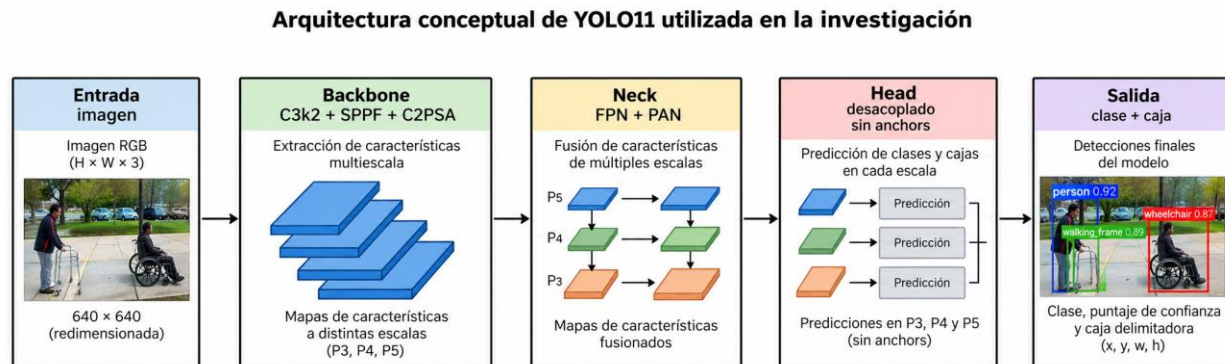


Figura 1. Arquitectura conceptual de YOLO11 utilizada en la investigación.

Fuente: Elaboración propia.

2.7.1 Backbone

El *backbone* es la parte de la red encargada de extraer características desde la imagen. En YOLO11 se utilizan bloques C3k2 y operaciones de reducción de resolución. Al aumentar la profundidad de la red se construyen representaciones progresivamente más abstractas, que posteriormente son utilizadas para detectar objetos de diferentes escalas [7].

2.7.2 Neck y fusión multiescala

El *neck* combina mapas de características provenientes de diferentes niveles de profundidad. La documentación de Ultralytics describe el uso de estructuras FPN y PAN para realizar fusión multiescala. Esta operación permite mantener información relevante tanto para objetos de gran tamaño como para objetos más pequeños [7].

2.7.3 Head de detección

La *head* transforma las características fusionadas en las predicciones finales. En YOLO11 la cabeza de detección es desacoplada y sin anclajes, con componentes separados para clasificación y regresión. La regresión de las cajas utiliza DFL para modelar las distancias respecto de los puntos de referencia de la predicción [7].

2.7.4 Task-Aligned Assigner

Durante el entrenamiento es necesario determinar qué predicciones son responsables de cada anotación real. El Task-Aligned Assigner utiliza información de clasificación y localización para seleccionar predicciones positivas que contribuyen al aprendizaje. Esta asignación influye en la señal de entrenamiento recibida por la red [7].

2.8 Bounding boxes y ground truth

Una *bounding box* es una representación rectangular utilizada para delimitar un objeto dentro de una imagen. En un conjunto de datos supervisado, las cajas anotadas forman parte del *ground truth*, que corresponde a la referencia con la cual se comparan las predicciones del modelo. En este trabajo las anotaciones se transformaron al formato requerido por YOLO.

2.9 Intersection over Union (IoU)

La intersección sobre unión, conocida como IoU, cuantifica el solapamiento entre una caja predicha y una caja de referencia. Su valor se obtiene como la razón entre el área de intersección y el área de unión de ambas cajas.

$$IoU = \text{Área de intersección} / \text{Área de unión}$$

El valor de IoU varía entre 0 y 1. Valores cercanos a 1 representan un mayor solapamiento. En detección de objetos, IoU se utiliza para determinar si una predicción puede considerarse correctamente localizada y constituye un componente central de las métricas AP y mAP [6], [8].

2.10 Precision

La precision mide qué proporción de las detecciones positivas realizadas por el modelo corresponde a instancias verdaderamente correctas. Se define, en términos de detecciones positivas, como la relación entre verdaderos positivos y el total de predicciones positivas.

$$Precision = TP / (TP + FP)$$

Una precision elevada indica que el detector genera una baja proporción de falsos positivos. En esta tesis la precision permite estudiar qué tan confiables son las detecciones producidas, especialmente cuando se cambia de dominio.

2.11 Recall

El recall representa la capacidad del modelo para recuperar las instancias que realmente existen en las imágenes. Se define mediante la relación entre verdaderos positivos y el total de objetos reales, considerando también los falsos negativos.

$$Recall = TP / (TP + FN)$$

Un recall elevado implica que el modelo omite menos objetos reales. En el contexto de la detección de personas con movilidad reducida, esta métrica es relevante porque un detector puede presentar alta precision y, simultáneamente, no detectar una parte importante de las instancias presentes.

2.12 F1-score

El F1-score combina precision y recall mediante su media armónica. La fórmula utilizada en los análisis de esta tesis es:

$$F1 = 2 \times Precision \times Recall / (Precision + Recall)$$

El F1 se utiliza como indicador complementario porque penaliza configuraciones que logran una métrica alta a costa de una métrica baja. En las tablas de esta tesis, el F1 marcado con asterisco se calcula a partir de los valores publicados de precision y recall y no corresponde a una métrica adicional extraída de un archivo de resultados independiente.

2.13 Average Precision y mAP

Average Precision (AP) resume la relación entre precision y recall para una clase determinada bajo un criterio de evaluación definido. Mean Average Precision (mAP) corresponde al promedio de AP entre las clases evaluadas. Por ello, mAP entrega una medida agregada, pero puede ocultar diferencias importantes entre clases.

En detección de objetos se pueden utilizar diferentes umbrales de IoU para definir una detección correcta. Cambiar este umbral modifica la exigencia de la localización espacial. La documentación de Ultralytics utiliza mAP50 para un único umbral de IoU de 0,50 y mAP50-95 para un promedio sobre múltiples umbrales entre 0,50 y 0,95 [8].

2.14 mAP50 y mAP50-95

mAP50 considera las detecciones bajo IoU igual a 0,50. Es una medida útil para evaluar si el detector identifica correctamente los objetos, pero es menos exigente respecto de la precisión de la localización que otros umbrales. mAP50-95 promedia el mAP obtenido en los umbrales 0,50, 0,55, ..., 0,95, por lo que entrega una evaluación más exigente del desempeño de localización [8].

En esta investigación mAP50-95 se utiliza como la principal métrica para discutir el equilibrio entre dominios, porque permite observar no solo si el objeto fue detectado, sino también la calidad del solapamiento de las cajas bajo distintos niveles de exigencia.

2.15 Funciones de pérdida

Durante el entrenamiento, las predicciones son comparadas con las anotaciones mediante funciones de pérdida. En la configuración de YOLO11 utilizada se consideran componentes

relacionados con clasificación, regresión de cajas y DFL. La pérdida total proporciona una señal que permite actualizar los parámetros de la red mediante un algoritmo de optimización.

La clasificación puede utilizar Binary Cross-Entropy (BCE) para medir la diferencia entre las predicciones y las clases objetivo. Para la localización se utiliza una función relacionada con IoU, como CloU, mientras DFL permite representar la regresión de las distancias de manera distribucional. La combinación y peso de estos componentes depende de la configuración utilizada por el framework [7].

2.16 Épocas, batch y learning rate

Una época corresponde a un recorrido completo por las imágenes del conjunto de entrenamiento. El *batch* corresponde al número de imágenes procesadas antes de actualizar los parámetros del modelo. El *learning rate* determina la magnitud de las actualizaciones de los parámetros durante la optimización.

El número de épocas y el learning rate no son variables independientes del resto de la configuración. Un learning rate demasiado elevado puede producir actualizaciones inestables, mientras un valor demasiado bajo puede ralentizar la convergencia. Del mismo modo, ampliar el número de épocas puede ser útil cuando el modelo todavía no ha aprendido suficientemente, pero no garantiza mejoras sobre datos pertenecientes a otros dominios.

2.17 Optimización mediante SGD

Stochastic Gradient Descent (SGD) actualiza los parámetros del modelo utilizando el gradiente de la función de pérdida calculado sobre lotes de entrenamiento. El uso de momentum incorpora información de actualizaciones anteriores para modificar el comportamiento del descenso. En el entrenamiento principal de esta investigación se utilizó SGD con momentum 0,9 y weight decay 0,0001, siguiendo los valores indicados en el script experimental basado en la configuración del trabajo de 2019.

2.18 Data augmentation

La *data augmentation* consiste en aplicar transformaciones a las imágenes de entrenamiento para aumentar la variabilidad efectiva de los datos. Entre las transformaciones habituales se encuentran cambios geométricos, variaciones de color y composición de imágenes. Su objetivo es reducir la dependencia del modelo respecto de ejemplos particulares y favorecer la generalización. En esta tesis, las intensidades concretas se reportan únicamente cuando forman parte de los scripts de entrenamiento entregados.

2.19 Transfer learning

El *transfer learning* corresponde al uso de conocimiento aprendido en un conjunto o tarea previa para facilitar el aprendizaje de un nuevo problema. En lugar de inicializar todos los parámetros sin información previa, se reutilizan pesos que contienen representaciones aprendidas. Pan y Yang presentan este enfoque como una forma de transferir conocimiento entre dominios o tareas relacionadas [5].

En el contexto de esta tesis, la carga de yolo11s.pt corresponde al inicio del entrenamiento desde pesos preentrenados. Esta condición es distinta de entrenar un modelo con inicialización aleatoria. La reutilización de pesos permite comenzar desde representaciones ya aprendidas y

concentrar el proceso de entrenamiento en adaptar esas representaciones al dominio específico.

2.20 Fine-tuning

El *fine-tuning* es una estrategia de adaptación en la cual un modelo previamente entrenado continúa su aprendizaje utilizando un nuevo conjunto de datos, normalmente con una configuración más conservadora que la empleada durante el entrenamiento inicial. El objetivo es ajustar las representaciones existentes al nuevo dominio sin perder innecesariamente el conocimiento adquirido [5], [9].

En esta investigación, el segundo experimento corresponde a un caso de adaptación secuencial: primero se entrena YOLO11s con Mobility Aids y luego se continúa desde el checkpoint last.pt utilizando USM. El learning rate se reduce de 0,0025 a 0,0005, se utilizan 50 épocas y se modifican otros parámetros de entrenamiento, incluyendo el warm-up y determinadas intensidades de augmentation. La estrategia utilizada se encuentra documentada en el script de fine-tuning entregado para este trabajo.

El quinto experimento utiliza una estrategia diferente: corresponde a un entrenamiento conjunto. El modelo se inicializa directamente con los pesos preentrenados de YOLO11s y se entrena utilizando simultáneamente datos de Mobility Aids y Dataset USM. Por tanto, E2 representa una adaptación secuencial hacia un nuevo dominio, mientras E5 evalúa el aprendizaje conjunto desde una inicialización común. Esta diferencia permite estudiar dos estrategias de cambio de dominio sin confundir el efecto de la adaptación secuencial con el del entrenamiento simultáneo sobre ambos dominios.

2.21 Domain shift y generalización

El *domain shift* se produce cuando la distribución de los datos de evaluación difiere de la distribución observada durante el entrenamiento. Las diferencias pueden originarse en cámaras, iluminación, fondo, distancia, perspectiva, oclusiones, población representada o composición de clases. La generalización corresponde a la capacidad del modelo para mantener un comportamiento adecuado sobre datos que no son idénticos a aquellos utilizados durante el ajuste.

La evaluación entre Mobility Aids y USM constituye en esta tesis un experimento de cambio de dominio. Un resultado elevado sobre un dominio no implica automáticamente un resultado equivalente sobre el otro, por lo que la evaluación cruzada es necesaria para estudiar este fenómeno.

2.22 Data leakage temporal

El *data leakage* temporal corresponde al riesgo de que información altamente relacionada temporalmente con el conjunto de evaluación esté presente durante el entrenamiento. En datasets contruidos a partir de video, frames consecutivos pueden compartir casi toda la composición visual. Una división aleatoria de imágenes puede distribuir ejemplos similares entre entrenamiento y validación y producir una estimación optimista.

Por esta razón, la tesis distingue entre una partición aleatoria preliminar y la partición temporal final por secuencias. En la partición definitiva, una secuencia completa se asigna a una sola

partición de aprendizaje y se audita la ausencia de solapamiento por video, por nombre de imagen y por hash de contenido. El conjunto de test publicado se mantiene fuera de este proceso.

2.23 GFLOPs y latencia

Los GFLOPs constituyen una estimación del número de operaciones de punto flotante necesarias para ejecutar una arquitectura bajo una configuración determinada. En esta tesis se utilizan como indicador de complejidad computacional al comparar YOLO11n, YOLO11s y YOLO11m. La latencia corresponde al tiempo requerido para producir una predicción y depende tanto del modelo como del hardware y del pipeline de ejecución.

Por este motivo, una menor latencia reportada en una plataforma no debe interpretarse automáticamente como una aceleración universal. En la comparación con Kollmitz et al. (2019), la diferencia de hardware y de componentes del sistema impide considerar los tiempos como una comparación controlada.

2.24 Síntesis del marco conceptual

Los conceptos desarrollados en este capítulo permiten establecer un marco común para interpretar los experimentos. La detección de objetos requiere simultáneamente identificar la clase y localizar cada instancia, mientras las métricas precision, recall, F1, mAP50 y mAP50-95 permiten observar distintas dimensiones del desempeño. IoU vincula la calidad espacial de las cajas con la decisión de considerar una predicción como correcta.

A su vez, los conceptos de transfer learning, fine-tuning y domain shift permiten interpretar los experimentos entre Mobility Aids y USM. E1 utiliza pesos preentrenados para establecer un modelo base; E2 continúa el aprendizaje desde el modelo base sobre USM; E3 entrena directamente sobre USM; E4 prolonga ese entrenamiento; y E5 entrena conjuntamente con ambos dominios desde una inicialización común. Finalmente, la partición temporal permite controlar la dependencia característica de datos provenientes de video.

Este marco conceptual delimita el alcance del estudio: la investigación evalúa detección 2D y no seguimiento, estimación de profundidad ni clasificación clínica. Por ello, las conclusiones se formulan exclusivamente a partir de las tareas, métricas, conjuntos de datos y configuraciones experimentales descritos en los capítulos siguientes.

CAPÍTULO 3: PROPUESTA DE SOLUCIÓN

3.1 Enfoque general

La propuesta consiste en desarrollar y evaluar un detector YOLO11 para identificar personas con movilidad reducida en imágenes mediante las clases observables disponibles en los conjuntos de datos. La metodología comienza con Mobility Aids para establecer una referencia experimental y posteriormente utiliza Dataset USM para analizar el cambio de dominio y las estrategias de adaptación.

El diseño experimental se separa en una etapa de selección de arquitectura y una etapa de evaluación de estrategias de entrenamiento. Primero se comparan YOLO11n, YOLO11s y YOLO11m en condiciones comunes. Una vez seleccionado YOLO11s para las pruebas posteriores, se mantiene la arquitectura fija y se modifican principalmente el dominio de entrenamiento y la continuidad del proceso de aprendizaje.

3.2 Mobility Aids Dataset

Mobility Aids es el conjunto de datos presentado y extendido en el trabajo de Kollmitz et al. (2019). Contiene imágenes RGB-D capturadas con una cámara Kinect 2 montada en una plataforma móvil y fue diseñado para la detección de personas de acuerdo con diferentes ayudas de movilidad [2]. El artículo describe una resolución de 960 x 540 píxeles y cinco clases: pedestrian, person in wheelchair, pedestrian pushing a person in a wheelchair, person using crutches y person using a walking frame [2].

En la implementación utilizada en esta tesis, las etiquetas se representan mediante las clases person, crutches, wheelchair, walking_frame y push_wheelchair. La clase denominada pedestrian aparece en documentación asociada, pero no corresponde a una categoría con instancias en la preparación utilizada para estos experimentos.

3.3 Preparación de Mobility Aids

La documentación disponible del conjunto no permitió reconstruir exactamente la partición original utilizada por los autores en el entrenamiento. Por esta razón, se utilizó el conjunto de test publicado por la fuente y se mantuvo completamente separado del proceso de entrenamiento. El resto de las imágenes RGB se utilizó para construir los subconjuntos de entrenamiento y validación.

Las anotaciones fueron transformadas desde el formato original al formato utilizado por YOLO. Este proceso fue necesario porque el entrenamiento requiere que las cajas y etiquetas estén

codificadas de acuerdo con el formato esperado por el framework. La preparación incluyó además la revisión de la correspondencia entre imágenes y archivos de anotación.

3.4 Partición temporal y distribución de clases 24

Debido a que Mobility Aids contiene frames derivados de secuencias de video, la unidad de partición definitiva fue la secuencia completa y no la imagen individual. El script `creardataset\_v2.py` identificó 115 secuencias a partir de los timestamps presentes en los nombres de archivo, las ordenó temporalmente y seleccionó una frontera de partición a nivel de secuencia. Ninguna secuencia fue dividida entre train y validation. La selección consideró tres condiciones: cobertura de las cinco clases activas en ambas particiones, cercanía al porcentaje objetivo y distribución de clases.

El porcentaje objetivo de validation se fijó inicialmente en 5 %. Sin embargo, ese valor no se impuso sobre imágenes individuales, porque hacerlo habría requerido dividir secuencias y reintroducir dependencia temporal entre particiones. Al trabajar con secuencias completas y exigir cobertura de clases, la solución seleccionada quedó compuesta por 92 secuencias para entrenamiento y 23 para validation. Esto corresponde a 12.437 frames en train (81,40 %) y 2.841 frames en validation (18,60 %). La diferencia respecto del 5 % objetivo es, por tanto, consecuencia de las restricciones metodológicas del procedimiento y no de una partición aleatoria.

Algoritmo resumido de partición temporal: (1) identificar el timestamp de cada frame; (2) agrupar los frames en secuencias cuando la separación temporal entre registros consecutivos pertenece al mismo bloque, utilizando el umbral de 2 s definido en `creardataset\_v2.py`; (3) ordenar las secuencias cronológicamente; (4) evaluar posibles fronteras entre secuencias; (5) descartar fronteras que no mantengan las cinco clases activas en ambas particiones; (6) entre las fronteras válidas, seleccionar la que combine cercanía al porcentaje objetivo y distribución de clases; y (7) auditar la partición resultante mediante solapamiento de secuencias, nombres de imagen y SHA-256, además de registrar la separación temporal.

Tabla 2. Composición de Mobility Aids por partición.

Partición	Secuencias	Imágenes	Objetos	Clases activas
Train	92	12.437	22.308	5
Validation	23	2.841	4.365	5
Test	Publicado; separado	1.801	4.502	5

Fuente: Elaboración propia.

Tabla 3. Distribución de objetos por clase en Mobility Aids.

Clase	Train obj.	Train %	Val obj.	Val %	Test obj.	Test %
person	5.740	25,73 %	432	9,90 %	2.199	48,84 %
crutches	3.955	17,73 %	730	16,72 %	689	15,30 %
wheelchair	4.450	19,95 %	1.666	38,17 %	342	7,60 %
walking_frame	5.268	23,61 %	1.457	33,38 %	924	20,52 %
push_wheelchair	2.895	12,98 %	80	1,83 %	348	7,73 %

Fuente: Elaboración propia.

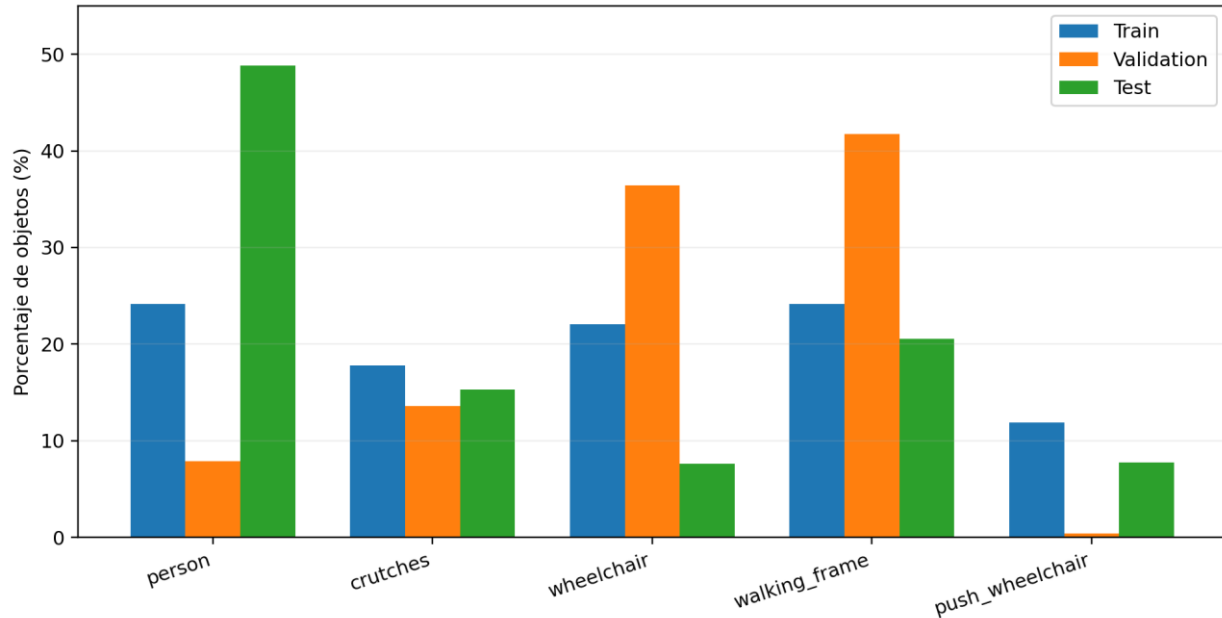


Figura 2. Distribución de objetos por clase en Mobility Aids.

Fuente: Elaboración propia.

3.5 Dataset USM

Dataset USM corresponde al conjunto construido en esta investigación a partir de seis grabaciones realizadas con la participación de 12 actores. Tres grabaciones se realizaron en instalaciones de la Universidad Técnica Federico Santa María y tres en distintos contextos urbanos de Viña del Mar. Las escenas fueron diseñadas para representar situaciones de uso de ayudas técnicas bajo distintas posiciones y condiciones visuales, introduciendo variabilidad de fondo, perspectiva y composición.

Los videos fueron procesados mediante el script ExtractFrames.py. El procedimiento abre cada video con OpenCV, consulta la cantidad total de frames y la frecuencia de cuadros original, y recorre secuencialmente todos los frames sin realizar submuestreo temporal. Cada cuadro se guarda individualmente en formato JPG con calidad 95. El script no realiza redimensionamiento, por lo que la resolución de salida corresponde a la resolución original de cada video. Los archivos se organizan en subdirectorios por video y se nombran mediante el identificador de la grabación y la posición del frame, por ejemplo video13_00000.jpg.

La frecuencia de extracción, por tanto, corresponde a la frecuencia de cuadros nativa de cada archivo de video. El valor numérico de FPS de cada grabación solo debe incorporarse si se encuentra documentado en el registro de ejecución del script.

Los videos fueron divididos en frames y posteriormente anotados. El etiquetado se efectuó mediante Labellmg y se realizó una revisión adicional de las anotaciones. Dataset USM

contempla tres clases: crutches, wheelchair y walking_frame. La partición se realizó por video completo: videos 11, 14, 18 y 19 para entrenamiento; video 15 para validation; y video 13 para test. Por tratarse de videos completos, un mismo video no aparece en más de una partición. Esta separación establece independencia operacional entre secuencias, pero no permite afirmar por sí sola que las personas o escenarios sean completamente distintos entre particiones.

Para hacer verificable la correspondencia entre datasets, Mobility Aids utiliza los identificadores 0: person, 2: crutches, 3: wheelchair, 4: walking_frame y 5: push_wheelchair; el identificador 1 (pedestrian) no presenta instancias en esta preparación. Dataset USM utiliza tres identificadores propios: 0: crutches, 1: wheelchair y 2: walking_frame. Por ello, la coincidencia semántica entre ambos datasets se determina por el nombre de la clase y no por el identificador numérico. Los modelos E1, E2 y E5 poseen una salida de cinco clases, mientras E3 y E4 fueron entrenados con una salida de tres clases. La semántica utilizada para las bounding boxes es: person, caja alrededor de la persona; crutches, caja alrededor de la persona junto con sus muletas; wheelchair, caja alrededor de la silla de ruedas; walking_frame, caja alrededor del andador; y push_wheelchair, caja alrededor de la persona que empuja junto con la silla.

Tabla 4. Composición de Dataset USM por partición.

Partición	Imágenes anotadas	Imágenes sin anotación	Objetos	Clases
Train	3.665	0	6.681	3
Validation	848	1	2.151	3
Test	516	0	1.269	3

Fuente: Elaboración propia.

Tabla 5. Distribución de objetos por clase en Dataset USM.

Clase	Train obj.	Train %	Val obj.	Val %	Test obj.	Test %
crutches	2.106	31,52 %	609	28,31 %	381	30,02 %
wheelchair	3.705	55,46 %	696	32,36 %	372	29,31 %
walking_frame	870	13,02 %	846	39,33 %	516	40,66 %

Fuente: Elaboración propia.

La distribución de Dataset USM cambia entre entrenamiento y evaluación. En entrenamiento, wheelchair concentra más de la mitad de las instancias, mientras validation y test presentan una distribución más próxima a un tercio por clase. Esta característica es relevante porque el modelo aprende a partir de una composición distinta de aquella que enfrenta al momento de la evaluación.

3.6 Configuración experimental

El entrenamiento principal de Mobility Aids utilizó YOLO11s a partir de `yolo11s.pt`. El script `entrenamiento\_final.py` fue diseñado para aproximar el esquema de optimización descrito en

TRABAJO DE TITULO

Kollmitz et al. (2019), adaptándolo a YOLO11. La corrida corregida de E1 utilizó la nueva partición temporal de Mobility Aids y generó un `best.pt` que posteriormente se utilizó como punto de partida del E2. En los experimentos posteriores se documentaron las configuraciones efectivamente ejecutadas mediante los parámetros registrados por el framework, incluyendo diferencias de semilla y learning rate entre protocolos.

Tabla 6. Configuración experimental principal de YOLO11s.

Parámetro	Configuración E1	Justificación
Modelo	YOLO11s	Arquitectura seleccionada posteriormente para experimentos entre dominios.
Optimizador	SGD	Mantener el esquema general del antecedente.
Learning rate inicial	0,0025	Valor utilizado en el script inspirado en el paper.
Momentum	0,9	Valor utilizado en el paper.
Weight decay	0,0001	Valor utilizado en el paper.
Iteraciones objetivo	60.000	Aproximación del número de iteraciones del entrenamiento del antecedente.
Reducciones de LR	30.000 y 40.000	Factor x0,1 en cada punto.
Warm-up	500 iteraciones	Esquema lineal implementado mediante callback.
Batch	16	Ajustado a la ejecución de YOLO11 y al hardware disponible.
Imagen	640 px	Resolución de entrada utilizada en YOLO11.
Workers	4	Configuración del DataLoader.
Seed	42	Reproducibilidad del entrenamiento.

Fuente: Elaboración propia a partir del script de entrenamiento y del antecedente [2].

El script utiliza un scheduler personalizado que modifica el learning rate por iteración. Durante las primeras 500 iteraciones se realiza un warm-up desde un tercio del learning rate base hasta su valor completo. Posteriormente se mantiene el valor base hasta 30.000 iteraciones, se reduce por un factor de 0,1 hasta 40.000 y se aplica una segunda reducción por un factor adicional de 0,1 hasta completar el objetivo de 60.000 iteraciones. El procedimiento se utiliza para aproximar el esquema temporal del antecedente dentro de YOLO11.

3.7 Diseño de los cinco experimentos

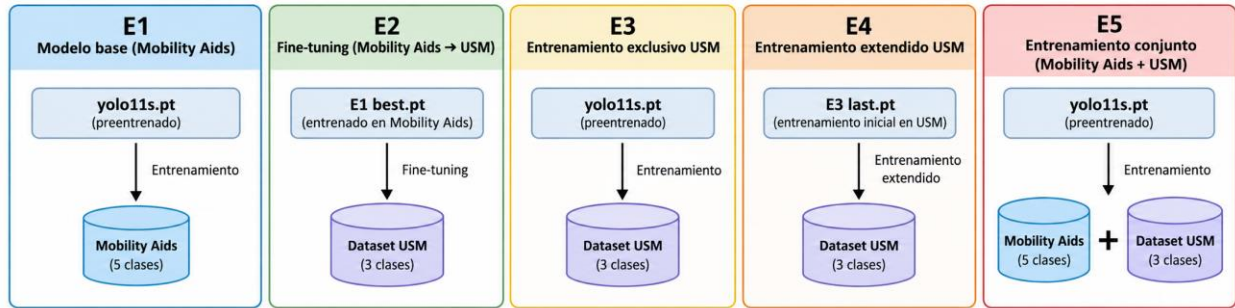


Figura 3. Flujo de los cinco experimentos con YOLO11s.

Fuente: Elaboración propia.

Tabla 7. Resumen de los cinco experimentos.

Experimento	Modelo de partida	Entrenamiento principal	Continuación / adaptación	Evaluación
E1	yolo11s.pt	Mobility Aids	No	Mobility Aids test + USM test
E2	best.pt de E1 corregido	USM	Sí; fine-tuning	Mobility Aids test + USM test
E3	yolo11s.pt	USM	No	USM test + evaluación diagnóstica en Mobility Aids
E4	last.pt de E3	USM	Sí; más épocas	USM test + evaluación diagnóstica en Mobility Aids
E5	yolo11s.pt	Mobility Aids + USM	No; entrenamiento conjunto	Mobility Aids test + USM test

Fuente: Elaboración propia.

En E2 se utilizó el `best.pt` generado por el nuevo entrenamiento de Mobility Aids como modelo de partida y se continuó el aprendizaje exclusivamente con Dataset USM. El script `fine-tuning.py` utiliza 50 épocas, batch 16, imágenes de 640 píxeles, SGD, learning rate 0,0005, momentum 0,9, weight decay 0,0001, warm-up de 1 época, scheduler coseno, mosaic 0,5, mixup 0 y label smoothing 0,05. La nomenclatura de una carpeta utilizada durante alguna ejecución podía conservar nombres anteriores; para esta versión de la tesis se identifica el origen del modelo por el checkpoint efectivo: el `best.pt` del E1 corregido.

E5 utiliza una estrategia distinta. El modelo se inicializa directamente con `yolo11s.pt` y se entrena sobre un conjunto combinado formado por Mobility Aids y Dataset USM. El train combinado contiene 16.102 archivos, de los cuales 16.075 poseen anotaciones y 27 no poseen anotaciones, con 28.989 objetos; validation contiene 3.690 archivos, de los cuales 3.689 poseen anotaciones y uno no posee anotaciones, con 6.516 objetos. Esta configuración no posee un test propio: las evaluaciones se realizan posteriormente sobre los dos conjuntos de test de dominio, que permanecen separados. De este modo, E5 representa entrenamiento conjunto desde pesos preentrenados y no una continuación desde el `best.pt` de Mobility Aids.

3.8 Evaluación y criterio de selección final

Los resultados se evaluaron mediante precision, recall, mAP50 y mAP50-95. Para las comparaciones entre experimentos se utilizaron los mismos conjuntos de test de cada dominio: Mobility Aids test, con 1.801 imágenes y 4.502 instancias, y USM test, con 516 imágenes y 1.269 instancias. Ninguna imagen de estos dos conjuntos fue incorporada al entrenamiento. Los resultados de test sí fueron observados durante el desarrollo y se utilizaron para comparar E1-E5 y para apoyar la selección de la arquitectura YOLO11s y de la configuración de referencia. Por ello, la comparación es consistente como benchmark interno bajo muestras constantes, pero el test no constituye una estimación completamente independiente del proceso de selección.

Para evaluar el equilibrio entre dominios se definió un criterio auxiliar propio. Debido a que Mobility Aids utiliza cinco clases activas y Dataset USM solo tres, el promedio interdominio se calcula únicamente sobre las tres clases compartidas: crutches, wheelchair y walking_frame. Primero se calcula el AP de cada clase en cada dominio y luego el promedio de esas tres clases. El valor de cada dominio así obtenido se utiliza para calcular el promedio entre dominios y el mínimo entre ambos. Esta decisión evita atribuir a una diferencia en el número de clases un efecto que en realidad proviene de la estructura de las etiquetas.

3.9 Reproducibilidad y gestión experimental

La reproducibilidad se documentó mediante los scripts utilizados, los archivos YAML de los datasets, los registros `args.yaml`, `results.csv` y los archivos de trazabilidad de la partición temporal. Para Mobility Aids, el script `creardataset\_v2.py` genera `split\_manifest.csv`, `videos\_summary.csv`, `class\_distribution.csv`, `audit\_report.json`, `overlap\_report.txt`, `run\_config.json` y `data.yaml`. Para USM, la partición por video se conserva en su estructura original y se documentan las rutas y clases mediante su YAML correspondiente.

Para cada experimento, la trazabilidad se establece mediante el modelo de partida, dataset utilizado, checkpoint evaluado, configuración de entrenamiento, cantidad de épocas, parámetros de optimización y archivos de resultados. La ejecución registrada utilizó Ultralytics 8.3.191 y una NVIDIA GeForce RTX 4060; los tiempos de validación y prueba reportados por el framework se descomponen en preprocesamiento, inferencia y postprocesamiento. Los checkpoints `best.pt` constituyen los modelos utilizados para las evaluaciones finales mostradas en este documento.

La comparación entre experimentos utiliza los mismos conjuntos de test de cada dominio. Esta condición permite comparar E1-E5 bajo muestras constantes de evaluación. No obstante, como los resultados de test participaron en decisiones de selección y comparación, no se presentan como una estimación completamente independiente de esas decisiones. En todos los casos, los conjuntos de test se mantuvieron físicamente separados de los datos de entrenamiento.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN

4.1 Consideraciones generales

Este capítulo presenta los resultados de los experimentos y su interpretación. Las métricas precision, recall, mAP50 y mAP50-95 provienen de las ejecuciones del framework y F1 se calcula a partir de precision y recall cuando no fue entregado directamente. Los conjuntos de test permanecen fuera del entrenamiento. Para las comparaciones entre experimentos se utilizan los mismos dos conjuntos de test, aunque sus resultados fueron observados durante el desarrollo y la selección, situación que se reconoce como una limitación.

4.2 Validación de la partición temporal final

Tabla 8. Auditoría de la partición temporal final de Mobility Aids.

Indicador	Resultado	Indicador	Resultado	Indicador	Resultado
Secuencias totales	115	Train / validation	92 / 23	Frames train / validation	12.437 / 2.841
Proporción	81,40 % / 18,60 %	Solapamiento de videos	0	Solapamiento de basenames	0
Duplicados SHA-256	0	Separación temporal	44,5204 s	Clases ausentes	Ninguna
Estado auditoría	OK	Unidad de partición	Secuencia completa	Test publicado	Separado

Fuente: Elaboración propia.

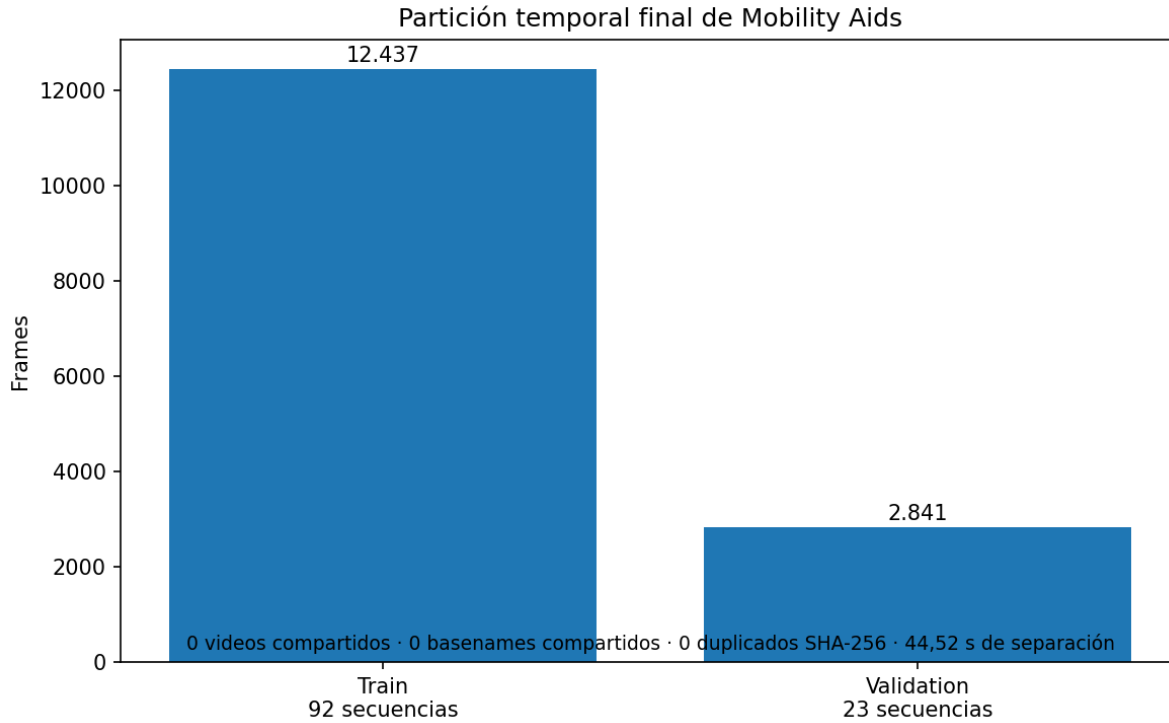


Figura 4. Partición temporal final de Mobility Aids y criterios de auditoría.

Fuente: Elaboración propia.

La partición definitiva no se basa en una selección independiente de frames. Se construyó a nivel de secuencia para evitar que imágenes temporalmente próximas quedaran distribuidas entre train y validation. La auditoría final identificó 115 secuencias, de las cuales 92 fueron asignadas a train y 23 a validation. El conjunto de 12.437 frames de train y 2.841 frames de validation representa 81,40 % y 18,60 %, respectivamente. La cobertura de las cinco clases activas se verifica en ambas particiones.

La auditoría automática verificó cero secuencias compartidas, cero nombres de imagen compartidos y cero duplicados de imagen por SHA-256 entre train y validation. Además, registró una separación temporal de 44,52 s entre el último timestamp de train y el primero de validation. Estos controles proporcionan evidencia de separación operacional entre las particiones de aprendizaje y reducen de manera explícita el riesgo de que frames de una misma secuencia o copias idénticas sean distribuidos entre train y validation. No se interpreta esta evidencia como una prueba de independencia estadística total ni como demostración de que las personas o escenarios sean necesariamente nuevos en cada partición.

4.3 Comparación entre YOLO11n, YOLO11s y YOLO11m

Tabla 9. Complejidad y desempeño del benchmark de variantes YOLO11.

Modelo	Parámetros	GFLOPs	P test	R test	mAP50 test	mAP50-95 test
YOLO11n	2.583.322	6,3	0,900	0,789	0,860	0,609
YOLO11s	9.415.122	21,3	0,909	0,804	0,869	0,628
YOLO11m	20.034.658	67,7	0,924	0,787	0,870	0,631

TRABAJO DE TITULO

Fuente: Elaboración propia.

Tabla 10. AP@50-95 por clase en test de Mobility Aids.

Clase	YOLO11n	YOLO11s	YOLO11m
person	0,503	0,517	0,523
crutches	0,760	0,765	0,781
wheelchair	0,293	0,339	0,341
walking_frame	0,672	0,716	0,691
push_wheelchair	0,814	0,801	0,821

Fuente: Elaboración propia.

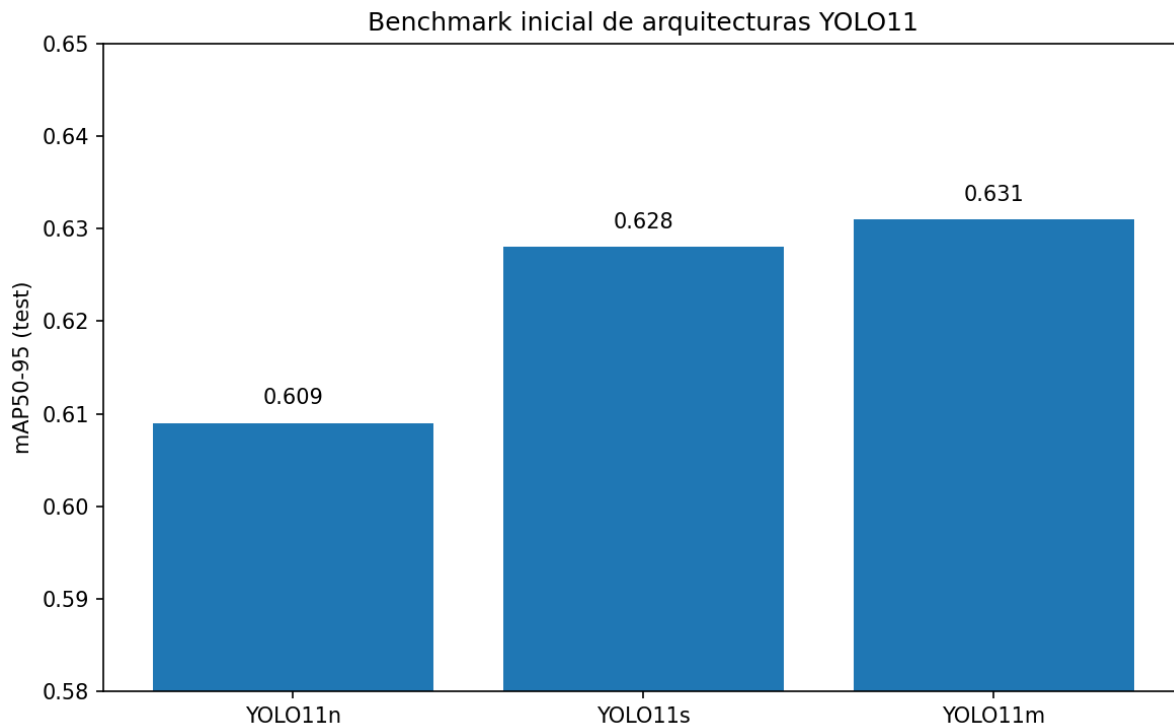


Figura 5. Comparación de mAP50-95 entre YOLO11n, YOLO11s y YOLO11m.

Fuente: Elaboración propia.

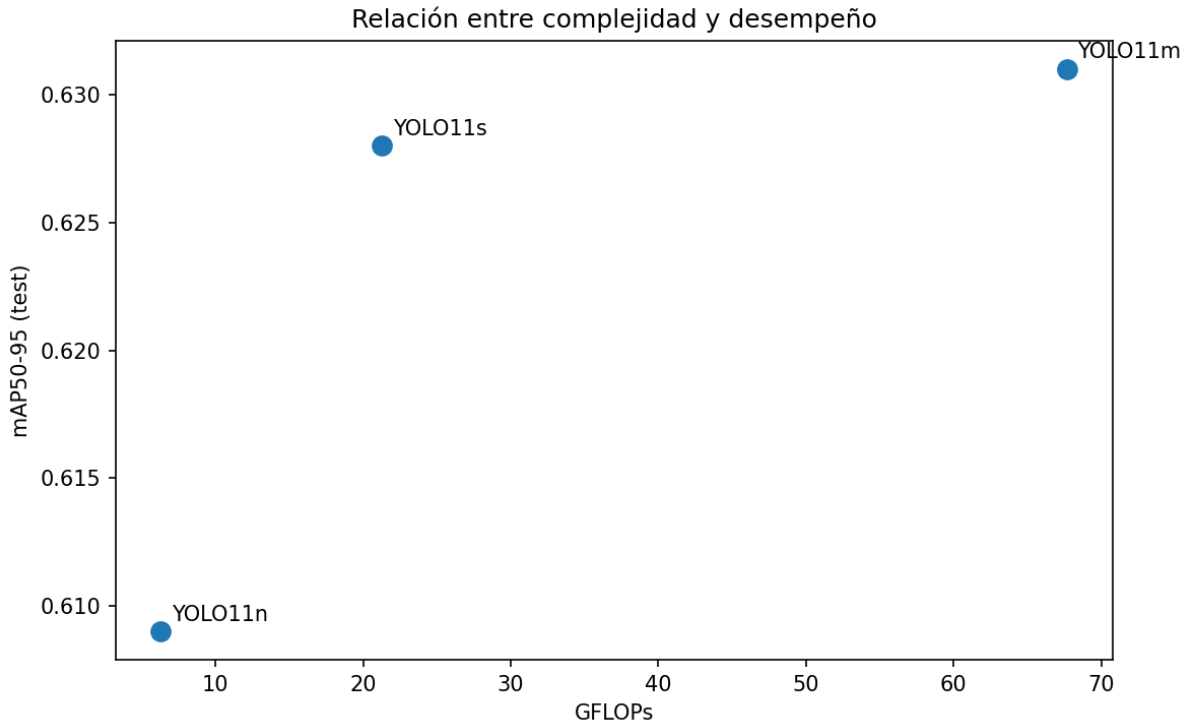


Figura 6. Relación entre complejidad computacional y mAP50-95.

Fuente: Elaboración propia.

En el benchmark inicial de arquitectura, YOLO11n alcanzó 0,609 de mAP50-95, YOLO11s 0,628 y YOLO11m 0,631. Al pasar de n a s, los GFLOPs aumentaron de 6,3 a 21,3 y el mAP50-95 aumentó 0,019. Al pasar de s a m, los GFLOPs aumentaron de 21,3 a 67,7, mientras el mAP50-95 aumentó solo 0,003. Esta comparación corresponde a la selección de arquitectura realizada sobre la partición temporal corregida de Mobility Aids. La misma corrida se utilizó como referencia para el compromiso entre desempeño y complejidad.

YOLO11m presentó la mayor precisión, con 0,924, mientras YOLO11s obtuvo un recall superior, 0,804 frente a 0,787. A nivel de clases, wheelchair permaneció como la categoría más difícil. La diferencia marginal entre s y m y el incremento sustancial de complejidad motivaron la selección de YOLO11s. Los resultados de esta etapa constituyen la base de la selección de arquitectura y se mantienen conceptualmente separados de los resultados de adaptación de E1-E5.

Justificación técnica. Incrementar el tamaño de la red no resolvió por sí mismo la dificultad de las clases más complejas. Aunque YOLO11m alcanzó el máximo mAP50-95 del benchmark, YOLO11s proporcionó un desempeño muy cercano con menos GFLOPs y fue la arquitectura utilizada de manera fija en las etapas posteriores. La selección debe entenderse como un compromiso de diseño, no como identificación del máximo valor aislado.

4.4 Selección de YOLO11s para los experimentos posteriores

YOLO11s alcanzó 0,628 de mAP50-95 en el benchmark de Mobility Aids con la partición temporal corregida, frente a 0,631 de YOLO11m. La diferencia fue 0,003, mientras los GFLOPs

fueron 21,3 y 67,7, respectivamente. Además, YOLO11s presentó mayor recall. La arquitectura se mantuvo fija para los experimentos posteriores.

La selección de YOLO11s se realizó utilizando el resultado del conjunto de test junto con los indicadores de complejidad computacional. En consecuencia, esta decisión forma parte del proceso de desarrollo experimental y el test no debe considerarse completamente independiente de la selección. La ventaja metodológica de la decisión radica en su trazabilidad explícita: se buscó una arquitectura que equilibrara desempeño y costo, y no simplemente la variante con el mayor mAP50-95.

4.5 Modelo base YOLO11s y cambio de dominio

Tabla 11. Resultados globales del modelo base YOLO11s.

Modelo	Fase / dominio	Precision	Recall	F1*	mAP50	mAP50-95
YOLO11s	Validation Mobility Aids (E1 corregido)	0,921	0,918	0,919	0,959	0,781
YOLO11s	Test Mobility Aids (E1 corregido)	0,934	0,805	0,865	0,871	0,625
YOLO11s	Test USM (E1 corregido)	0,759	0,731	0,745	0,841	0,534

Fuente: Elaboración propia.

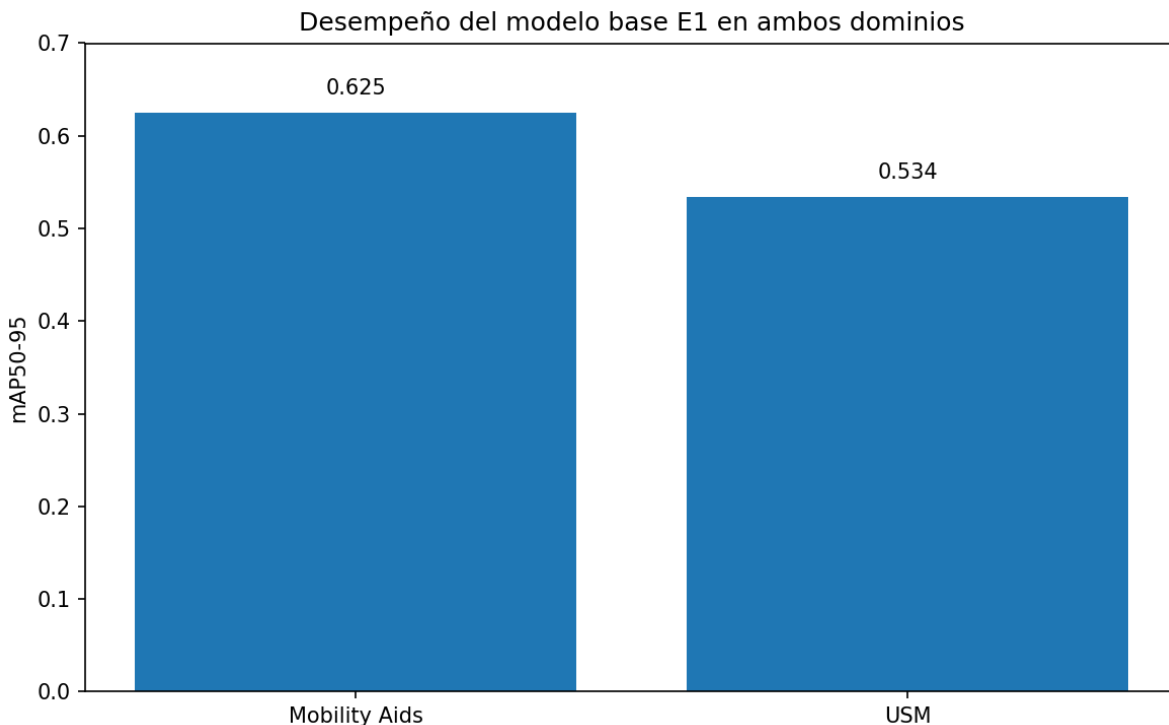


Figura 7. Desempeño del modelo base E1 en ambos dominios.

Fuente: Elaboración propia.

Con la partición temporal corregida, el nuevo modelo base E1 obtuvo 0,625 de mAP50-95 en Mobility Aids y 0,534 en USM. La diferencia absoluta es de 0,091 puntos. El modelo mantiene mejor desempeño en el dominio con el que fue entrenado, aunque conserva capacidad de detección en el segundo dominio. La evaluación cruzada confirma que la generalización depende de las características visuales del dominio de evaluación.

Justificación técnica. El cambio de dominio no se refleja necesariamente de la misma forma en todas las métricas. Por ello, la interpretación utiliza conjuntamente precisión, recall y mAP, y se complementa con el análisis por clase. Esta evaluación constituye la referencia para estudiar posteriormente qué ocurre cuando se modifica la estrategia de entrenamiento.

4.6 Experimentos de adaptación y entrenamiento

Se realizaron cinco configuraciones con YOLO11s. E1 corresponde al modelo base entrenado con Mobility Aids. E2 continúa el aprendizaje desde el `best.pt` del nuevo E1 y utiliza exclusivamente USM. E3 entrena YOLO11s desde `yolo11s.pt` utilizando únicamente USM. E4 continúa el entrenamiento de E3 sobre USM durante un máximo del doble de épocas. E5 inicializa directamente `yolo11s.pt` y entrena conjuntamente con datos de Mobility Aids y USM. Para Mobility Aids, E3 y E4 poseen un head de tres clases correspondiente a USM, por lo que sus evaluaciones sobre el test de Mobility Aids no se utilizan como comparación semánticamente equivalente con E1, E2 y E5.

El objetivo de esta etapa es distinguir tres fenómenos: adaptación secuencial hacia USM, especialización mediante entrenamiento exclusivo en USM y aprendizaje conjunto desde pesos preentrenados. La evaluación de USM permite comparar las cinco configuraciones dentro del mismo esquema de clases. La comparación interdominio de Mobility Aids se realiza directamente solo para los modelos cuyo espacio semántico es compatible con las cinco clases activas de ese dataset.

4.6.1 Resultados sobre Mobility Aids

Tabla 12. Resultados globales de las configuraciones semánticamente comparables en test de Mobility Aids.

Exp.	P	R	F1*	mAP50	mAP50-95	Delta vs E1
E1	0,934	0,805	0,865	0,871	0,625	0,000
E2	0,536	0,756	0,627	0,758	0,549	-0,076
E3	N/C	N/C	N/C	N/C	N/C	N/C
E4	N/C	N/C	N/C	N/C	N/C	N/C
E5	0,945	0,808	0,871	0,880	0,633	+0,008

TRABAJO DE TITULO

Fuente: Elaboración propia.

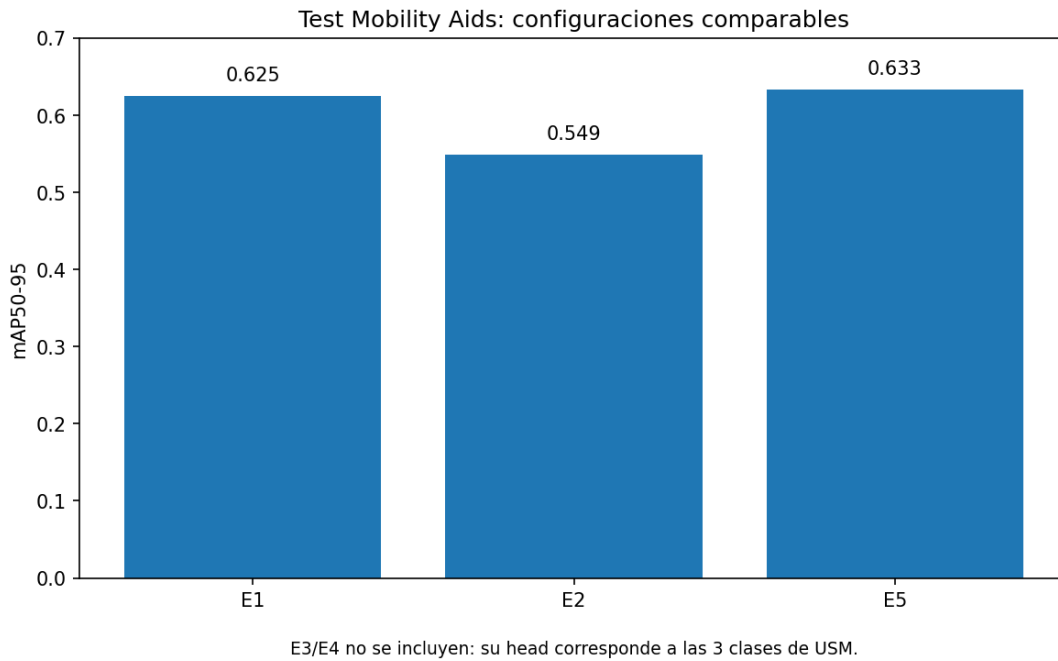


Figura 8. mAP50-95 de configuraciones comparables en Mobility Aids. 36

Fuente: Elaboración propia.

En el test de Mobility Aids, el nuevo E1 obtiene 0,625 de mAP50-95, E2 0,549 y E5 0,633. E3 y E4 no se incorporan a esta comparación global como si fueran modelos de cinco clases, porque fueron entrenados con un head de tres clases correspondiente a USM. Las evaluaciones históricas de esos modelos sobre Mobility Aids produjeron 0,0818 para E3 y 0,145 para E4, pero esos valores no se utilizan para establecer comparaciones semánticas directas. Para una comparación interdominio estricta de E3 y E4 sería necesario reevaluar el test de Mobility Aids con un YAML reducido a las tres clases comunes y verificar el remapeo de sus identificadores hacia el head de tres clases de USM.

Entre las configuraciones semánticamente comparables, E5 supera ligeramente a E1 en Mobility Aids, 0,633 frente a 0,625, mientras E2 presenta una reducción hasta 0,549 después de la adaptación hacia USM.

Justificación técnica. El patrón de E2 es consistente con una especialización hacia el dominio objetivo, acompañada de pérdida sobre el dominio de origen. E5, en cambio, incorpora datos de ambos dominios desde una inicialización común y consigue mantener un rendimiento alto en Mobility Aids y USM simultáneamente.

4.6.2 Resultados sobre USM

Tabla 13. Resultados globales en test de USM.

Exp.	P	R	F1*	mAP50	mAP50-95	Delta vs E1
E1	0,759	0,731	0,745	0,841	0,534	0,000
E2	0,949	0,946	0,947	0,994	0,852	+0,318
E3	0,509	0,957	0,665	0,939	0,757	+0,223
E4	0,894	0,742	0,811	0,926	0,755	+0,221

TRABAJO DE TITULO

Exp.	P	R	F1*	mAP50	mAP50-95	Delta vs E1
E5	0,996	0,994	0,995	0,995	0,851	+0,317

Fuente: Elaboración propia.

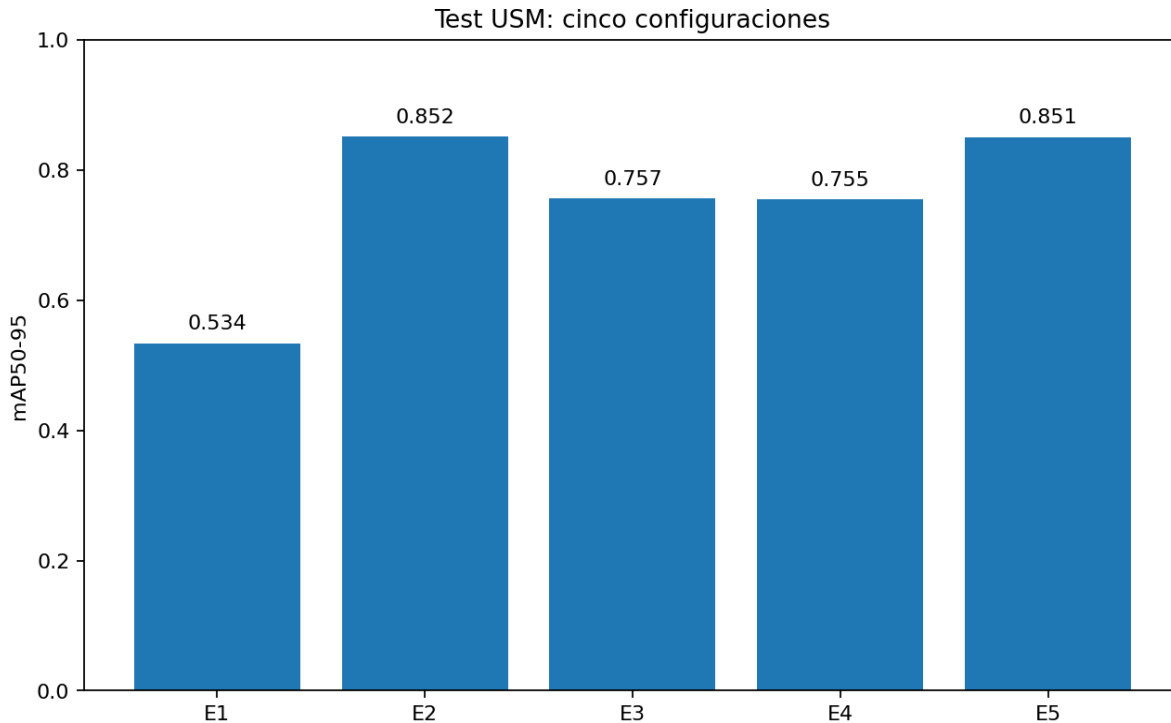


Figura 9. mAP50-95 de los cinco experimentos en USM.

Fuente: Elaboración propia.

E2 obtiene el mayor mAP50-95 en USM, con 0,852, junto con precisión 0,949 y recall 0,946. E3 alcanza 0,757 y E4 0,755. E5 obtiene 0,851, acompañado de precisión 0,996 y recall 0,994. E3 y E4 muestran que prolongar el entrenamiento sobre USM no produjo una mejora del mAP50-95 dentro del dominio.

El cambio de E3 a E4 permite observar un cambio importante en el equilibrio entre precisión y recall, mientras el mAP50-95 permanece prácticamente estable. E3 presenta precisión 0,509 y recall 0,957; E4 presenta precisión 0,894 y recall 0,742. El resultado sugiere que el entrenamiento adicional modifica el comportamiento del detector sin producir una ganancia en la métrica principal de precisión promedio sobre IoU.

Justificación técnica. El entrenamiento adicional sobre el mismo dominio no garantiza una mejora sistemática. E2 ofrece la mayor especialización en USM, mientras E5 presenta un resultado inferior al máximo de E2 pero superior al de E1 y con mayor equilibrio entre los dos dominios comparables.

4.6.3 Análisis por clase

Tabla 14. AP@50-95 por clase en Mobility Aids para E1, E2 y E5.

Clase	E1	E2	E3	E4	E5
-------	----	----	----	----	----

TRABAJO DE TITULO

Clase	E1	E2	E3	E4	E5
person	0,523	0,446	N/A	N/A	0,528
crutches	0,761	0,715	N/A	N/A	0,767
wheelchair	0,313	0,161	N/A	N/A	0,333
walking_frame	0,699	0,715	N/A	N/A	0,709
push_wheelchair	0,830	0,710	N/A	N/A	0,830

Fuente: Elaboración propia.

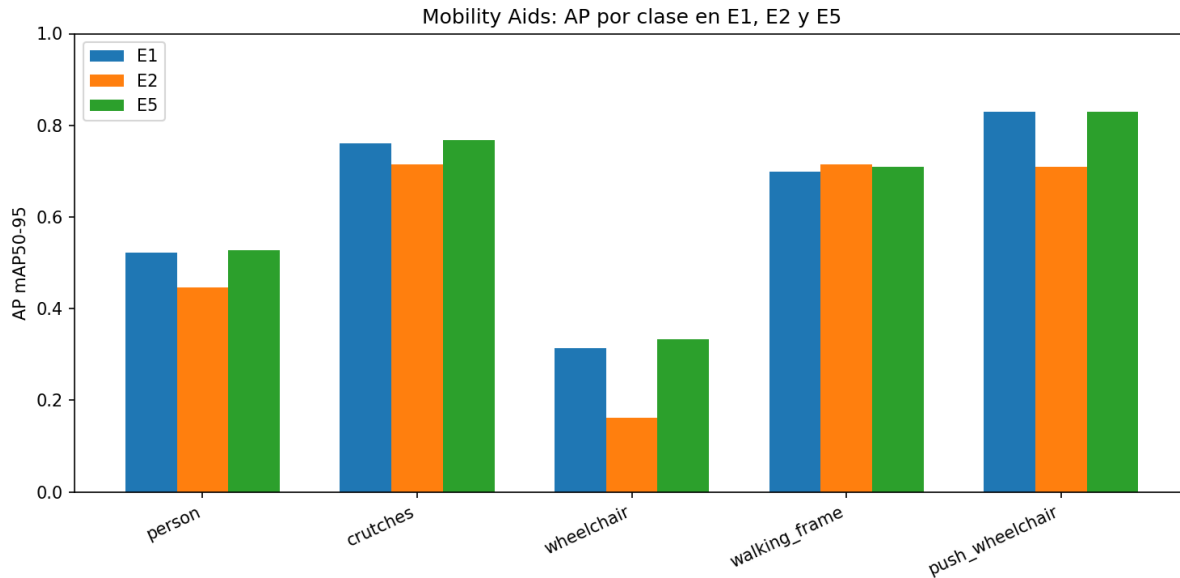


Figura 10. AP@50-95 por clase en Mobility Aids.

Fuente: Elaboración propia.

En Mobility Aids, el análisis por clase se restringe a los modelos cuyo espacio de clases coincide con las cinco categorías activas del dataset: E1, E2 y E5. E5 alcanza 0,767 en crutches, 0,333 en wheelchair, 0,709 en walking_frame y 0,830 en push_wheelchair, mientras E1 alcanza 0,761, 0,313, 0,699 y 0,830, respectivamente, y E2 obtiene 0,715, 0,161, 0,715 y 0,710. La clase wheelchair continúa siendo la más difícil.

E5 mejora respecto de E1 en crutches, wheelchair y walking_frame, aunque el margen es pequeño en las dos primeras y no uniforme en todas las clases. En person, E5 alcanza 0,528 frente a 0,523 de E1. Las diferencias deben interpretarse como resultados observados en una sola corrida, sin repeticiones de semillas.

Tabla 15. AP@50-95 por clase en USM.

Clase	E1	E2	E3	E4	E5
crutches	0,389	0,795	0,665	0,692	0,788
wheelchair	0,599	0,886	0,821	0,881	0,852
walking_frame	0,612	0,876	0,785	0,692	0,851

TRABAJO DE TITULO

Fuente: Elaboración propia. Nota: E3 y E4 se excluyen del análisis de Mobility Aids por incompatibilidad entre sus tres clases de salida y el espacio de clases de Mobility Aids; sus evaluaciones cruzadas requieren un remapeo semántico para ser comparables.

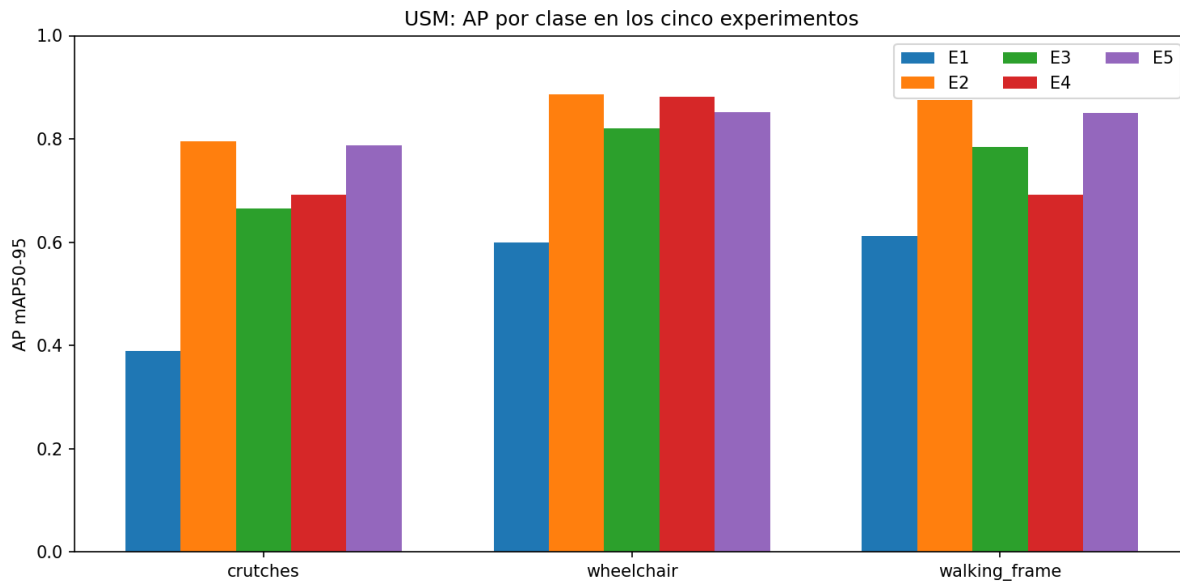


Figura 11. AP@50-95 por clase en USM.

Fuente: Elaboración propia.

En USM, E2 presenta 0,795 para crutches, 0,886 para wheelchair y 0,876 para walking_frame. E3 alcanza 0,665, 0,821 y 0,785; E4 obtiene 0,692, 0,881 y 0,692; y E5 alcanza 0,788, 0,852 y 0,851. La comparación muestra que E2 es la mejor estrategia específica para USM bajo mAP50-95, mientras E5 conserva un desempeño elevado sin especialización secuencial.

Justificación técnica. El análisis por clase demuestra que una métrica agregada puede ocultar dificultades específicas. wheelchair presenta una dificultad persistente en Mobility Aids, mientras que en USM es una de las clases con mejor desempeño. Esta diferencia refuerza la necesidad de interpretar los resultados junto con la distribución de clases, el contexto de captura y la semántica exacta de las etiquetas.

4.6.4 Análisis visual de errores

Las matrices de confusión y las curvas precision-recall complementan las métricas agregadas y permiten identificar patrones de error que no quedan reflejados completamente en el mAP global. En Mobility Aids, la clase wheelchair presenta las mayores dificultades de detección y localización, coherentes con sus valores inferiores de AP50-95 en E1, E2 y E5. En USM, el comportamiento de wheelchair es considerablemente mejor, mientras que crutches y walking_frame presentan variaciones según la estrategia de entrenamiento. Las matrices de confusión muestran además errores de asignación entre ayudas técnicas y predicciones clasificadas como background. Estos resultados justifican el análisis conjunto de métricas globales, métricas por clase y evidencia visual de verdaderos positivos, falsos positivos, falsos negativos y errores de localización. Las imágenes de predicción incluidas en los anexos deben seleccionarse para representar estos casos y vincularlos explícitamente con las dificultades observadas en las métricas.

4.7 Fine-tuning Mobility Aids -> USM

Tabla 16. Comparación directa de las principales estrategias.

Configuración	Dominio	P	R	F1*	mAP50-95
E1 base	USM	0,759	0,731	0,745	0,534
E2 fine-tuning	USM	0,949	0,946	0,947	0,852
E1 base	Mobility Aids	0,934	0,805	0,865	0,625
E2 fine-tuning	Mobility Aids	0,536	0,756	0,627	0,549
E5 conjunto	USM	0,996	0,994	0,995	0,851
E5 conjunto	Mobility Aids	0,945	0,808	0,871	0,633

Fuente: Elaboración propia.

El fine-tuning E2 aumenta el mAP50-95 del test USM desde 0,534 en E1 hasta 0,852, una diferencia absoluta de 0,318 puntos. Al mismo tiempo, Mobility Aids disminuye desde 0,625 hasta 0,549. El patrón es consistente con especialización hacia el dominio USM. No obstante, E2 modifica simultáneamente el dominio de entrenamiento y varios hiperparámetros, por lo que no es posible atribuir la mejora a una única variable de manera causal.

El entrenamiento conjunto E5 produce 0,851 aproximadamente en USM y 0,633 en Mobility Aids. Para evitar la diferencia de cardinalidad entre datasets, el análisis de equilibrio utiliza las tres clases compartidas y no las medias directas de las métricas globales de cinco y tres clases.

Justificación técnica. E2 maximiza el resultado específico sobre USM, mientras E5 presenta un compromiso favorable cuando la meta es mantener simultáneamente el rendimiento en ambos dominios. La comparación se limita al protocolo estudiado y no permite afirmar estabilidad estadística o superioridad universal.

4.8 Comparación con Kollnitz et al. (2019)

El trabajo de Kollnitz et al. reporta un mAP de 0,87 para RGB y 53 ms por imagen bajo su protocolo [2]. La presente investigación, con el nuevo E1 sobre Mobility Aids, obtiene 0,871 de

TRABAJO DE TITULO

mAP50 y 0,625 de mAP50-95, con precisión 0,934 y recall 0,805. Las métricas y protocolos no son idénticos, por lo que la comparación se mantiene como referencia contextual.

Tabla 17. Comparación descriptiva con Kollnitz et al. (2019).

Trabajo	Modelo	Protocolo / datos	Precisión	Recall	mAP reportado	mAP50-95	Tiempo
Kollnitz et al. (2019)	Faster R-CNN + estimación 3D + seguimiento	RGB/depth; umbral de distancia 0,5 m	-	-	0,870 RGB*	-	53 ms/imagen
Presente investigación	YOLO11s	Mobility Aids; test temporal corregido	0,934	0,805	0,871 mAP50**	0,625	≈3,0 ms/imagen

Fuente: Elaboración propia a partir de [2] y de las ejecuciones experimentales. * El mAP del antecedente incorpora un criterio de distancia; ** el valor 0,885 corresponde a mAP50 de detección 2D y no es equivalente al mAP del antecedente.

La diferencia numérica entre el mAP50 de E1 y el mAP RGB reportado por Kollnitz et al. es pequeña, pero ambas cifras no representan exactamente la misma métrica ni el mismo protocolo. El antecedente incorpora profundidad y componentes temporales, mientras esta tesis se concentra en detección 2D. Por ello, la comparación no constituye una demostración de superioridad entre sistemas.

La latencia de la ejecución actual, aproximadamente 3 ms por imagen en la GPU RTX 4060, tampoco constituye una medición de aceleración controlada respecto de los 53 ms reportados por el antecedente. Las plataformas, versiones de software y componentes del pipeline son diferentes. El valor debe interpretarse como una medición de esta implementación específica.

Justificación técnica. La comparación cumple una función de contextualización histórica y metodológica. Permite observar que una arquitectura moderna de una etapa puede alcanzar un mAP50 numéricamente cercano al mAP reportado en el antecedente sobre el mismo universo general de datos, pero las diferencias de protocolo impiden concluir que una arquitectura sea globalmente superior a la otra.

4.9 Discusión integrada

Tabla 18. Criterio de generalización equilibrada entre ambos dominios y clases comunes.

Exp.	Mobility Aids común	USM común	Promedio	Mínimo	Brecha
E1	0,591	0,533	0,562	0,533	0,058
E2	0,530	0,852	0,691	0,530	0,322
E3	N/C	0,757	N/C	N/C	N/C
E4	N/C	0,755	N/C	N/C	N/C
E5	0,603	0,830	0,717	0,603	0,227

Fuente: Elaboración propia.

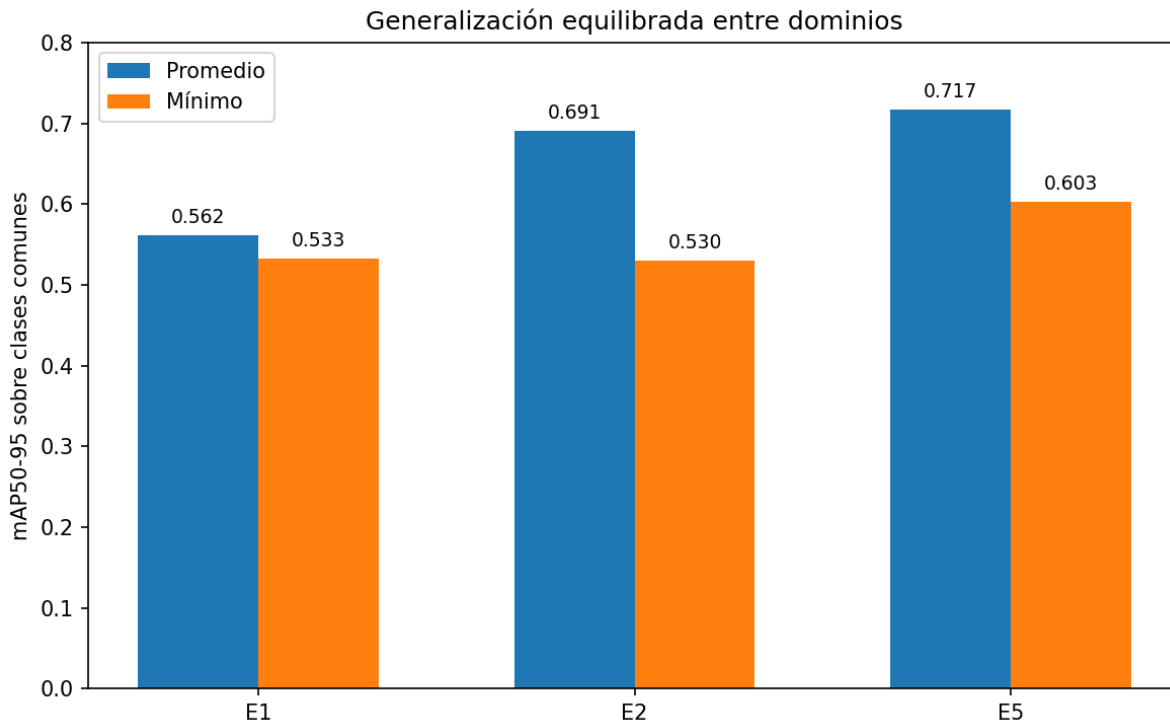


Figura 12. Promedio y mínimo de mAP50-95 entre ambos dominios.

Fuente: Elaboración propia.

Para comparar el comportamiento interdominio se utilizan solo las tres clases compartidas por Mobility Aids y USM. Con los valores de AP por clase disponibles, E1 presenta un mAP común de aproximadamente 0,591 en Mobility Aids y 0,533 en USM, con un promedio entre dominios de aproximadamente 0,562 y un mínimo de 0,533. E2 presenta aproximadamente 0,530 y 0,852, con un promedio de 0,691 y un mínimo de 0,530. E5 alcanza aproximadamente 0,603 y 0,830, con un promedio de 0,717 y un mínimo de 0,603. E5 presenta, por tanto, el mayor promedio y el mayor mínimo entre las configuraciones directamente comparables bajo este criterio.

Este criterio evita que la diferencia de número de clases entre los datasets determine artificialmente el resultado. E1 mantiene una brecha menor entre dominios, pero sus niveles absolutos son inferiores a los de E5. E2 maximiza USM, pero su desempeño en las clases comunes de Mobility Aids disminuye. E5 ofrece el compromiso más favorable observado bajo la definición adoptada.

Justificación técnica. La evidencia integrada respalda que la composición del conjunto de entrenamiento es un factor relevante para el comportamiento entre dominios. La adaptación secuencial favorece la especialización, mientras el entrenamiento conjunto permite mantener simultáneamente un desempeño elevado en ambos dominios. La conclusión se limita a las dos colecciones evaluadas y a una sola corrida por configuración.

TRABAJO DE TITULO

Tabla 19. Resumen metodológico de resultados corregidos.

Etapas	Configuración relevante	Resultado principal	Interpretación metodológica
Partición	Temporal por secuencias	115 secuencias; 92 train / 23 val; 44,52 s de separación	Control explícito de la dependencia temporal y auditoría de solapamientos.
Arquitectura	YOLO11m vs YOLO11s	0,631 vs 0,628; 67,7 vs 21,3 GFLOPs	YOLO11s seleccionado como compromiso entre desempeño y complejidad; test utilizado en la decisión.
Modelo base	E1 YOLO11s	0,625 Mobility; 0,534 USM	Referencia corregida con la nueva partición temporal.
Adaptación específica	E2 fine-tuning -> USM	0,549 Mobility; 0,852 USM	Especialización hacia USM con pérdida en el dominio de origen.
Entrenamiento conjunto	E5 desde yolo11s.pt	0,633 Mobility; 0,851 USM; criterio común 0,717/0,603	Mejor compromiso observado entre las configuraciones directamente comparables.

Fuente: Elaboración propia.

CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

5.1 Síntesis de la investigación

La investigación tuvo como objetivo desarrollar y validar un sistema de detección automática de personas con movilidad reducida mediante técnicas de visión por computador y aprendizaje profundo. El estudio se estructuró en torno a YOLO11 y utilizó Mobility Aids como dominio inicial y Dataset USM como segundo dominio experimental. La metodología permitió analizar la influencia de la partición, la capacidad de la arquitectura, el cambio de dominio y las estrategias de adaptación.

La primera etapa mostró que el diseño de la partición debe considerar la naturaleza secuencial de Mobility Aids. La partición definitiva por secuencias produjo 12.437 frames de entrenamiento y 2.841 de validation, sin solapamiento de secuencias, nombres ni hashes entre ambas particiones y con una separación temporal de 44,52 s. Estos controles permiten documentar la separación operacional del aprendizaje y reducir el riesgo de dependencia temporal directa.

La segunda etapa correspondió al benchmark de arquitectura realizado sobre la partición temporal corregida. YOLO11m obtuvo 0,631 de mAP50-95 y YOLO11s 0,628. La diferencia fue 0,003, mientras los GFLOPs fueron 67,7 y 21,3, respectivamente. YOLO11s fue seleccionada por el equilibrio entre desempeño y complejidad, decisión realizada utilizando resultados de test y que se reconoce como parte del proceso de selección.

La tercera etapa estudió el cambio de dominio. El nuevo modelo base E1 alcanzó 0,625 de mAP50-95 en Mobility Aids y 0,534 en USM. La diferencia confirma que el cambio de dominio modifica el comportamiento de un detector entrenado en un contexto visual específico.

La cuarta etapa comparó cinco estrategias. E2 mostró la mayor especialización en USM, con 0,852 de mAP50-95, pero redujo Mobility Aids a 0,549. E3 alcanzó 0,757 en USM y E4 0,755, mostrando que el entrenamiento adicional sobre el mismo dominio no incrementó el mAP50-95 dentro de USM. E5 alcanzó 0,851 en USM y 0,633 en Mobility Aids mediante entrenamiento conjunto.

Para la comparación interdominio se utilizaron las tres clases comunes. E5 presentó los mayores valores combinados observados: aproximadamente 0,603 en Mobility Aids y 0,830 en USM, con promedio 0,717 y mínimo 0,603. La conclusión se restringe a las configuraciones directamente comparables y a una sola corrida por configuración.

5.2 Conclusiones metodológicas

La primera conclusión metodológica es que la partición de datos debe considerar cómo fueron generadas las imágenes. En Mobility Aids se adoptó una unidad de partición a nivel de secuencia. La auditoría confirmó que ninguna secuencia se comparte entre train y validation y que existe una separación temporal operacional de 44,52 s. Esto responde de manera directa al riesgo identificado en datasets de video, aunque no constituye por sí mismo una demostración de independencia estadística total.

La segunda conclusión es que los resultados de validación deben interpretarse junto con la estructura de las particiones. La decisión de utilizar secuencias completas evita que la proporción objetivo de validation fuerce una división de una secuencia en dos subconjuntos. Por esta razón, la proporción final fue 81,40 % train y 18,60 % validation, diferente del 95/5 inicialmente objetivo.

La tercera conclusión es que aumentar la capacidad de la arquitectura no produjo una mejora proporcional. YOLO11m obtuvo el máximo mAP50-95 del benchmark, pero solo superó a YOLO11s por 0,003 con un incremento importante de GFLOPs. La selección de YOLO11s se justifica, por tanto, como compromiso de diseño y no como máximo absoluto de rendimiento.

La cuarta conclusión se refiere al cambio de dominio. E1 obtuvo 0,625 en Mobility Aids y 0,534 en USM. Esta diferencia muestra que un modelo puede perder desempeño al enfrentar condiciones visuales distintas a las presentes durante el entrenamiento.

La quinta conclusión es que el fine-tuning puede mejorar sustancialmente el desempeño específico de un dominio. E2 aumentó el mAP50-95 en USM en 0,318 puntos respecto de E1. No obstante, la mejora se acompañó de una reducción en Mobility Aids. El efecto debe interpretarse como especialización hacia el dominio objetivo y no como mejora universal.

La sexta conclusión es que aumentar el número de épocas sobre un único dominio no constituye por sí solo una solución al cambio de dominio. E3 alcanzó 0,757 en USM y E4 0,755, mientras el principal efecto visible fue un cambio en el balance precision-recall. Debido a la diferencia de clases entre USM y Mobility Aids, sus evaluaciones sobre Mobility Aids no se utilizan como comparación semántica directa en el criterio interdominio.

La séptima conclusión corresponde al entrenamiento conjunto. E5 alcanzó 0,851 en USM y 0,633 en Mobility Aids. Considerando las tres clases comunes, obtuvo el mayor promedio interdominio observado, 0,717, y el mayor mínimo, 0,603, entre las configuraciones directamente comparables. Esto respalda la utilidad del entrenamiento conjunto dentro del protocolo estudiado, sin establecer superioridad universal.

Esta última conclusión debe interpretarse con cautela. E5 no parte del `best.pt` de Mobility Aids, sino directamente de `yolo11s.pt`, por lo que representa una estrategia de entrenamiento conjunto desde una inicialización común. Además, los resultados de test fueron observados durante la comparación de configuraciones y no se realizaron repeticiones con múltiples semillas. Por ello, las diferencias deben considerarse observadas y no necesariamente estables.

5.3 Cumplimiento de objetivos

El primer objetivo específico, asociado a la comparación de YOLO11n, YOLO11s y YOLO11m, fue cumplido mediante el benchmark realizado sobre la partición temporal corregida de Mobility

Aids. Se analizaron métricas de detección y complejidad computacional y se seleccionó YOLO11s como compromiso de diseño. La selección utilizó resultados de test y queda registrada como limitación del protocolo.

El segundo objetivo, relacionado con la partición de datos, fue cumplido mediante una partición final a nivel de secuencia y una auditoría de independencia operacional. Se documentaron 115 secuencias, 92 para train y 23 para validation, además de solapamientos y duplicados entre particiones.

El tercer objetivo, referido a la construcción y documentación de Dataset USM, fue cumplido dentro del alcance planteado. Se generó un conjunto propio a partir de videos, se dividió en train, validation y test, y se documentó su composición por clase. El conjunto contempla crutches, wheelchair y walking_frame y presenta una distribución desigual en entrenamiento.

El cuarto objetivo, relacionado con la evaluación cuantitativa, fue cumplido mediante precision, recall, F1, mAP50 y mAP50-95. Además, se complementaron las métricas globales con análisis por clase. Esto permitió identificar dificultades específicas que no aparecen completamente en los valores agregados.

El quinto y sexto objetivos, asociados al *fine-tuning* y a la comparación de estrategias de entrenamiento, fueron abordados mediante E2, E3, E4 y E5. Los resultados permiten establecer diferencias cuantitativas entre especialización y generalización.

5.4 Evaluación de hipótesis

La hipótesis principal planteó que YOLO11 puede detectar categorías asociadas a movilidad reducida en distintos dominios y que las estrategias de adaptación modifican la generalización. Los resultados la respaldan parcialmente: E1 mantiene desempeño en ambos dominios, E2 mejora de manera marcada USM y E5 ofrece un comportamiento equilibrado. Sin embargo, el desempeño no es uniforme entre dominios y el cambio favorable en uno puede acompañarse de una pérdida en el otro.

La hipótesis secundaria planteó que el entrenamiento conjunto de Mobility Aids y USM puede producir un comportamiento más equilibrado entre ambos dominios. Dentro del protocolo y utilizando solo las tres clases comunes, E5 alcanza aproximadamente 0,717 de promedio y 0,603 de mínimo, frente a aproximadamente 0,562 y 0,533 de E1 y 0,691 y 0,530 de E2. En las métricas globales de test, E5 alcanza 0,633 de mAP50-95 en Mobility Aids y 0,851 en USM. Por lo tanto, la hipótesis recibe apoyo dentro del conjunto de configuraciones directamente comparables, pero no se establece como una afirmación universal ni como un resultado estadísticamente estable.

5.5 Limitaciones

- La partición original del entrenamiento de Mobility Aids utilizada por los autores de 2019 no pudo ser reconstruida exactamente a partir de la información disponible. Se utilizó una partición propia por secuencias, manteniendo separado el test publicado y auditando la ausencia de solapamiento entre train y validation.
- La partición final combina un criterio temporal a nivel de secuencia con cobertura de clases y una búsqueda de distribución razonable. El procedimiento reduce el riesgo de mezclar

frames temporalmente cercanos, pero una sola partición no permite estimar la variabilidad estadística de la decisión.

- Dataset USM fue construido a partir de videos propios y utiliza actores para representar situaciones de uso de ayudas técnicas. Por ello, el conjunto constituye un dominio experimental útil, pero no representa necesariamente toda la variabilidad de situaciones reales.
- Dataset USM contiene tres clases, mientras Mobility Aids utiliza cinco clases activas. Por esta razón, las comparaciones interdominio se calculan únicamente con crutches, wheelchair y walking_frame. Las evaluaciones de E3 y E4 sobre Mobility Aids realizadas con su head de tres clases se consideran diagnósticas y no se utilizan como evidencia cuantitativa directamente comparable con los modelos de cinco clases.
- La clase wheelchair presenta una dificultad persistente en Mobility Aids. La cantidad y distribución de sus ejemplos son distintas entre train, validation y test, por lo que su desempeño debe analizarse considerando la composición del conjunto y no únicamente la capacidad de la arquitectura.
- E2 modifica simultáneamente el dominio de entrenamiento y varios hiperparámetros. Por ello, no es posible atribuir la mejora específica de USM a una sola variable de forma causal.
- Los experimentos E3 y E4 permiten estudiar el efecto de prolongar el entrenamiento, pero no controlan múltiples alternativas de aprendizaje que podrían afectar la comparación.
- El conjunto adicional obtenido mediante colaboración con el Metro de Valparaíso no fue integrado cuantitativamente debido a restricciones de tiempo. Por lo tanto, no existen conclusiones experimentales de esta tesis sobre ese dominio.
- La comparación temporal con Kollmitz et al. no utiliza el mismo hardware ni el mismo pipeline. Las cifras de latencia se presentan solo como referencia contextual.
- No se realizó una campaña sistemática de múltiples semillas o repeticiones independientes. Por lo tanto, las diferencias pequeñas entre configuraciones, como las observadas entre E1 y E5 en Mobility Aids, no permiten afirmar estabilidad estadística.

El benchmark de YOLO11n, YOLO11s y YOLO11m se considera realizado sobre la partición temporal corregida de Mobility Aids. La misma preparación de datos se utiliza como base homogénea para comparar las tres variantes de arquitectura.

5.6 Recomendaciones y trabajo futuro

La primera línea de trabajo futuro corresponde a incorporar de manera sistemática el dataset obtenido en colaboración con el Metro de Valparaíso. Este recurso puede aportar escenas reales del transporte público y permitiría estudiar un dominio adicional, más cercano a una situación operacional, sin reemplazar los conjuntos utilizados en esta investigación.

Una segunda línea consiste en ampliar Dataset USM mediante una mayor diversidad de personas, ayudas técnicas, distancias, perspectivas, fondos, iluminación y oclusiones. Debido a que los videos actuales fueron contruidos con actores, ampliar la captura con situaciones reales permitiría reducir una fuente de sesgo de dominio.

Una tercera línea consiste en abordar la clase wheelchair de Mobility Aids mediante una estrategia específica de datos. La mejora no debería limitarse a añadir más frames similares, sino aumentar la diversidad visual de las instancias y controlar su distribución entre entrenamiento, validación y test.

Una cuarta línea es realizar experimentos de *fine-tuning* controlados por variable. Para determinar qué parámetros contribuyen realmente a la mejora observada, se debería mantener constante el dataset y modificar por separado learning rate, número de épocas, scheduler y augmentation. Este diseño permitiría separar el efecto del cambio de dominio del efecto de la configuración de optimización.

Una quinta línea corresponde a evaluar varias particiones temporales independientes. En lugar de utilizar una única separación, podrían construirse varios cortes temporales y repetirse la evaluación. Esto permitiría estimar la estabilidad de las métricas y determinar si el comportamiento observado se mantiene en diferentes segmentos de las secuencias.

Una sexta línea es complementar el mAP con análisis de curvas precision-recall, matrices de confusión, error de localización, distribución de tamaños de objetos y métricas por distancia o condición de iluminación. Estos indicadores permitirían caracterizar con mayor precisión las causas de los errores.

Una séptima línea consiste en medir la latencia sobre el hardware objetivo bajo un protocolo común. Para comparar arquitecturas de manera rigurosa, deberían fijarse resolución, batch, dispositivo, versión del framework y procedimiento de inferencia. De esta manera sería posible determinar si una mejora marginal de mAP justifica el aumento de complejidad computacional.

Finalmente, una extensión posterior podría incorporar seguimiento temporal. El trabajo de Kollmitz et al. muestra que la información temporal puede utilizarse para estabilizar las detecciones y estimar posición y clase a lo largo de una secuencia [2]. Sin embargo, esta ampliación debería desarrollarse después de consolidar el detector frame a frame, dado que la presente tesis se limita deliberadamente a la detección.

5.7 Conclusión final

La conclusión principal de esta investigación es que la detección de categorías asociadas a movilidad reducida mediante YOLO11 es técnicamente viable dentro de los conjuntos estudiados, pero su desempeño depende de la partición, el dominio y la estrategia de entrenamiento. La auditoría temporal de Mobility Aids demuestra que la estructura de secuencias debe formar parte del diseño experimental y no tratarse como una colección arbitraria de frames.

En el benchmark de arquitectura realizado sobre la partición temporal corregida, YOLO11m obtuvo 0,631 de mAP50-95 y YOLO11s 0,628. La diferencia fue 0,003 frente a un aumento de complejidad de 21,3 a 67,7 GFLOPs. Por ello, YOLO11s se mantuvo como arquitectura de referencia bajo un criterio de compromiso entre desempeño y costo. Dado que la decisión utilizó el test, esta selección forma parte de la limitación metodológica del estudio.

En los experimentos entre dominios, E2 produjo el mayor resultado específico sobre USM, con 0,852 de mAP50-95. La mejora respecto de E1 fue de 0,318 puntos, mientras Mobility Aids

disminuyó de 0,625 a 0,549. El resultado evidencia una especialización marcada hacia USM, acompañada de pérdida en el dominio de origen.

E3 y E4 muestran que el entrenamiento exclusivo sobre USM mantiene un desempeño elevado dentro del dominio, pero prolongar el entrenamiento no incrementó su mAP50-95, que pasó de 0,757 a 0,755. Sus resultados sobre Mobility Aids no se utilizan como comparación semántica directa porque el head de ambos modelos contiene solo las tres clases de USM.

E5 corresponde a un entrenamiento conjunto iniciado directamente desde `yolo11s.pt`, utilizando simultáneamente Mobility Aids y USM. Obtuvo 0,633 de mAP50-95 en Mobility Aids y 0,851 en USM. Considerando exclusivamente las tres clases comunes, alcanzó aproximadamente 0,717 de promedio y 0,603 de mínimo, los valores más altos observados entre las configuraciones directamente comparables. Este resultado se interpreta como un compromiso favorable dentro del protocolo, no como una superioridad estable.

La principal contribución metodológica del trabajo radica en mostrar que la generalización no depende únicamente de la arquitectura. La forma de particionar datos secuenciales, el cambio de dominio y la estrategia de entrenamiento producen efectos observables en las métricas. La comparación entre E2 y E5 permite distinguir especialización secuencial de entrenamiento conjunto.

Los resultados no permiten afirmar que el modelo final sea óptimo para todos los escenarios urbanos, cámaras o poblaciones. La evidencia se limita a Mobility Aids y Dataset USM. En particular, E5 demuestra un comportamiento favorable sobre estos dos dominios conocidos, pero no constituye evidencia de generalización hacia un tercer dominio como Metro Valparaíso.

En síntesis, la tesis muestra que YOLO11s puede adaptarse a distintos contextos de detección asociados a movilidad reducida y que, dentro del protocolo estudiado, el entrenamiento conjunto de Mobility Aids y USM ofrece el mejor equilibrio observado entre los dos dominios cuando se comparan únicamente las clases compartidas. La conclusión se acompaña de controles de partición temporal y de una declaración explícita sobre el uso del test durante el desarrollo experimental.

Desde una perspectiva metodológica, una de las principales lecciones es que la validez de una evaluación no depende solo de la métrica final, sino también de cómo se construyen las particiones, cómo se relacionan los espacios de clases y qué papel cumplen los conjuntos de prueba durante la selección. La trazabilidad de videos, checkpoints, YAML y configuraciones constituye una parte integral de la evidencia experimental.

Asimismo, los experimentos muestran que la adaptación debe entenderse como una decisión dependiente del objetivo. E2 maximiza la especialización en USM, mientras E5 mantiene un desempeño elevado en ambos dominios. Ninguna de las dos estrategias puede considerarse universalmente mejor; su utilidad depende del objetivo operativo y del conjunto de datos disponible.

Por último, el trabajo establece una base experimental sobre la cual pueden desarrollarse futuras etapas con datos reales adicionales. La incorporación del conjunto obtenido mediante colaboración con el Metro de Valparaíso, la ampliación de Dataset USM y la evaluación temporal en múltiples cortes son extensiones naturales. Estas futuras investigaciones deberán

TRABAJO DE TITULO

conservar los principios de control experimental establecidos en esta tesis para asegurar que las mejoras observadas representen avances en generalización y no solamente un mejor ajuste a una colección particular de frames.

REFERENCIAS BIBLIOGRÁFICAS

- [1] A. Vasquez, M. Kollmitz, A. Eitel and W. Burgard, "Deep Detection of People and their Mobility Aids for a Hospital Robot," in Proc. European Conference on Mobile Robots (ECMR), 2017.
- [2] M. Kollmitz, A. Eitel, A. Vasquez and W. Burgard, "Deep 3D Perception of People and Their Mobility Aids," Robotics and Autonomous Systems, vol. 114, pp. 29-40, 2019, doi: 10.1016/j.robot.2019.01.011.
- [3] F. Dumitrescu, C.-A. Boiangiu and M.-L. Vongilă, "Fast and Robust People Detection in RGB Images," Applied Sciences, vol. 12, no. 3, art. 1225, 2022, doi: 10.3390/app12031225.
- [4] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
- [5] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345-1359, 2010, doi: 10.1109/TKDE.2009.191.
- [6] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto and E. A. B. da Silva, "A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit," Electronics, vol. 10, no. 3, art. 279, 2021, doi: 10.3390/electronics10030279.
- [7] G. Jocher and J. Qiu, "Ultralytics YOLO11," Ultralytics, 2024. [Online]. Available: <https://docs.ultralytics.com/models/yolo11>
- [8] Ultralytics, "YOLO Performance Metrics," Ultralytics Documentation. [Online]. Available: <https://docs.ultralytics.com/guides/yolo-performance-metrics>
- [9] Ultralytics, "Fine-Tuning Guide," Ultralytics Documentation. [Online]. Available: <https://docs.ultralytics.com/guides/finetuning-guide>
- [10] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788.
- [11] R. Girshick, "Fast R-CNN," in Proc. IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440-1448.
- [12] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in Proc. Advances in Neural Information Processing Systems, vol. 28, 2015.
- [13] Y. Chen, W. Li, C. Sakaridis, D. Dai and L. Van Gool, "Domain Adaptive Faster R-CNN for Object Detection in the Wild," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 3339-3348, doi: 10.1109/CVPR.2018.00352.

ANEXOS

ANEXO A. Resumen de configuración de entrenamiento

Tabla A1. Parámetros documentados de entrenamiento para E1 y E2.

Parámetro	Mobility Aids E1	Fine-tuning E2
Modelo inicial	yolo11s.pt	best.pt de E1 corregido
Dataset	Mobility Aids	USM
Epochs	Calculadas para aproximar 60.000 iteraciones	50
Batch	16	16
Image size	640	640
Optimizer	SGD	SGD
Learning rate	0,0025	0,0005
Momentum	0,9	0,9
Weight decay	0,0001	0,0001
Warm-up	500 iteraciones	1 época
Scheduler	Personalizado por iteración	Coseno
Mosaic	Configuración del framework; valor no fijado en el script principal	0,5
Mixup	Configuración del framework; valor no fijado en el script principal	0
Label smoothing	No especificado en el script principal	0,05

Fuente: Elaboración propia a partir de los scripts entregados para esta investigación.

ANEXO B. F1 calculado a partir de precision y recall

Tabla A2. Valores de F1 calculados mediante $F1 = 2PR/(P+R)$.

Experimento	Dominio	Precision	Recall	F1 calculado
E1	Mobility Aids	0,934	0,805	0,865
E2	Mobility Aids	0,536	0,756	0,627
E3	Mobility Aids	N/C	N/C	N/C
E4	Mobility Aids	N/C	N/C	N/C
E5	Mobility Aids	0,945	0,808	0,871
E1	USM	0,759	0,731	0,745
E2	USM	0,949	0,946	0,947
E3	USM	0,509	0,957	0,665
E4	USM	0,894	0,742	0,811
E5	USM	0,996	0,994	0,995

Fuente: Elaboración propia.

ANEXO C. Consideraciones de reproducibilidad

- El test de Mobility Aids se mantuvo separado durante la preparación del modelo base y de los experimentos posteriores.
- La partición definitiva de Mobility Aids utilizó secuencias completas como unidad, con 92 secuencias en train y 23 en validation; la auditoría confirmó cero solapamientos y una separación temporal de 44,52 s.
- E2 parte del best.pt generado por el nuevo E1 de Mobility Aids y continúa el aprendizaje exclusivamente con Dataset USM, utilizando la configuración de fine-tuning registrada.
- E5 parte directamente de yolo11s.pt y entrena conjuntamente sobre Mobility Aids + USM. No corresponde a una continuación desde best.pt de Mobility Aids.
- El criterio interdominio utiliza únicamente las tres clases compartidas por ambos datasets; por ello E3 y E4 no se incluyen en la comparación directa sobre Mobility Aids hasta disponer de una evaluación semánticamente remapeada.
- Los resultados de test fueron utilizados para comparar configuraciones E1-E5 y para seleccionar la arquitectura; esta participación se reconoce como limitación metodológica. Los test permanecieron físicamente separados del entrenamiento y no se realizó una repetición sistemática con múltiples semillas.

ANEXO D. Auditoría y trazabilidad de la partición temporal de Mobility Aids

La partición definitiva fue generada mediante `creardataset\_v2.py`. El registro de ejecución documenta 115 secuencias, 92 para entrenamiento y 23 para validation. Se seleccionó la frontera a nivel de secuencia con cobertura de las cinco clases activas y sin solapamiento entre particiones.

Archivos de trazabilidad generados: `split\_manifest.csv`, `videos\_summary.csv`, `class\_distribution.csv`, `audit\_report.json`, `overlap\_report.txt`, `run\_config.json` y `data.yaml`.

Resumen de auditoría: solapamiento de videos = 0; solapamiento de basenames = 0; duplicados por SHA-256 entre train y validation = 0; último timestamp de train = 1477481081.4941281040; primer timestamp de validation = 1477481126.0145338060; separación temporal = 44.5204057020 s.

Inventario por bloques: train = secuencias 1–92; validation = secuencias 93–115. La asignación es por secuencia completa, de modo que todas las imágenes de una misma secuencia pertenecen a una única partición.

Parámetro	E1	E2	E3	E4	E5
Modelo inicial	yolo11s.pt	best.pt de E1 corregido	yolo11s.pt	last.pt de E3	yolo11s.pt
Dataset	Mobility Aids corregido	USM	USM	USM	Mobility Aids + USM
Épocas	Calculadas para 60.000 iteraciones	50	200 máx.	400 máx.	50
Patience	100	100	100	100	100

TRABAJO DE TITULO

Parámetro	E1	E2	E3	E4	E5
Batch	16	16	16	16	16
Imagen	640	640	640	640	640
Optimizer	SGD	SGD	SGD	SGD	SGD
Learning rate	0,0025	0,0005	0,003	0,003	0,0005
Momentum	0,9	0,9	0,9	0,9	0,9
Weight decay	0,0001	0,0001	0,0001	0,0001	0,0001
Warm-up	500 iteraciones	1 época	3 épocas	3 épocas	1 época
Scheduler	Personalizado por iteración	Coseno	Coseno	Coseno	Coseno
Mosaic	framework	0,5	1,0	1,0	0,5
Mixup	framework	0	0	0	0
Seed	42	0	0	0	0
Framework / GPU	Ultralytics 8.3.191 / RTX 4060	Ultralytics 8.3.191 / RTX 4060	Ultralytics 8.3.191 / RTX 4060	Ultralytics 8.3.191 / RTX 4060	Ultralytics 8.3.191 / RTX 4060