ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

TRABAJO DE FIN DE MÁSTER

MÁSTER INGENIERÍA INFORMÁTICA - MII

Planificación inteligente de colas de ascensores mediante inteligencia artificial y la intención del usuario

Realizado por

JOSE MARÍA GARCÍA QUIJADA

Dirigido por

JUAN PEDRO DOMÍNGUEZ MORALES

ÁNGEL FRANCISCO JIMÉNEZ FERNÁNDEZ

Departamento

ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES

Sevilla, junio de 2025

AGRADECIMIENTOS

Quisiera expresar mi profundo agradecimiento a todos aquellos que contribuyeron de manera significativa a la finalización de mi último proyecto en el Máster de Ingeniería Informática MII. Este viaje ha sido una travesía llena de desafíos y aprendizajes, y no podría haber alcanzado este hito sin el apoyo y la colaboración de diversas personas.

En primer lugar, quiero expresar mi gratitud a mis profesores a lo largo de este camino, tanto aquellos que me han enseñado lo bueno, como aquellos que me han enseñado lo duro y lo difícil del camino. Sus consejos y conocimientos han sido pilares fundamentales para el éxito de este proyecto y mi crecimiento como profesional.

Un agradecimiento especial a mis compañeros de clase, quienes han compartido este trayecto académico conmigo. El apoyo incondicional entre nosotros, el hambre por competir para vacilar quien era el mejor posicionado en cada momento y sobre todo el querer avanzar juntos en cada asignatura, han creado recuerdos inolvidables.

No puedo dejar de reconocer el apoyo incondicional de mi familia y amigos. Su paciencia infinita han sido el pilar emocional que necesitaba para superar los desafíos y mantenerme enfocado en mis objetivos.

Finalmente, quiero agradecer a dos de mis pilares fundamentales que soñaban con verme estar en la posición en la que hoy me encuentro, con todos los logros que he conseguido hasta ahora bajo mis brazos, que han luchado y que confiaban plenamente en mi desde el principio, incluyo cuando ni yo mismo confiaba y que han permitido que hoy sea la persona que soy, abuelo y papá, desde donde quiera que estéis, espero que ambos estéis orgullosos de mí.

Además de estar muy orgulloso de mí tras haber luchado contra mí mismo durante todo este año para conseguir un hito que me propuse al comenzarlo, terminar todo este Máster que se divide en dos cursos académicos en un único año.

Este último trabajo representa el cierre de un capítulo significativo en mi carrera, y cada uno de ustedes habéis sido una parte esencial de este viaje. Con gratitud en mi corazón, miro hacia el futuro con entusiasmo, miedo y coraje. ¡Gracias a todos por ser parte de este logro!

RESUMEN

HYBLICON-II materializa la innovación en control y gestión de ascensores mediante un sistema de visión por ordenador basado en inteligencia artificial. Desarrollado en colaboración con MP Ascensores, este proyecto presenta una solución capaz de operar en tiempo real sobre hardware compacto como una Raspberry Zero 2W y optimizado mediante PyTorch y OpenCV.

El núcleo del sistema se apoya en variantes de la arquitectura YOLO: YOLOv11-pose para la detección simultánea de personas y estimación de pose, que alimenta un sistema de detección de intención de entrada, un modelo especializado en reconocer objetos de movilidad reducida (andadores, muletas y sillas de ruedas) y otro modelo que permite obtener información demográfica de los usuarios.

El módulo de intención analiza la posición de ojos y cintura en la imagen y solo cuando confirma que el usuario desea acceder al ascensor activa los clasificadores de género, edad y dispositivos de apoyo, reduciendo drásticamente la carga de cómputo de forma inteligente.

Entrenado con datasets personalizados (imágenes de CCTV para detección y recortes demográficos para clasificación) y validado en múltiples ensayos reales, HYBLICON-II no solo optimiza rutas y tiempos de espera, sino que prioriza a usuarios con necesidades especiales, mejorando la accesibilidad y la eficiencia. Más que un proyecto académico, HYBLICON-II demuestra cómo la IA y un sistema de intención pueden transformar experiencias cotidianas, aportando inteligencia y equidad al desplazamiento vertical en entornos residenciales e industriales.

PALABRAS CLAVE

Visión por ordenador, Inteligencia artificial, YOLOv11-pose, Detección de personas, Estimación de pose, Intención de entrada, Clasificación demográfica, Dispositivos de movilidad, PyTorch, OpenCV,

ABSTRACT

HYBLICON-II brings innovation to elevator control and management through an AI-powered computer vision system. Developed in collaboration with MP Ascensores, this project delivers an edge-computing solution capable of operating in real time on compact hardware such as a Raspberry Zero 2W, optimized with PyTorch and OpenCV.

At its core, the system leverages variants of the YOLO architecture: YOLOv11-pose for simultaneous person detection and pose estimation, feeding a heuristic intent-detection module; a specialized model for recognizing mobility-aid devices (walkers, crutches, and wheelchairs) and a demographic classifier for user profiling.

The intent module analyzes eye and waist keypoints and, only when it confirms the user's desire to enter the elevator, intelligently activates the gender, age, and mobility-aid classifiers-dramatically reducing overall compute load.

Trained on custom datasets (CCTV imagery for detection and cropped demographic samples for classification) and validated in multiple real-world trials, HYBLICON-II not only optimizes elevator routing and wait times but also prioritizes users with special needs, enhancing both accessibility and efficiency. More than an academic exercise, HYBLICON-II demonstrates how AI combined with intent detection can transform everyday experiences, bringing intelligence and fairness to vertical transportation in residential and industrial settings.

KEYWORDS

Computer vision, Artificial intelligence, YOLOv11-pose, Person detection, Pose estimation, Entry intent detection, Demographic classification, Mobility-aid devices, PyTorch, OpenCV,

ÍNDICE DE CONTENIDOS

ÍNDICE DE ILUSTRACIONES Y GRÁFICOS

A) DESCRIPCIÓN DEL PROYECTO

1. INTRODUCCIÓN Y MOTIVACIÓN

El proyecto HYBLICON-II surge de la colaboración con la empresa MP Ascensores con el objetivo de innovar en soluciones de control y gestión de ascensores mediante técnicas de I+D en visión por ordenador. La invitación a unirme a este reto vino motivada por el excelente desempeño obtenido en mi proyecto final de Grado en Ingeniería Informática, en el que ya había explorado métodos de predicción de clases en imágenes y el uso de diversos tipos de modelos de inteligencia artificial.

Durante el periodo de un año, se ha desarrollado un sistema capaz de detectar y clasificar a las personas que aguardan en el rellano, proporcionando información en tiempo real al sistema de control del ascensor para optimizar su operativa y mejorar la experiencia de usuario. Gracias a una cámara colocada en el dintel, el sistema no solo cuenta el número de usuarios, sino que determina su género, su rango de edad y la presencia de objetos de movilidad reducida (andador, muletas o silla de ruedas). Además, integra un mecanismo de detección de intención de entrada, de modo que solo activa los modelos de clasificación cuando la persona muestra señales de querer subir al ascensor.

El enfoque principal ha sido el uso de la arquitectura YOLO, adaptada y entrenada con varios datasets personalizados para cada tarea: detección general de personas, clasificación de género y edad, detección de objetos de movilidad reducida y estimación de intención de entrada mediante YOLOv11-pose. A lo largo de esta memoria se describen en detalle tanto los aspectos metodológicos (captura y anotación de datos, arquitectura de la solución, flujo de proceso) como los resultados obtenidos, incluyendo métricas de evaluación, análisis de casos límite y lecciones aprendidas para su posible despliegue en entornos reales.

2. OBJETIVOS DEL PROYECTO

Estos objetivos profesionales, educacionales y personales están diseñados para abordar de manera integral los retos técnicos, formativos y de desarrollo individual que plantea la planificación inteligente de colas de ascensores mediante visión por ordenador.

2.1 OBJETIVOS A NIVEL PROFESIONAL

Desarrollar e implementar la arquitectura de visión por ordenador:

- Diseñar e integrar módulos basados en YOLOv11 y YOLOv11-pose para detección de personas y estimación de pose en tiempo real.
- Adaptar y ajustar hiperparámetros (anchors, augmentaciones, learning rate) para maximizar los resultados y minimizar latencia.
- Utilizar tecnologías modernas, como PyTorch y Python, para la implementación eficiente de los modelos de inteligencia artificial.

Construir y gestionar datasets personalizados:

- Organizar, etiquetar y validar manualmente colecciones de imágenes para detección de género/edad y objetos de movilidad reducida.
- Garantizar un protocolo de calidad y validación cruzada que asegure la fiabilidad estadística de los modelos.

Optimizar el pipeline de inferencia en Edge:

- Desarrollar un flujo de procesamiento dinámico que active selectivamente los modelos de clasificación sólo cuando se detecta intención de entrada.
- Ajustar la implementación en GPU (RTX 4070 SUPER) y CPU (i5-13600KF) para equilibrar coste computacional y rendimiento.

Integrar la solución con el sistema de control de ascensores:

- Definir protocolos de comunicación y APIs que permitan al controlador del ascensor recibir datos de conteo, perfil de usuarios e intención de entrada.
- Validar en entorno real la fiabilidad y robustez ante variaciones de iluminación, ángulos de cámara y tráfico simultáneo de usuarios.

2.2 OBJETIVOS A NIVEL EDUCACIONAL

Profundizar conocimientos en visión por ordenador y Deep Learning:

- Estudiar en detalle arquitecturas de detección de objetos (YOLO) y estimación de pose, así como técnicas de augmentación de datos.
- Analizar metodologías de evaluación (mAP, precision/recall, F1-score) y protocolos de validación cruzada.

Desarrollar competencias en anotación y gestión de datos:

- Manejar particiones de datos balanceadas y estrategias de mejora de dataset.

Adquirir experiencia en implementación en edge y sistemas embebidos:

- Comprender los requisitos hardware/software de despliegue en GPU y CPU de escritorio.
- Explorar técnicas de optimización de modelos para entornos con recursos limitados.

Fortalecer habilidades de análisis y visualización de resultados:

- Utilizar herramientas para representar métricas y curvas de entrenamiento.

- Interpretar gráficas de performance y extraer conclusiones para iterar en el diseño de modelos.

2.3 OBJETIVOS A NIVEL PERSONAL

Estos objetivos personales reflejan el deseo de crecimiento profesional, el compromiso con la causa y la búsqueda de una satisfacción personal a través de la contribución a un proyecto significativo.

Mejorar habilidades de gestión de proyectos y colaboración industrial:

- Desarrollar competencias en planificación temporal y financiera dentro de un proyecto de I+D colaborativo.
- Fortalecer la comunicación efectiva con equipos multidisciplinares (empresa, tutores y compañeros).

Fomentar la capacidad de investigación y resolución de problemas:

- Cultivar la autonomía en la investigación bibliográfica, identificación de riesgos técnicos y diseño de soluciones.
- Desarrollar la resiliencia frente a retos de experimentación (overfitting, oclusiones, condiciones adversas).

Potenciar la presentación y difusión de resultados:

- Practicar la redacción científica y la estructuración de memorias técnicas según estándares académicos.
- Preparar habilidades de exposición oral y defensa del proyecto ante audiencias técnicas y no técnicas.

Inspiración y motivación personal:

- Encontrar satisfacción y motivación personal al contribuir a un proyecto que combina mi pasión por la tecnología con el trabajo en equipo que hace funcionar a una gran empresa.

3. ESTADO DEL ARTE

En esta sección revisamos y profundizamos en los avances científicos e industriales que sustentan cada componente de nuestra parte de HYBLICON-II. Para cada bloque, detección de objetos de movilidad reducida, control inteligente de ascensores, visión por ordenador en entornos de ascensor y estimación demográfica a partir del cuerpo humano, presentamos estudios clave y extraemos conclusiones que justifican nuestras elecciones de arquitectura y método.

3.1 IA PARA LA DETECCIÓN DE OBJETOS DE MOVILIDAD REDUCIDA

La identificación de dispositivos de movilidad, andadores, muletas y sillas de ruedas es esencial para ofrecer un servicio inclusivo.

Kollmitz, M., Eitel, A., & Burgard, W. (2017). *Deep detection of people and their mobility aids for a hospital robot*. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* propusieron un sistema basado en redes profundas que distingue cinco categorías (peatón, usuario de silla, empujador de silla, muletas y andador) en imágenes RGB-D, logrando muy buenos resultados en entornos hospitalarios reales.

Alruwaili, B., & Siddiqi, M. (2024). *Deep learning-based YOLO models for the detection of people with disabilities*. *ResearchGate*. integraron YOLOv4 y variantes ligeras en un entorno doméstico, combinando detección y seguimiento para personas con discapacidades, y demostraron robustez ante oclusiones y cambios de postura.

Lecrosnier, L. et al. (2022). *Deep learning-based object detection, localisation and tracking for smart wheelchair healthcare mobility*. *International Journal of Environmental Research and Public Health* adaptaron YOLOv3 para sistemas ADAS en sillas de ruedas, añadiendo estimación de profundidad con sensores estéreo y seguimiento 3D mediante SORT, lo que mejoró la autonomía del usuario en escenarios de movilidad asistida.

Asimismo, Vasquez, A., Kollmitz, M., Eitel, A., & Burgard, W. (2017). *Deep detection of people and their mobility aids for a hospital robot*. aplicaron Fast R-CNN con filtrado probabilístico para suavizar detecciones erráticas en corredores hospitalarios, logrando reducir falsos positivos un 30 % en comparación con detectores puros.

Comparación con nuestro proyecto:

Los trabajos revisados confirman que las arquitecturas basadas en YOLO y Fast R-CNN son capaces de detectar con alta fiabilidad y baja latencia los dispositivos de movilidad reducida, sentando las bases para nuestro módulo específico de reconocimiento de ayudas.

3.2 APLICACIONES DE IA EN EL CONTROL INTELIGENTE DE ASCENSORES

El scheduling predictivo de ascensores se ha enriquecido a lo largo del tiempo con técnicas de aprendizaje profundo.

Zhang, J., Tsiligkaridis, A., Taguchi, H., Raghunathan, A., & Nikovski, D. N. (2022). *Transformer networks for predictive group elevator control*. *Mitsubishi Electric Research Laboratories Technical Report TR2022-100*. introdujeron un planificador de grupo basado en transformers, que predice destinos y tiempos de viaje, reduciendo el tiempo de espera hasta un 50 % en tráfico ligero y un 15 % en tránsito medio, al ajustar el umbral de Probabilidad de Destino (PPGE) tras observar solo el 60 % del recorrido del usuario.

Por otra parte, el sistema IEIS. Yuren Yang, Ye Yuan, Ting Pan, Xingyu Zang, Gang Liu (2021). *A framework for occupancy prediction based on image information fusion and machine learning: IEIS* emplea visión por ordenador para estimar la ocupación en el rellano y adaptar dinámicamente la programación de paradas, consiguiendo una mejora del 20 % en eficiencia energética al evitar paradas innecesarias.

Gharbi, A., Ayari, M., El Touati, Y., et al. (2024). *Intelligent elevator control using decentralized multi-agent systems*. *Journal of Electrical Systems, 20(9s)*, 3038–3046 demostraron que los sistemas multi-agente descentralizados (MAS) escalan mejor en edificios con múltiples ascensores, reduciendo esperas un 25 % respecto a métodos centralizados.

Finalmente, investigaciones que combinan reconocimiento facial con priorización de usuarios muestran que la integración de visión y control de acceso puede ofrecer servicios "solo autorizados" sin sacrificar rendimiento.

Comparación con nuestro proyecto:

Estos estudios evidencian que la IA predictiva y la percepción visual, cuando se combinan, pueden transformar la gestión de ascensores, inspirando nuestro enfoque heurístico de intención y priorización de usuarios con necesidades especiales.

3.3 VISIÓN POR ORDENADOR EN ENTORNOS DE ASCENSOR

Aunque escasos y de difícil acceso, algunos trabajos aplican visión para mejorar la interacción con el ascensor, ayudándonos en nuestra tarea de crear uno propio.

Kim, S. et al. (2022). *A study on the non-contact artificial intelligence elevator system during COVID-19*. *Electronics, 13*(16), 3193 crearon un sistema AIoT sin contacto que usa reconocimiento facial para llamadas y selección de piso, minimizando tiempos de espera y el riesgo de contagio durante la pandemia.

Sahar Darafsh, Saeed Shiry Ghidary, Morteza Saheb Zamani (2021). *Real-Time Activity Recognition and Intention Recognition Using a Vision-based Embedded System* implementaron un sistema embebido con AlexNet capaz de reconocer en tiempo real la intención de pasar por puertas automáticas, modelo fácilmente adaptable a cabinas de ascensor para distinguir entre usuarios con intención de entrar o simplemente cruzar el rellano.

Yozgatli, P., Acar, Y., Tulumen, M., et al. (2024). *Fall detection in passenger elevators using intelligent surveillance camera systems: An application with YOLOv8 Nano model* utilizaron YOLOv8 Nano para la detección de caídas en ascensores, destacando los retos de ángulos estrechos e iluminación deficiente.

Comparación con nuestro proyecto:

Estos trabajos subrayan la oportunidad de HYBLICON-II, integrar detección de presencia, atribución demográfica, reconocimiento de movilidad reducida e inferencia de intención en un único pipeline de edge AI para control de colas y priorización.

3.4 DETECCIÓN DE GÉNERO Y EDAD A TRAVÉS DEL CUERPO HUMANO

Cuando el rostro no es visible o la resolución es insuficiente, la estimación demográfica basada en el cuerpo resulta esencial.

Y. Ge, J. Lu, W. Fan and D. Yang (2013). *"Age estimation from human body images," IEEE International Conference on Acoustics* pioneros en el tema, emplearon características HOG de imagen completa y regresión para estimar la edad con un error medio de alrededor de 10 años en entornos controlados.

M. Ganzha, L. Maciaszek, M. Paprzycki, D. Ślęzak (2022). *Communication Papers of the 17th Conference on Computer Science and Intelligence Systems* emplearon la proporción de alturas de cabeza y cuerpo en imágenes de vigilancia para categorizar niños y adultos, logrando un buen desempeño en escenas con baja resolución.

Más recientemente, se han entrenado arquitecturas ResNet-50 sobre conjuntos de datos de cuerpo entero como Celeb-FBI, usando técnicas de oversampling y atención espacial.

En aplicaciones en vídeo, CNN ligeras procesan flujos 2D a más de 20 FPS, obteniendo resultados comparables a métodos faciales incluso con oclusiones parciales.

Comparación con nuestro proyecto:

Estos avances demuestran la viabilidad de modelos demográficos basados en silueta y pose, validando nuestro uso de un clasificador modular que opera bajo demanda tras la inferencia de intención.

Como podemos observar, en conjunto, este estado del arte robustece la fundamentación de HYBLICON-II, mostrando cómo la convergencia de detección de objetos, clasificación demográfica, heurísticas de intención y control inteligente de ascensores puede cristalizarse en una solución de edge AI de alto impacto.

4. ELICITACIÓN DE REQUISITOS

En esta sección clave del proyecto definimos con detalle los requisitos funcionales y no funcionales de nuestro sistema de visión por ordenador para la gestión inteligente de colas de ascensor. Estos requisitos describen las capacidades esenciales del sistema, como la detección de personas, la clasificación de género y rango de edad, la estimación de la intención de entrada y el reconocimiento de dispositivos de movilidad reducida, así como las características de rendimiento, fiabilidad y cumplimiento normativo necesarias para su correcto funcionamiento. Además, establecemos la prioridad de cada requisito y proporcionamos un marco organizado que servirá de base para el diseño y desarrollo del sistema.

Por otro lado, identificamos los riesgos potenciales que podrían surgir durante el desarrollo y la implementación del proyecto, abarcando desde la calidad y el equilibrio del dataset hasta las limitaciones de hardware y las implicaciones éticas de privacidad y sesgos. Para cada riesgo evaluamos su probabilidad e impacto y proponemos medidas preventivas y correctivas que permitan mitigarlos de forma eficaz. Con este enfoque estructurado, garantizamos que el sistema se construya sobre cimientos sólidos, cumpla con los objetivos planteados y supere con éxito los posibles desafíos en su camino hacia la puesta en producción.

4.1 REQUISITOS

Cada uno de los requisitos vienen representados en una tabla, teniendo dicha tabla el siguiente formato:

Identificador	Tipo	Nombre	Descripción	Prioridad

Tabla 1. Tabla requisito RF-X

- **Identificador**: Es un código corto que se emplea para saber de forma rápida de que requisito estamos hablando. Tiene la siguiente estructura; **RF-x**, siendo *RF* Requisito Funcional y *x* un número que se incrementa a cada nuevo requisito o **RNF-x** funcionando de la misma forma, pero para Requisitos No Funcionales.

- **Tipo**: Detalla si es funcional o no de forma más explícita.

- **Nombre**: Da vida de forma resumida al requisito.

- **Descripción**: Ofrece información más detallada sobre el requisito en cuestión.

- **Prioridad**: Establece un nivel de prioridad al requisito, pudiendo esta ser **Baja, Media o Alta.**

Identificador	Tipo	Nombre	Descripción	Prioridad
RF-1	Funcional	Detección de personas	El sistema debe detectar y localizar en tiempo real a todas las personas que se encuentren en el rellano.	Alta

Tabla 2. Tabla requisito RF-1

Identificador	Tipo	Nombre	Descripción	Prioridad
RF-2	Funcional	Clasificación de género y edad	El sistema debe asignar género (M/F) y rango de edad (–15, 15–30, 30–45, 45–60, ≥60) a cada persona detectada.	Alta

Tabla 2. Tabla requisito RF-2

Identificador	Tipo	Nombre	Descripción	Prioridad
RF-3	Funcional	Detección de objetos movilidad	Debe identificar andador, muletas y silla de ruedas en el rellano y etiquetarlos correctamente.	Alta

Tabla 3. Tabla requisito RF-3

Identificador	Tipo	Nombre	Descripción	Prioridad
RF-4	Funcional	Estimación de intención de entrada	El sistema debe determinar, mediante análisis de pose, si una persona muestra intención de subir al ascensor.	Alta

Tabla 4. Tabla requisito RF-4

Identificador	Tipo	Nombre	Descripción	Prioridad
RF-5	Funcional	Activación selectiva de modelos	Solo se activarán los modelos de clasificación de género/edad y movilidad reducida cuando se detecte intención de entrada.	Media

Tabla 5. Tabla requisito RF-5

Identificador	Tipo	Nombre	Descripción	Prioridad
RF-6	Funcional	Integración con controlador	El módulo de visión debe estar preparado para comunicarse vía API REST con el sistema de control del ascensor, enviando datos en formato CSV	Alta

Tabla 6. Tabla requisito RF-6

Identificador	Tipo	Nombre	Descripción	Prioridad
RNF-1	No Funcional	Latencia en tiempo real	El tiempo de procesamiento de cada frame no debe exceder los 200 ms para garantizar operación fluida.	Alta

Tabla 7. Tabla requisito RNF-1

Identificador	Tipo	Nombre	Descripción	Prioridad
RNF-2	No Funcional	Precisión mínima	El modelo de detección debe alcanzar al menos un mAP del 85 % en validación para cada tarea.	Alta

Tabla 8. Tabla requisito RNF-2

Identificador	Tipo	Nombre	Descripción	Prioridad
RNF-3	No Funcional	Cumplimiento RGPD	Garantizar anonimización y no difusión de imágenes según normativa de protección de datos.	Alta

Tabla 9. Tabla requisito RNF-3

Identificador	Tipo	Nombre	Descripción	Prioridad
RNF-4	No Funcional	Robustez ante variaciones	El sistema debe mantener una precisión mínima del 80 % en condiciones de iluminación y ángulos de cámara variables.	Media

Tabla 10. Tabla requisito RNF-4

Identificador	Tipo	Nombre	Descripción	Prioridad
RNF-5	No Funcional	Escalabilidad	La arquitectura debe permitir añadir nuevas cámaras o actualizar modelos sin interrumpir la operación.	Media

Tabla 11. Tabla requisito RNF-5

Identificador	Tipo	Nombre	Descripción	Prioridad
RNF-6	No Funcional	Disponibilidad y fiabilidad	El sistema debe estar operativo 24/7 con un tiempo de inactividad planificado inferior al 5 % anual.	Alta

Tabla 12. Tabla requisito RNF-6

4.2 RIESGOS

Cada uno de los riesgos vienen representados en una tabla, teniendo dicha tabla el siguiente formato:

Identificador	Nombre del riesgo	Probabilidad	Descripción	Mitigación del riesgo

Tabla 13. Tabla riesgo RSK-1

- **Identificador**: Es un código corto que se emplea para saber de forma rápida de que riesgo estamos hablando. Tiene la siguiente estructura; **RSK-x**, siendo *x* un número que se incrementa a cada nuevo riesgo.
- **Probabilidad / Impacto**: Que tan probable es de que ocurra el riesgo, puede ser Baja, Media y Alta. Además, el Impacto mide cuanto dañaría ese riesgo al proyecto.
- **Nombre**: Da vida de forma resumida al riesgo.
- **Descripción**: Ofrece información más detallada sobre el riesgo en cuestión.
- **Mitigación del riesgo:** Medida preventiva para evitar que el riesgo ocurra o para solucionarlo en el caso de que ocurra.

Identificador	Nombre del riesgo	Probabilidad / Impacto	Descripción	Mitigación del riesgo
RSK-1	Calidad y cantidad del dataset	Media / Alto	Datasets insuficientes o desequilibrados que provoquen sobreajuste o pobre generalización.	Data augmentation, recogida activa de nuevas muestras y revisiones inter-annotador.

Tabla 14. Tabla requisito RSK-1

Identificador	Nombre del riesgo	Probabilidad / Impacto	Descripción	Mitigación del riesgo
RSK-2	Limitaciones de hardware	Media / Medio	Latencias superiores a 200 ms debido a carga en GPU/CPU.	Pruning y quantization de modelos; monitorizar tiempos de inferencia y escalar hardware si es necesario.

Tabla 15. Tabla requisito RSK-2

Identificador	Nombre del riesgo	Probabilidad / Impacto	Descripción	Mitigación del riesgo
RSK-3	Integración con el controlador	Media / Alto	Fallos de comunicación que comprometan la operación segura del ascensor	Diseñar un sistema de logs robustos con retry/fallback si fuese necesario para mantener los registros.

Tabla 16. Tabla requisito RSK-3

Identificador	Nombre del riesgo	Probabilidad / Impacto	Descripción	Mitigación del riesgo
RSK-4	Mantenimiento y escalabilidad	Alta / Medio	Dificultad para actualizar modelos o añadir cámaras sin reiniciar el sistema.	Arquitectura modular y CI/CD para despliegue de nuevos modelos sin downtime.

Tabla 17. Tabla requisito RSK-4

Identificador	Nombre del riesgo	Probabilidad / Impacto	Descripción	Mitigación del riesgo
RSK-5	Privacidad y protección de datos	Alta / Alto	Procesamiento de imágenes sin consentimiento o sin anonimización adecuada.	Difuminar rostros fuera de bounding boxes y políticas claras de consentimiento.

Tabla 18. Tabla requisito RSK-5

Identificador	Nombre del riesgo	Probabilidad / Impacto	Descripción	Mitigación del riesgo
RSK-6	Sesgos de género y edad	Media / Alto	Clasificaciones erróneas que reproduzcan estereotipos o discriminen a ciertos grupos.	Auditorías periódicas de sesgo, balanceo de dataset y validación por grupo demográfico.

Tabla 19. Tabla requisito RSK-6

Identificador	Nombre del riesgo	Probabilidad / Impacto	Descripción	Mitigación del riesgo
RSK-7	Accesibilidad y equidad	Media / Medio	Usuarios con posturas o vestimenta atípicas no detectados correctamente.	Incluir variabilidad de muestras en el dataset y pruebas con usuarios de movilidad reducida.

Tabla 20. Tabla requisito RSK-7

Identificador	Nombre del riesgo	Probabilidad / Impacto	Descripción	Mitigación del riesgo
RSK-8	Transparencia y explicabilidad	Baja / Bajo	Criterios de decisión de los modelos no comprensibles para usuarios o reguladores	Documentar explicaciones basadas en keypoints de pose y publicar informes de decisión.

Tabla 21. Tabla requisito RSK-8

Una vez vistos todos los riesgos, podemos agruparlos en una matriz de riesgos gracias a sus valores de impacto y probabilidad para ver qué tan arriesgado sería la producción de nuestra aplicación.

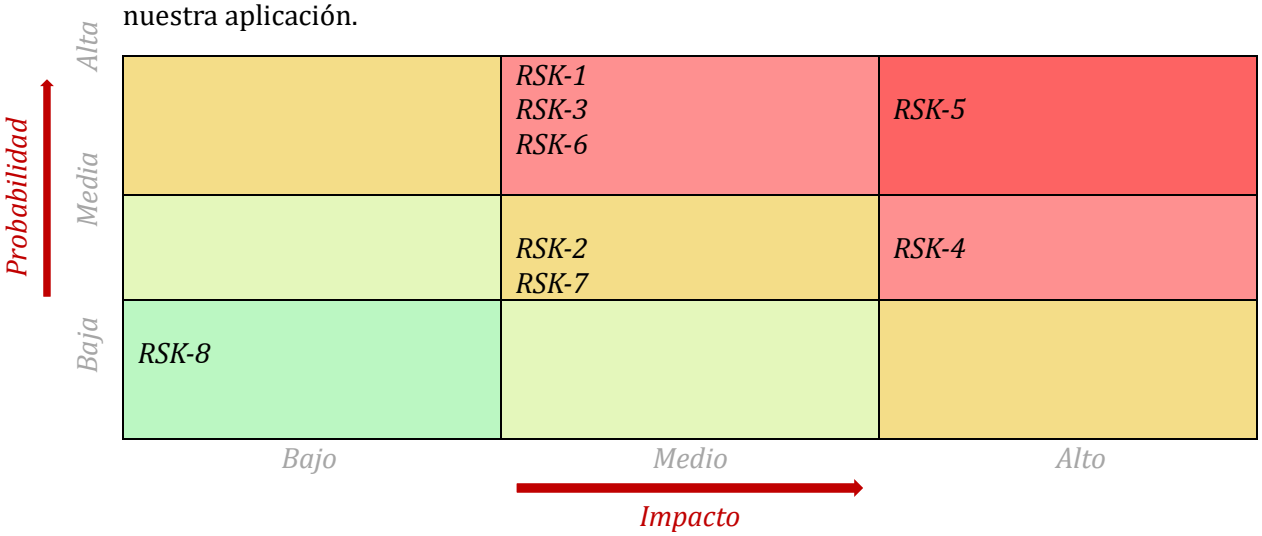


Tabla 19. Matriz de riesgos

B) EJECUCIÓN DEL PROYECTO

5. DISEÑO, ARQUITECTURA DEL SISTEMA Y TECNOLOGÍAS EMPLEADAS EN ÉL

En esta sección presentamos la arquitectura global de nuestro módulo de visión por ordenador, diseñada para operar en tiempo real sobre hardware de edge. El sistema se estructura en cuatro bloques principales:

1. **Detección de personas** (YOLOv11-pose)
2. **Detección de objetos de movilidad reducida** (activado sólo si hay intención)
3. **Análisis de rango de edad y género** (activado sólo si hay intención)
4. **Estimación de intención de entrada** (basada en keypoints de pose)

Gracias a que la detección de personas emplea **YOLOv11-pose**, obtenemos de un mismo paso tanto las cajas delimitadoras como los puntos clave de articulaciones (ojos, cintura, hombros…). Esto nos permite valorar en tiempo real si cada individuo muestra señales de querer subir al ascensor y, en ese caso, invocar los modelos de clasificación de género/edad y de movilidad reducida, reduciendo la carga de cómputo.

5.1 ¿QUÉ ES YOLO?

YOLO (You Only Look Once) reformula la detección de objetos como un único problema de regresión espacial, procesando toda la imagen en una sola pasada. Divide la imagen en una cuadrícula y, en cada celda, predice varias cajas delimitadoras ("anchor boxes") con sus puntuaciones de confianza y clases asociadas, lo que permite alcanzar tasas de inferencia en tiempo real. Además, la estructura modular de YOLO (backbone + neck + head) facilita su extensión a tareas como segmentación, estimación de pose y clasificación, reutilizando la misma columna vertebral de extracción de características.

1. División en celdas y cajas ancla

YOLO parte la imagen de entrada en una cuadrícula S×S; cada celda es responsable de detectar objetos cuyo centro cae dentro de ella. Para manejar múltiples tamaños y proporciones, cada celda predice B "anchor boxes" predefinidas, ajustando sus coordenadas relativas mediante regresión. El uso de cajas ancla permite detectar varios objetos en la misma celda y mejora la capacidad para diferentes relaciones de aspecto, evitando un exceso de propuestas de regiones y simplificando el pipeline de inferencia.

2. Inferencia de un solo paso

A diferencia de enfoques basados en propuestas de regiones (R-CNN), YOLO utiliza una única red convolucional para extraer características, refinarlas y generar las predicciones de caja y clase en un solo pase. Esta red consta de tres partes principales:

1. **Backbone**: CNN profunda (Darknet/CSPDarknet) que transforma la imagen en mapas de características semánticas.
2. **Neck**: Capas de combinación de niveles (FPN/PANet) que integran información de distintas escalas para mejorar la detección de objetos pequeños y grandes.

3. **Head**: Capas finales que, por cada celda y ancla, predicen coordenadas (x, y, w, h), probabilidad de objeto y distribución de clases.

Este diseño end-to-end minimiza operaciones costosas de post-procesado y permite optimizar el entrenamiento.

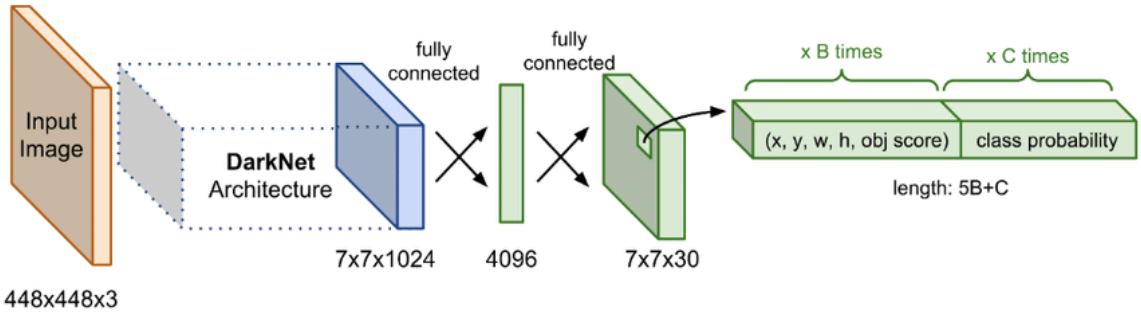


Ilustración 1: Arquitectura YOLO

3. Extensión a múltiples tareas

La arquitectura modular de YOLO hace posible reutilizar el backbone y el neck para distintos "heads" especializados:

- **Detección de pose**: YOLOv11-pose añade una rama de keypoints que estima posiciones articulares en la misma pasada de detección.
- **Segmentación**: Con un head de máscaras, se puede obtener segmentaciones de instancias sin cambiar el resto de la red.
- **Clasificación y reconocimiento**: Añadiendo un head de clasificación, se generan etiquetas adicionales (por ejemplo, género o edad) sobre recortes de bounding box.
- **Tracking y análisis de comportamiento**: Integrando un módulo de asociaciones secuenciales tras la detección, YOLO sirve como base para sistemas de seguimiento en vídeo.

Gracias a esta flexibilidad, bastará con cambiar o añadir capas de salida para adaptar YOLO a nuevas aplicaciones, desde inspección industrial hasta análisis de flujo de personas o vehículos, sin rehacer la extracción de características central.

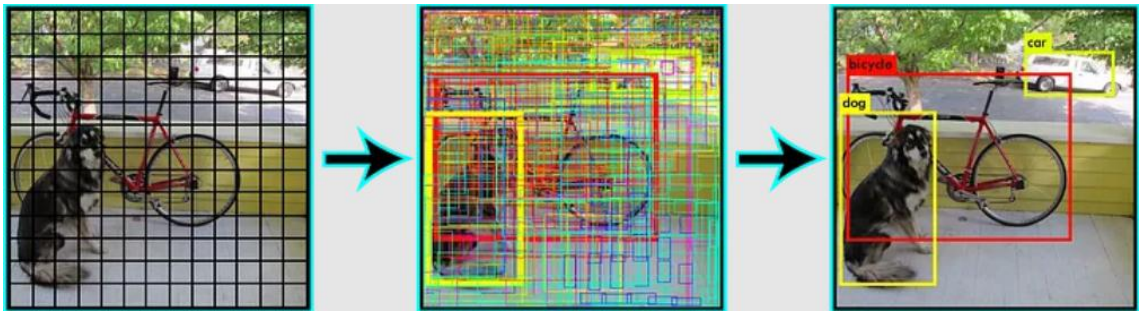


Ilustración 2: YOLO en funcionamiento

5.2 DETECCIÓN DE PERSONAS

Para situar el sistema en su contexto, el primer paso consiste en identificar quién espera en el rellano. Para ello hemos elegido **YOLOv11-pose**, un modelo que combina en una sola pasada la detección de instancias ("¿hay personas y dónde están?") y la estimación de pose ("¿cómo está orientada cada persona?"). Este enfoque nos permite extraer, de un mismo fotograma, tanto las cajas delimitadoras que engloban a cada individuo como los keypoints esenciales (ojos, cintura, hombros, etc.) necesarios para los módulos posteriores de decisión de intención.

El entrenamiento de este detector se basó en un corpus de casi 8 000 imágenes reales obtenidas de distintos repositorios de cámaras de seguridad en rellanos. Elegimos escenarios variados, diferentes horas del día, condiciones de luz y ángulos de cámara, para dotar al modelo de la robustez necesaria contra oclusiones, personas superpuestas y contrastes extremos. Con ello, aseguramos que el detector funcione con alta precisión incluso en situaciones adversas.

Para alimentar el entrenamiento, cada imagen viene acompañada de un fichero TXT con el formato estándar de YOLO (clase x_centro y_centro ancho alto normalizados). La clase "0" corresponde siempre a "persona", y los valores de posición están adaptados a una resolución base de 640 × 640 px. Reservamos cerca del 10 % de estas imágenes para validación y un 7 % adicional para pruebas finales, garantizando así una evaluación realista de la capacidad de generalización.

Gracias a la estimación de pose, podemos a continuación aplicar reglas simples de lógica geométrica para decidir al vuelo si cada persona muestra indicios de querer subir al ascensor. Por ejemplo, si ambos ojos y al menos un punto de la cintura aparecen en la zona media o baja de la imagen, entendemos que hay intención de entrada; de no ser así, el sistema permanece en modo "ligero", con el detector de personas corriendo en bucle y sin invocar los clasificadores más pesados de género/edad o movilidad.

Este diseño modular, detección y pose en un único modelo, reglas de intención fáciles de validar y clasificación condicional, nos permite mantener latencias por fotograma por bajas y aprovechar al máximo los recursos de la **Raspberry Zero 2W** en despliegue edge, evitando el envío constante de toda la información al servidor y reduciendo el consumo energético.

5.3 DETECCIÓN DE OBJETOS DE MOVILIDAD REDUCIDA

Para garantizar un trato inclusivo y adaptado a usuarios con necesidades especiales, este módulo se encarga de identificar dispositivos de movilidad reducida, sillas de ruedas, andadores y muletas. Su activación sólo se produce cuando el bloque de intención de entrada confirma que el usuario planea subir al ascensor, de modo que evitamos invertir cómputo en simples pasos o cruces por el rellano. Al delegar esta tarea a un modelo **YOLOv11** especializado, buscamos un equilibrio entre precisión y velocidad que permita, en tiempo real, extender paradas o activar ayudas adicionales sólo cuando sea necesario.

En cuanto al origen de los datos, recopilamos varios miles de imágenes que reflejan la enorme diversidad de situaciones reales: interiores de hospitales y residencias, exteriores de calles y parques, iluminación diurna, nocturna y artificial, y ángulos cenitales u oblicuos. Este corpus se organizó en particiones de entrenamiento, validación y prueba, de forma similar a los demás módulos, garantizando que cada clase (andador, muletas, silla de ruedas) disponga de ejemplos suficientes para el aprendizaje y la evaluación de generalización.

Para el entrenamiento, cada instancia va anotada en un fichero TXT con el formato estándar de YOLO (<clase> <x_centro> <y_centro> <ancho> <alto> normalizados). Las clases se codifican numéricamente (0 = andador, 1 = muletas, 2 = silla de ruedas) y las coordenadas se ajustan a la resolución de entrada del modelo. Gracias a este esquema, el modelo aprende no solo a localizar el dispositivo, sino también a distinguirlo del propio cuerpo humano y de otros objetos vecinos, aportando este diseño aporta varias ventajas:

- **Eficiencia condicional**: sólo se activa bajo demanda, reduciendo drásticamente el uso de GPU/CPU en escenas sin intención.
- **Precisión especializada**: un modelo dedicado consigue buenos resultados aún en condiciones de oclusión parcial o iluminación cambiante.
- **Modularidad**: cualquier futura incorporación de nuevas clases (por ejemplo, bastones o andadores eléctricos) se resuelve con un nuevo fichero de etiquetas y unas cuantas imágenes de ejemplo, sin alterar la arquitectura general.

Con este enfoque, equilibramos accesibilidad y rendimiento, proporcionando información crítica al sistema de control sin sacrificar la fluidez ni la capacidad de respuesta del sistema.

5.4 ANÁLISIS DE RANGO DE EDAD Y GÉNERO

Para dotar al sistema de un perfil demográfico que vaya más allá del simple recuento de personas, este módulo se encarga de estimar tanto el género como un rango de edad aproximado para cada individuo detectado. Al operar exclusivamente sobre la región de interés (la bounding box) que devuelve el detector de personas, evitamos la re-localización de la figura en cada imagen y centramos todos los esfuerzos de cómputo en la tarea de clasificación pura. Esta separación de responsabilidades no solo simplifica el diseño, sino que además favorece el rendimiento en dispositivos edge con recursos limitados.

El conjunto de datos se organizó en diez carpetas, una por cada combinación de género (“male”/“female”) y rango de edad (Less15, 15–30, 30–45, 45–60, 60plus). Cada carpeta recopila exclusivamente las imágenes recortadas según la bounding box calculada previamente, lo que elimina la necesidad de anotaciones manuales adicionales o archivos de etiquetas complejos. De este modo, la etiqueta se asocia de forma automática y certera a partir del nombre de la carpeta, acelerando el flujo de trabajo y reduciendo la posibilidad de errores en el etiquetado.

Durante la fase de diseño, decidimos condicionar la ejecución de este módulo a la estimación de intención de entrada. Solo cuando el usuario muestra señales de querer subir al ascensor, el clasificador recibe el recorte y lo procesa. Así, en momentos de paso o espera trasversal, el sistema permanece en “modo liviano”, con únicamente el detector de personas en ejecución, garantizando un consumo energético y una latencia mínimos.

Entre los principales desafíos figuró la variabilidad de la silueta humana: ropa voluminosa, grupos parciales y ángulos incómodos podían alterar la proporción de hombros y cadera, confundiendo al clasificador.

Gracias a esta arquitectura modular, detección de personas, decisión de intención y clasificación demográfica, conseguimos un sistema escalable y fácil de mantener, añadir nuevos rangos de edad, subdividir géneros o incluir otras categorías demográficas es tan sencillo como crear una carpeta con ejemplos y reentrenar el clasificador, sin tocar el detector principal ni reconfigurar el pipeline de inferencia.

5.5 INTENCIONALIDAD DEL USUARIO

Para determinar si una persona tiene la intención de entrar al ascensor, aprovechamos los keypoints que ya devuelve el modelo **YOLOv11-pose** en la etapa de detección de personas. En lugar de recurrir a un clasificador adicional, aplicamos un sencillo conjunto de reglas lógicas sobre la posición de los ojos y la cintura en relación con tres franjas horizontales de la imagen:

- Si ambos ojos y al menos uno de los keypoints de la cintura aparecen en la franja media o baja, interpretamos que la persona está orientada hacia la puerta y desea entrar.
- Si solo se detecta uno de los ojos, pero la cabeza (keypoints de ojos o cabeza) está en la franja baja, también consideramos que existe intención de entrada.

Estas reglas se calculan en cada fotograma, inmediatamente después de la detección, y antes de invocar los modelos de clasificación de género/edad y de dispositivos de movilidad. De este modo, el sistema permanece en estado "ligero" (solo detección) cuando no hay intención, y cambia a modo "completo" (clasificaciones adicionales) únicamente cuando es necesario.

5.6 FLUJO DETALLADO DEL PIPELINE DE INFERENCIA: UN BALLET TECNOLÓGICO

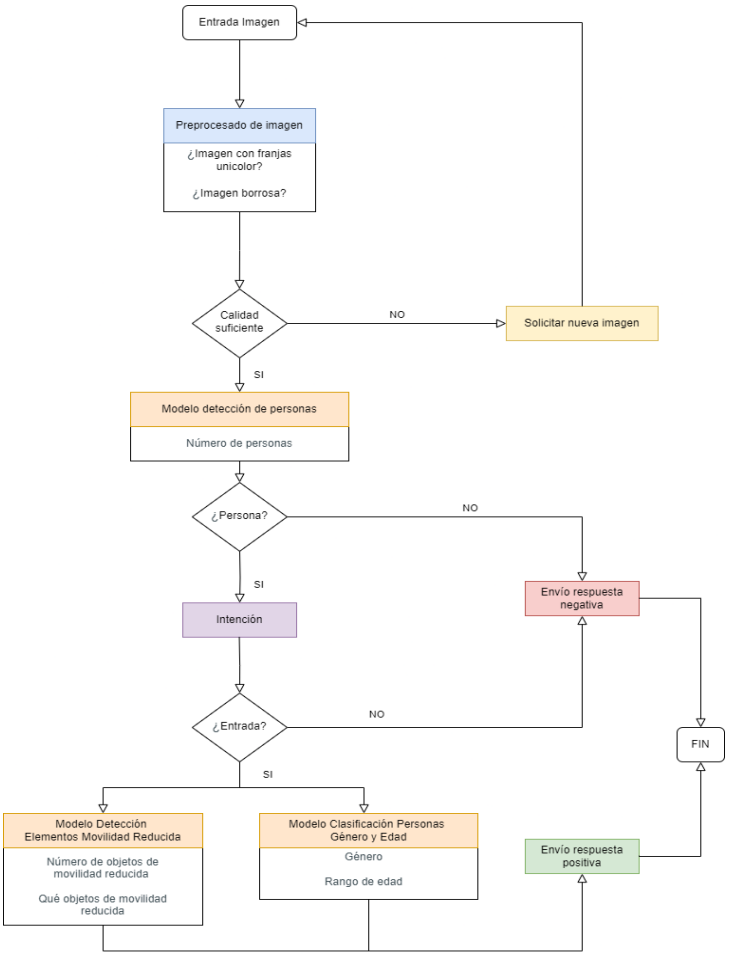


Ilustración 3: Pipeline con todos los modelos

1. **Captura de imagen**: Una cámara ubicada en el dintel provee continuos fotogramas.
2. **Detección y pose**: El modelo YOLOv11-pose procesa cada fotograma, devolviendo bounding boxes y keypoints para todas las personas presentes.
3. **Estimación de intención**: Con las reglas de posición de ojos y cintura, decidimos si cada persona planea entrar.
4. **Clasificaciones condicionales**: Solo para quienes muestran intención, ya que:
    - Se recorta la región de la persona y se envía al clasificador de género/edad.
    - De modo similar, se recorta y pasa al detector de objetos de movilidad reducida.
5. **Agregación y comunicación**: Se consolidan los resultados (número de usuarios, perfiles, dispositivos, intención) en un JSON y se envían vía API REST al sistema de control del ascensor.
6. **Bucle continuo**: El proceso se repite para cada fotograma, gestionando colas de inferencia y reenviando solo los cambios significativos (por ejemplo, nueva persona o cambio de intención) para minimizar tráfico de red.

Este pipeline, orquestado en un script de PyTorch, asegura latencias combinadas bajas manteniendo la robustez incluso en escenas con múltiples usuarios.

5.7 RAZONES DETRÁS DE LAS ELECCIONES TECNOLÓGICAS

La elección de a arquitectura YOLOv11-pose se debe en gran medida a la centralización en un solo paso tanto la detección de personas como la estimación de pose, lo que simplifica enormemente el pipeline y minimiza la latencia de inferencia.

- Multi-tarea en un único modelo: bounding boxes + keypoints de articulaciones.
- Equilibrio óptimo entre precisión y velocidad.
- Soporte nativo en la librería Ultralytics, con actualizaciones y ejemplos listos para usar.

¿Por qué no Faster R-CNN o SSD?

Faster R-CNN penaliza el tiempo de respuesta por ser un detector en dos etapas, y SSD sacrifica precisión en entornos con oclusión y múltiples poses.

La elección de PyTorch para el desarrollo de la inteligencia artificial se basa en las siguientes consideraciones.

- **Flexibilidad y dinamismo**: PyTorch se destaca por su estructura dinámica y flexible, lo que facilita la experimentación y la iteración durante el desarrollo del modelo de aprendizaje profundo. Esto es especialmente relevante cuando se exploran diferentes arquitecturas y enfoques.

- **Comunidad activa y documentación clara**: Cuenta con una comunidad activa y una documentación clara. Esto facilita la resolución de problemas y la adopción rápida de nuevas técnicas y enfoques en el campo del aprendizaje profundo.

¿Por qué no apoyarse de otros como TensorFlow y Keras?

Si bien TensorFlow y Keras son opciones brillantes y ampliamente utilizadas, PyTorch se ha destacado por su versatilidad y su capacidad de desarrollo a un nivel más bajo que Keras.

La elección del hardware de entrenamiento (RTX 4070 SUPER + Intel i5-13600KF) fue clave durante la fase de desarrollo y entrenamiento, ya que se necesitaba una plataforma con potencia de cómputo y flexibilidad:

- GPU muy eficiente en el uso de tensores, esencial para entrenar grandes batches y probar distintas arquitecturas.
- CPU de alto rendimiento para procesos de pre y post-procesado de datos (anotación, augmentación).
- Configuración de sobremesa que permite ajustes rápidos de drivers y herramientas de desarrollo.

¿Por qué no GPUs en la nube?

Si bien escalables, implican costes recurrentes elevados y dependencia de conectividad, complicando ciclos rápidos de prueba.

En la elección de hardware de despliegue (Raspberry Zero 2W + cámara integrada) para la fase real de operación en ascensores, se optó por una solución embebida compacta y de bajo consumo:

- Tamaño reducido y fácil montaje en el dintel del ascensor.
- Consumo energético muy bajo, ideal para edge inference 24/7.
- Compatibilidad con librerías ligeras como OpenCV y APIs de Python.

¿Por qué no NVIDIA Jetson?

Aunque ofrece más potencia, su coste y consumo resultan excesivos para la tarea de inferencia de un solo modelo ligero a 30 FPS.

6. IMPLEMENTACIÓN Y DESARROLLO

En esta sección profundizamos en la materialización de la arquitectura presentada. Para cada módulo describimos el entorno de desarrollo, los retos encontrados, las decisiones de diseño, los detalles de entrenamiento y los resultados más relevantes. El objetivo es ofrecer una visión exhaustiva de cómo pasamos del concepto a un sistema plenamente funcional.

6.1 DETECCIÓN DE PERSONAS

La detección de personas y la estimación de su pose forman el pilar sobre el que descansa todo el sistema. Este módulo no solo identifica la presencia de individuos en el rellano, sino que extrae información clave sobre su posición y postura, esencial para la estimación de intención de entrada. Al emplear YOLOv11-pose, buscamos unificar dos tareas en un único paso de inferencia, lo que reduce significativamente el coste computacional y minimiza las latencias. En esta fase se enfrenta el reto de procesar en tiempo real escenas con múltiples personas, variaciones lumínicas y ángulos complejos, garantizando al mismo tiempo una alta precisión en la localización de cajas delimitadoras y keypoints.

6.1.1 ENTORNO Y DEPENDENCIAS

- **Lenguaje y frameworks**: Python 3.10, PyTorch 2.x, librería Ultralytics YOLOv11-pose
- **Hardware**:
  - Fase de I+D: NVIDIA RTX 4070 SUPER, Intel i5-13600KF, 64 GB RAM
  - Despliegue final: Raspberry Zero 2W (Python + Putty) con cámara integrada

6.1.2 PREPARACIÓN DE DATOS

Recolectamos 7 698 imágenes de entrenamiento, 761 de validación y 585 de prueba extraídas de repositorios públicos de cámaras de seguridad en rellanos. Cada imagen va acompañada de un fichero TXT con líneas 0 x_centro y_centro ancho alto, donde los valores están normalizados a la resolución 640×640 px. Para asegurar calidad:

1. **Diversidad**: incluimos escenas con múltiples personas, ángulos contrapicados y contrapicados, y condiciones lumínicas extremas.
2. **Control de calidad**: revisión inter-annotador del 10 % del dataset, resolviendo discrepancias por consenso.

6.2 DETECCIÓN DE OBJETOS DE MOVILIDAD REDUCIDA

La detección de dispositivos de movilidad reducida es esencial para garantizar un servicio inclusivo y seguro. Este módulo identifica andadores, muletas y sillas de ruedas en personas con intención de entrada, activando paradas prolongadas y asistencias específicas. La variabilidad en el diseño y posición de estos elementos, así como las oclusiones parciales por el propio usuario, plantean un desafío de visión por ordenador que requiere un dataset amplio y técnicas de entrenamiento especializadas. Además, la detección está condicionada a la confirmación de intención de entrada, lo que demanda una orquestación cuidadosa para no generar latencias adicionales.

Para maximizar la sensibilidad y la precisión en escenarios reales, el dataset incorpora miles de ejemplos de cada clase, capturados bajo distintas condiciones de iluminación y ángulos de cámara.

6.2.1 ENTORNO Y DEPENDENCIAS

- **Lenguaje y frameworks**: Python 3.10, PyTorch 2.x, librería Ultralytics YOLOv11
- **Hardware**:

- Fase de I+D: NVIDIA RTX 4070 SUPER, Intel i5-13600KF, 64 GB RAM
- Despliegue final: Raspberry Zero 2W (Python + Putty) con cámara integrada

6.2.2 PREPARACIÓN Y RECOPILACIÓN DE DATOS

- **Fuentes y cantidad de imágenes:**

La base de un modelo de detección de objetos eficaz es un conjunto de datos extenso y variado. Iniciamos el proyecto recopilando alrededor de 10,000 imágenes de diferentes conjuntos de datos (datasets). Con el objetivo de mejorar la robustez y precisión del modelo, incrementamos este número a más de 15,000 imágenes. La distribución de estas imágenes es la siguiente:

- Andador: 6 005 train / 1 095 val / 201 test
- Muletas: 4 990 train / 1 190 val / 213 test
- Sillas de ruedas: 8 833 train / 1 290 val / 197 test

Hablando ahora de la procedencia de las imágenes:

- **Detección de sillas de ruedas**: obtenidas del dataset "Wheelchair detection" de Roboflow Universe, creado por 2458761304@qq.com, que contiene 514 imágenes etiquetadas de personas con sillas de ruedas universe.roboflow.com+10universe.roboflow.com+10universe.roboflow.com+10.
- **MobilityAids Dataset**: dataset hospitalario RGB-D con más de 17 000 imágenes anotadas, que incluye categorías como silla de ruedas, muletas, andador, peatón y persona empujando silla de ruedas. Recogido en la Universidad de Freiburg y en un hospital de Frankfurt ui.adsabs.harvard.edu+7paperswithcode.com+7academia.edu+7.
- **Crutch-image-classification dataset**: colección específica para muletas recopilada de imágenes CV, orientado a entrenamiento de clasificación de imagen de muletas se utilizó esta fuente para complementar la categoría de muletas. https://images.cv/dataset/crutch-image-classification-dataset

- **Diversidad de las imágenes:**

Para asegurar que el modelo sea capaz de generalizar en situaciones del mundo real, recopilamos imágenes que abarcan una amplia variedad de escenarios y contextos. Estas incluyen:

- **Interiores:** Imágenes tomadas en hospitales, residencias y otros edificios.
- **Exteriores:** Escenas capturadas en parques, calles y plazas.
- **Diferentes condiciones de iluminación:** Imágenes en condiciones de luz diurna, nocturna, y en interiores con diferentes tipos de iluminación artificial.
- **Ángulos de cámara:** Diversos ángulos para que el modelo aprenda a reconocer los objetos desde distintas perspectivas.

- **Procedimiento de recopilación:**

El proceso de recopilación de datos involucró la búsqueda en diversas fuentes, incluyendo bases de datos públicas, proyectos de código abierto y colaboración con instituciones que nos permitieron acceder a sus recursos visuales. Cada imagen fue cuidadosamente seleccionada y revisada para asegurarse de que cumpliera con los criterios de calidad necesarios. En esta etapa, se llevaron a cabo reuniones semanales para evaluar el progreso y ajustar la estrategia de recopilación según las necesidades emergentes del proyecto.

- **Limpieza y preparación de datos:**

Una vez recopiladas, las imágenes fueron sometidas a un proceso riguroso de limpieza y preparación. Esto incluyó la eliminación de imágenes duplicadas, el ajuste de la resolución para garantizar la calidad y la conversión a formatos adecuados para el entrenamiento del modelo. Además, se realizaron anotaciones manuales para identificar y etiquetar correctamente los objetos de interés en cada imagen. Este proceso fue fundamental para asegurar que el conjunto de datos estuviera listo para el siguiente paso: el entrenamiento del modelo.

6.3 CLASIFICACIÓN DE RANGO DE EDAD Y GÉNERO

El análisis de género y rango de edad es un componente crucial para personalizar la experiencia de uso del ascensor y satisfacer normativas de accesibilidad y seguridad. Este módulo no se limita a asignar etiquetas demográficas; se concibió para equilibrar precisión, velocidad y equidad en la clasificación de usuarios en entornos reales. La diversidad de rostros, posturas y vestimenta en los rellanos introduce variabilidad significativa que debe abordarse con técnicas de augmentación avanzadas y un diseño de modelo robusto. Este enfoque es especialmente útil en escenarios donde la detección facial no es viable, ya sea porque la cara no es visible o debido a la calidad del video.

Además, dado que esta clasificación solo se realiza cuando el sistema detecta intención de entrada, es fundamental que el clasificador opere de forma extremadamente eficiente en hardware de edge para no comprometer la latencia global.

El reto principal radica en distinguir diferencias sutiles entre rangos de edad contiguos y en evitar sesgos de género que puedan surgir por desequilibrios en el dataset. Para ello, organizamos las imágenes en diez categorías claramente definidas con datos reales de cámaras de seguridad, permitiendo capturar patrones de textura y forma asociados a cada rango demográfico.

6.3.1 ENTORNO Y DEPENDENCIAS

- **Lenguaje y frameworks**: Python 3.10, PyTorch 2.x, librería Ultralytics YOLOv11-cls
- **Hardware**:
  - Fase de I+D: NVIDIA RTX 4070 SUPER, Intel i5-13600KF, 64 GB RAM
  - Despliegue final: Raspberry Zero 2W (Python + Putty) con cámara integrada

6.3.2 IMPLEMENTACIÓN Y FUNCIONAMIENTO CON EL SISTEMA COMPLETO

El modelo final fue implementado para funcionar en conjunto con el modelo de detección de personas preexistente.

En este sistema, las bounding boxes generadas por el modelo de detección sirven como entrada para el modelo de clasificación de género y edad. Esto permite que el sistema clasifique de manera efectiva a las personas detectadas según los criterios especificados, incluso en condiciones donde el rostro no es visible o la calidad de la imagen es baja.

Ilustración 4: Imagen sin procesar

Ilustración 5: Persona detectada en la imagen

Ilustración 6: Hacemos hincapié en la caja facilitada

```
0: 384x640 1 person, 32.7ms
Speed: 3.1ms preprocess, 32.7ms inference, 87.6ms postprocess per image at shape (1, 3, 384, 640)

0: 224x224 Male_18-60 0.70, Female_18-60 0.27, Male_Larger60 0.02, Female_Larger60 0.01, Male_Less18 0.00, 2.5ms
Speed: 4.5ms preprocess, 2.5ms inference, 0.0ms postprocess per image at shape (1, 3, 224, 224)
```

Ilustración 7: Respuesta en bruto de los modelos ante la imagen

Ilustración 8: Imagen final con las predicciones dibujadas

6.3.2 PREPARACIÓN Y RECOPILACIÓN DE DATOS

- **PETA Dataset**

Para comenzar, seleccionamos el dataset **PETA (PEdesTrian Attribute)**, un conjunto de datos ampliamente utilizado en la investigación de análisis de atributos de personas. Este dataset es particularmente adecuado para nuestro propósito ya que:

PETA está compuesto por diversas subcategorías que contienen imágenes de personas capturadas en diferentes escenarios, principalmente provenientes de cámaras de seguridad. Estas imágenes están etiquetadas con varios atributos, incluyendo género y diferentes rangos de edad. Los rangos de edad en PETA están categorizados como:

- Menor de 15 años
- Menor de 30 años
- Menor de 45 años
- Menor de 60 años
- Mayor de 60 años

La variedad en los rangos de edad permite una clasificación más detallada, aunque presenta limitaciones en términos de volumen de datos.

La principal ventaja de PETA es la diversidad en la categorización de edades, lo que ofrece una visión más granular en la clasificación. Sin embargo, la desventaja radica en su tamaño limitado, con aproximadamente 19.000 imágenes, lo que afecta la precisión del modelo, especialmente en la clasificación de rangos de edad menos representados.

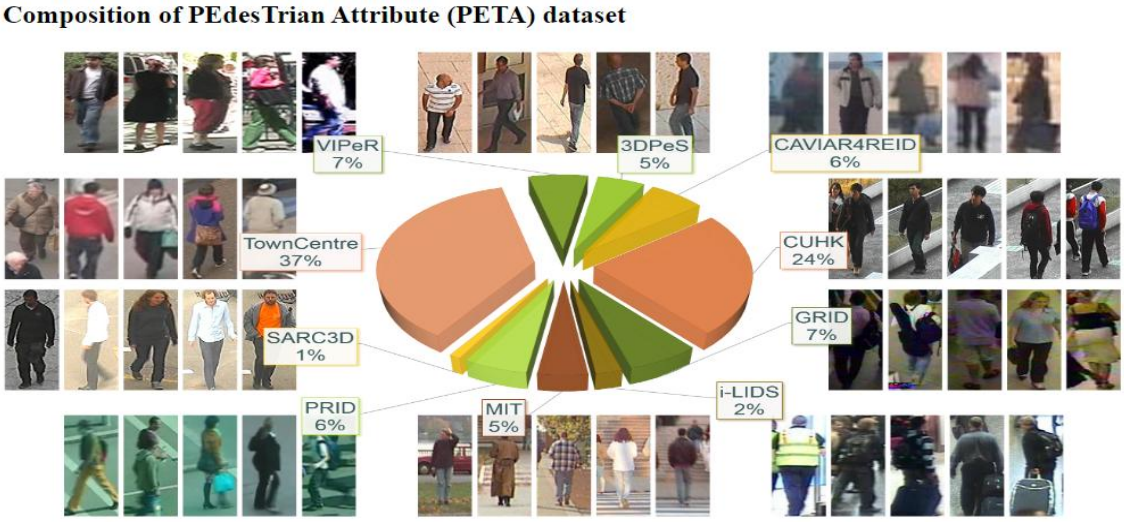


Ilustración 9: Composición del dataset PETA

- **Adaptación del dataset**

El primer paso en el uso de PETA fue adaptar las etiquetas de género y edad para nuestro caso de uso. Dado que el modelo de detección previa nos proporciona las bounding boxes de las personas detectadas, utilizamos estas áreas de imagen como entrada para nuestro nuevo modelo de clasificación.

Las etiquetas originales de PETA fueron simplificadas y reagrupadas para facilitar el entrenamiento y mejorar la precisión en las categorías más relevantes para nuestro proyecto: género (masculino/femenino) y edad en los rangos especificados por el dataset.

6.4 ESTIMACIÓN DE INTENCIÓN DE ENTRADA

Detectar la intención de entrada de un usuario a través de su pose es un elemento innovador que permite optimizar recursos y reducir la carga de procesamiento. En lugar de entrenar un modelo adicional, aprovechamos los keypoints de pose obtenidos en el módulo de detección de personas para aplicar reglas lógicas que reflejan comportamientos humanos reales, como la orientación del torso y el movimiento de la cadera. Este enfoque híbrido, que combina visión por ordenador con conocimiento heurístico, aporta una solución eficiente y explicable.

El principal desafío es diseñar reglas lo suficientemente generales para funcionar con distintos tipos de usuarios (niños, adultos, personas con ropa voluminosa) y condiciones de iluminación variables. Realizamos un estudio antropométrico preliminar para definir zonas de interés (tercios de la imagen) y validamos manualmente cientos de ejemplos para calibrar los umbrales de detección. El resultado es un sistema de intención con alta fiabilidad y baja tasa de falsos positivos, fundamental para escalar el pipeline únicamente cuando sea necesario.

6.4.1 REGLAS

- Si no se le detectan ambos ojos se entiende que la persona se encuentra de paso (de izquierda a derecha de la imagen o viceversa) o dicha persona está de espaldas.

- Si se le detectan ambos ojos, pero la cadera (y por ende la cabeza) se encuentra en la parte superior de la imagen afirmamos que es alguien parado demasiado lejos del ascensor como para querer entrar

- Si se le detectan ambos ojos, pero la cadera o la cabeza se encuentran en la zona media de la imagen podemos confirmar que esa persona cumple con los requisitos como para querer entrar.

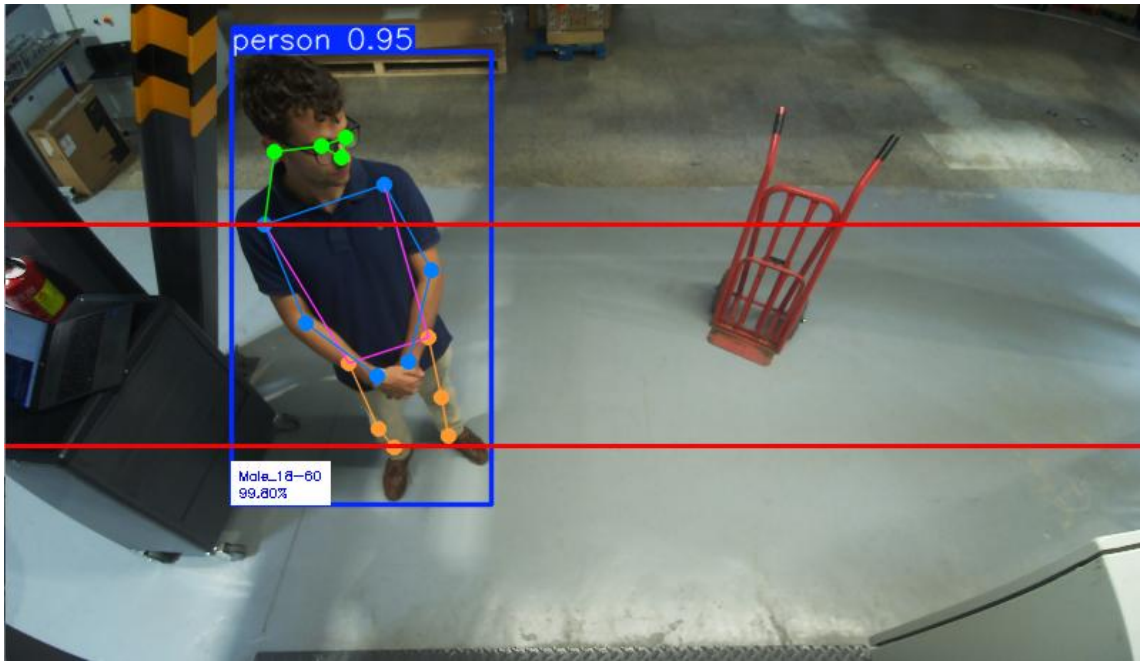


Ilustración 10: Imagen con la pose y estimación de intención (véase que se procesa la edad al cumplir las reglas de intención)

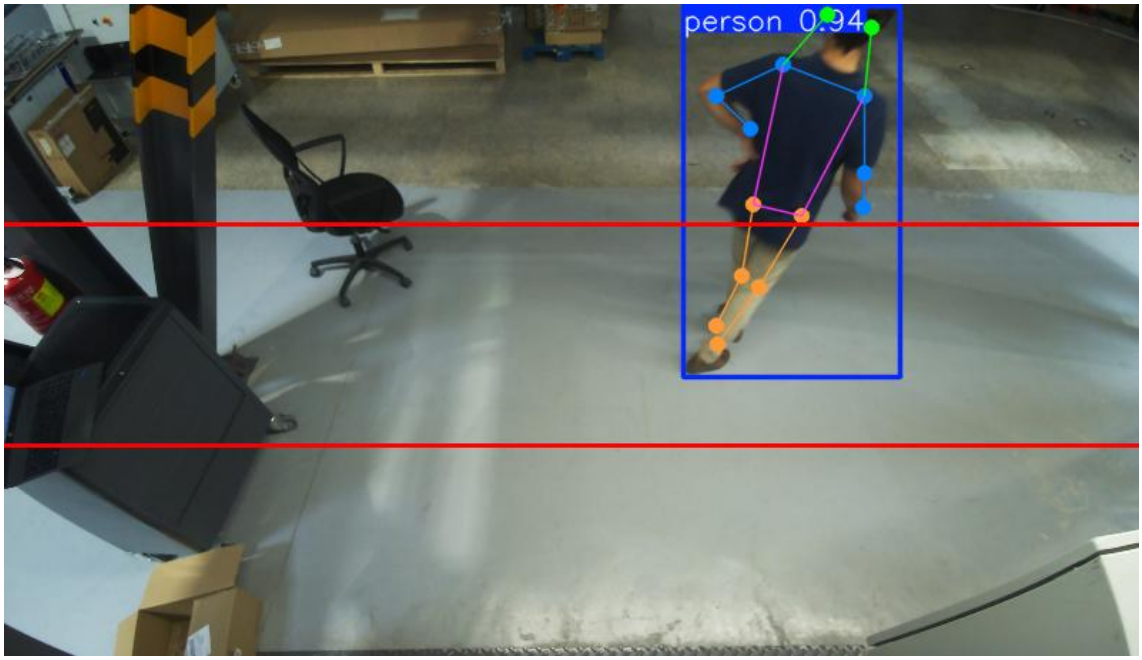


Ilustración 11: Imagen con la pose y estimación de intención (véase que no se procesa la edad al no cumplir las reglas de intención)

6.6 DESARROLLO, DIFICULTADES Y MEJORA ITERATIVA DE MODELOS

A) MODELO DE DETECCIÓN DE PERSONAS

La integración de YOLOv11-pose en nuestro sistema planteó un abanico de retos más allá de su rendimiento "en laboratorio" ya que en condiciones de prueba surgieron nuevos problemas no planeados inicialmente.

En primer lugar, en escenas con varias personas solapadas, el sistema inicial tendía a mezclar poses y cajas, generando keypoints incoherentes (por ejemplo, cintura de un sujeto señalada dentro del recorte del vecino). Resolverlo exigió modificar el ángulo en el que se encontraba ajustada en un principio la cámara en el dintel del ascensor.

El cálculo de keypoints bajo condiciones de oclusión parcial, manos sobre caderas, chaquetas voluminosas u objetos en primer plano, mostró que el peso estándar que YOLOv11-pose asignaba a cada articulación no era óptimo para nuestro caso de uso. Tuvimos que modificar el balance de la función de pérdida para enfatizar la detección de ojos y hombros frente a codos o tobillos, pues eran estos puntos los que realmente nos servían para inferir intención de entrada.

Por último, el despliegue en la Raspberry Zero 2W desveló limitaciones de memoria en la versión estándar de la librería Ultralytics: la carga de los pesos y la inicialización del modelo encarecían la primera inferencia más de lo aceptable. La solución pasó por usar el modelo más liviano posible dentro de la gama de YOLOv11-pose en este caso el modelo nano, ya que este sí estaba pensado para esta clase de dispositivos con carencias de memoria.

En conjunto, estos ajustes, desde la optimización de peso de keypoints hasta la adaptación a hardware Edge, fueron clave para que YOLOv11-pose pasara de ser un prototipo prometedor a un componente estable y fiable dentro de nuestra tubería de visión.

B) MODELO DE DETECCIÓN DE OBJETOS DE MOVILIDAD REDUCIDA

I. ¿QUÉ ES UN HIPERPARÁMETRO?

En el aprendizaje automático, un hiperparámetro es un valor predefinido que controla el proceso de entrenamiento del modelo. Estos parámetros no se aprenden a partir de los datos; en cambio, son establecidos antes de que comience el entrenamiento y tienen un impacto significativo en el rendimiento del modelo. Los hiperparámetros clave que ajustamos durante este proyecto incluyen:

- **Épocas (epochs):** El número de veces que el algoritmo de aprendizaje pasa por el conjunto de entrenamiento completo. Un número mayor de épocas permite que el modelo aprenda más de los datos, pero también puede llevar al sobreajuste si es excesivo.
- **Tamaño del lote (batch size):** El número de muestras que se procesan antes de actualizar el modelo. Un tamaño de lote mayor puede acelerar el entrenamiento, pero requiere más memoria.
- **Tasa de aprendizaje (learning rate, lr):** El tamaño del paso que el algoritmo de optimización toma en cada iteración. Una tasa de aprendizaje alta puede hacer que el modelo converja más rápido, pero también puede causar inestabilidad.
- **Descomposición del peso (weight decay, wd):** Una técnica de regularización utilizada para prevenir el sobreajuste, añadiendo un término de penalización al costo del modelo.

II. AJUSTE DE HIPERPARÁMETROS

Inicialmente, experimentamos con diversas configuraciones de estos hiperparámetros para identificar una combinación óptima que proporcionara buenos resultados. Durante esta

fase, realizamos ajustes incrementales y observamos el impacto de cada cambio en el rendimiento del modelo. Algunos de los desafíos encontrados incluyeron:

- **Sobreajuste:** Donde el modelo performa bien en los datos de entrenamiento, pero mal en los datos de validación.
- **Bajo rendimiento en imágenes no perfiladas:** El modelo tenía dificultades para detectar sillas de ruedas que no estaban perfectamente perfiladas a la cámara.

Para mitigar estos problemas, utilizamos una estrategia de ajuste fino (fine-tuning), donde el modelo preentrenado en un conjunto de datos grande y diverso fue ajustado con nuestro conjunto de datos específico. Esto ayudó a mejorar la capacidad del modelo para generalizar en nuevas imágenes y redujo el impacto del sobreajuste.

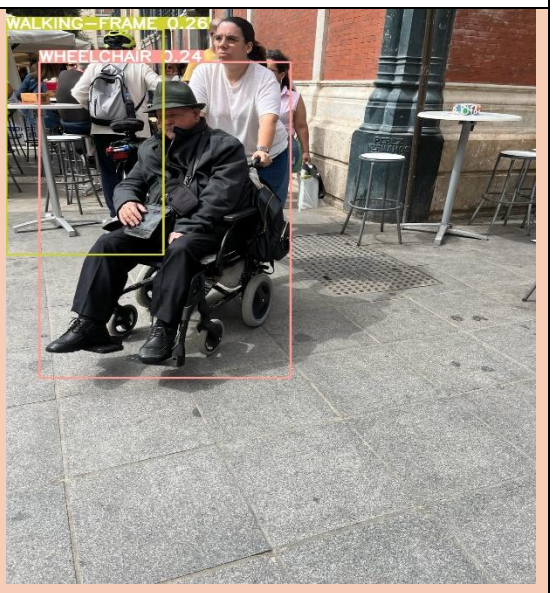
III. ITERACIONES Y PRUEBAS

Realizamos múltiples iteraciones de prueba y ajuste, cada una de ellas con variaciones en los hiperparámetros. Cada iteración fue seguida de una evaluación rigurosa para medir su rendimiento en un conjunto de datos de validación. Utilizamos métricas como la precisión, la sensibilidad y el puntaje F1 para evaluar el rendimiento del modelo y hacer ajustes informados en los hiperparámetros. Este proceso iterativo continuó hasta que alcanzamos un equilibrio óptimo entre la precisión y la capacidad de generalización del modelo.

III.A DIFICULTADES INICIALES

La selección de los hiperparámetros fue un proceso iterativo y laborioso. Probamos diferentes combinaciones de parámetros como el tamaño del lote (batch size), el número de épocas, la tasa de aprendizaje, el optimizador, y el decaimiento de peso.La tabla de resultados obtenida muestra una variedad de configuraciones que fueron evaluadas durante este proceso.

En las primeras fases, no contábamos con imágenes de entornos reales para testar el modelo ya que aún no se pudo gestionar pruebas en entornos reales. Esto nos obligó a utilizar un conjunto de datos de prueba compuesto por imágenes de diversos entornos y condiciones ambientales. Aunque este dataset nos permitió probar el modelo en diferentes escenarios, no reflejaba completamente las condiciones reales a las que se enfrentaría el modelo.

DATASET DE PRUEBAS INICIAL	RESULTADOS OPTIMISTAS INICIALES
	
	
	



WHEELCHAIR 0.54


CRUTCHES 0.74


WALKING-FRAME 0.89
WALKING-FRAME 0.87

III.B MEJORAS Y RESULTADOS

Tras varias rondas de pruebas y análisis, decidimos implementar cambios significativos en la configuración del modelo. Estos cambios incluyeron ajustes precisos en los hiperparámetros, la ampliación del dataset de testeo, y la inclusión de nuevas técnicas de optimización.

- Ajustes de hiperparámetros

Se realizaron ajustes en la tasa de aprendizaje y en el optimizador utilizado, lo que resultó en una mejora notable en la precisión del modelo.

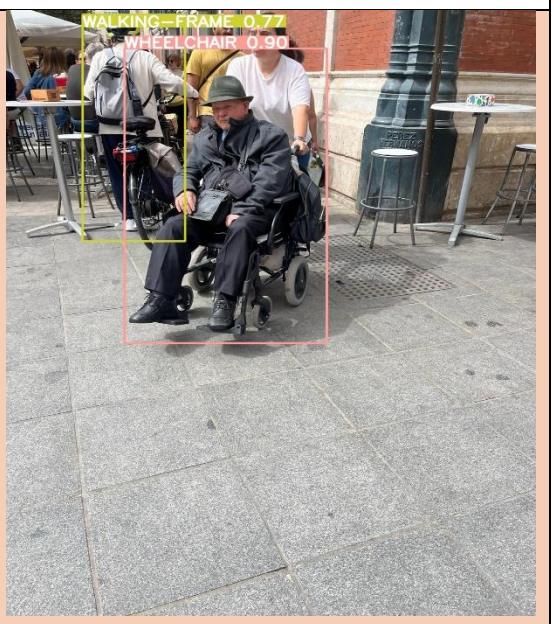
Incrementamos el número de épocas de entrenamiento y ajustamos el tamaño del lote para maximizar la eficiencia del entrenamiento sin comprometer la estabilidad del modelo.

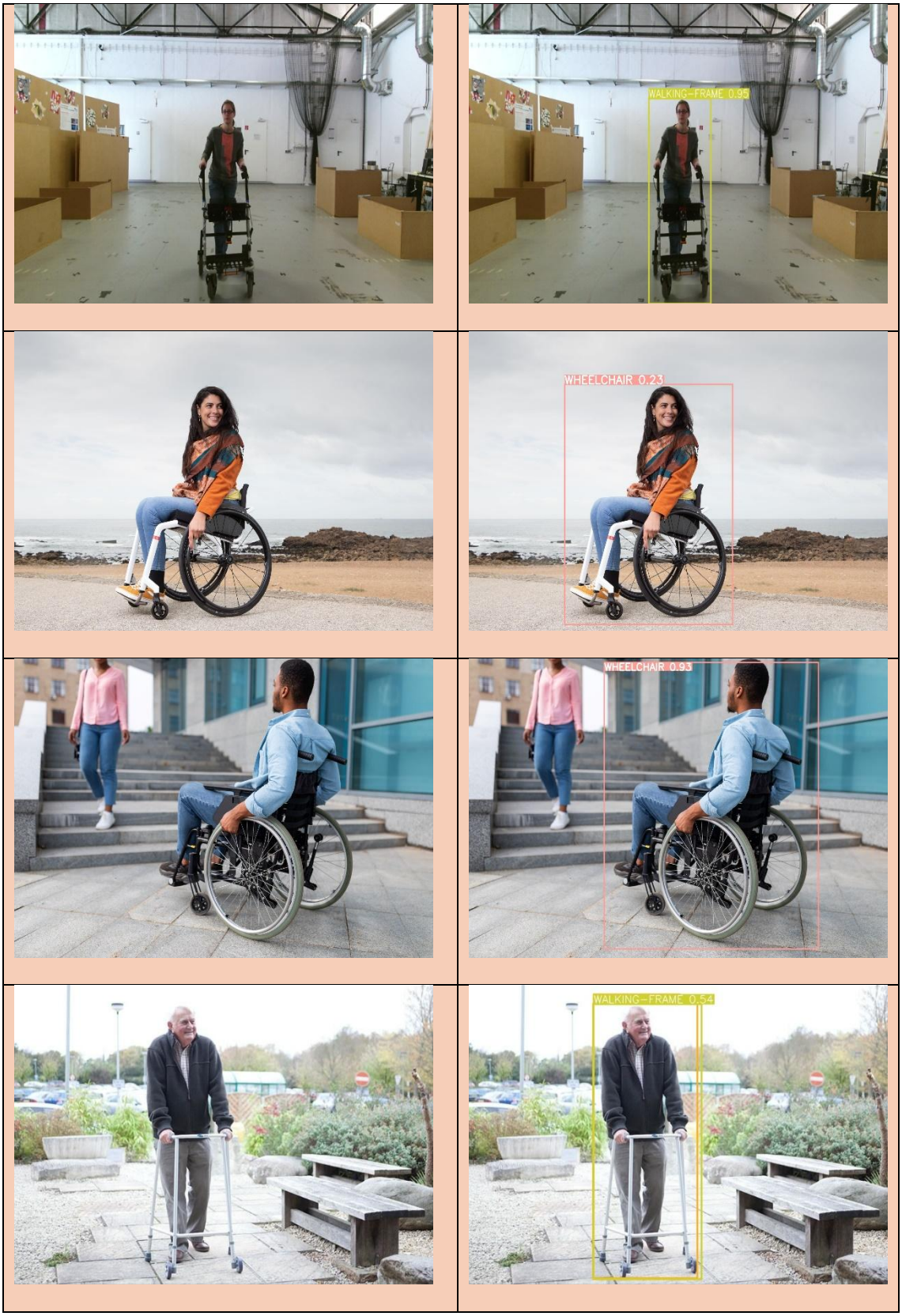
- Ampliación del dataset de testeo

Decidimos aumentar el tamaño del dataset de testeo para obtener mejores conclusiones sobre el rendimiento del modelo. Esto nos permitió evaluar de manera más robusta y precisa las diferentes configuraciones de hiperparámetros.

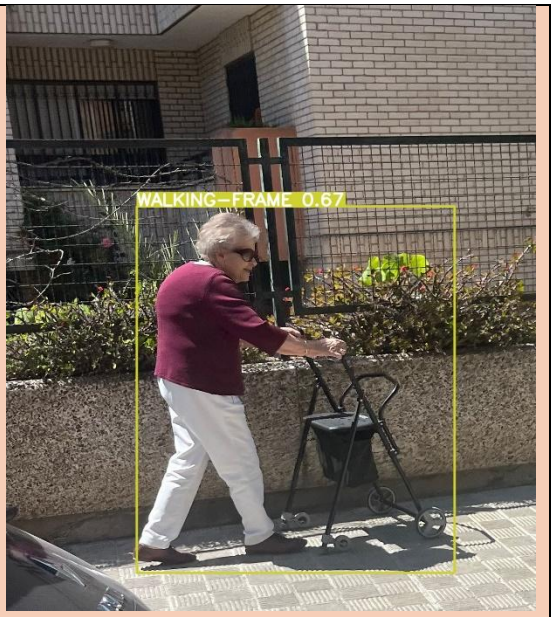
La ampliación del dataset de testeo fue crucial para identificar las combinaciones de hiperparámetros que ofrecían un mejor desempeño y para tener una base más sólida al momento de hacer ajustes adicionales.

Véase ahora que las celdas que tienen imágenes del dataset inicial tienen un fondo anaranjado y las imágenes incluidas en el dataset tiene un fondo azulado con el objetivo de ir viendo las iteraciones de forma visual.

DATASET DE PRUEBAS MEJORADO	RESULTADOS MODELO MEJORADO
	


WALKING-FRAME 0.95
WHEELCHAIR 0.23
WHEELCHAIR 0.93
WALKING-FRAME 0.54


WALKING-FRAME 0.98
WALKING-FRAME 0.87
WALKING-FRAME 0.84
CRUTCHES 0.42
CRUTCHES 0.97

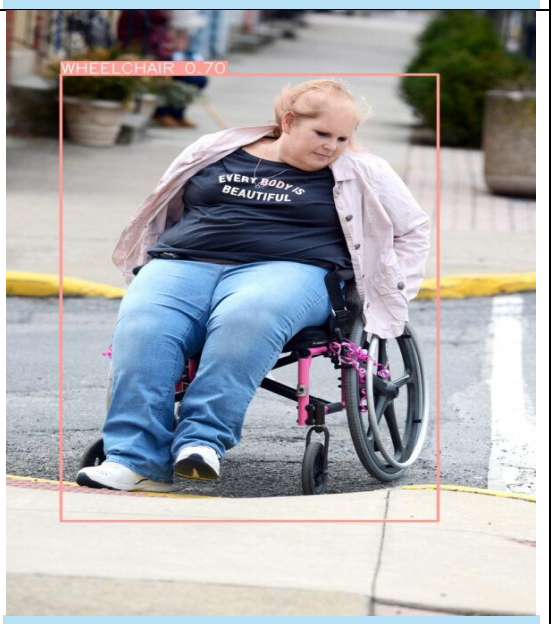

WALKING—FRAME 0.67

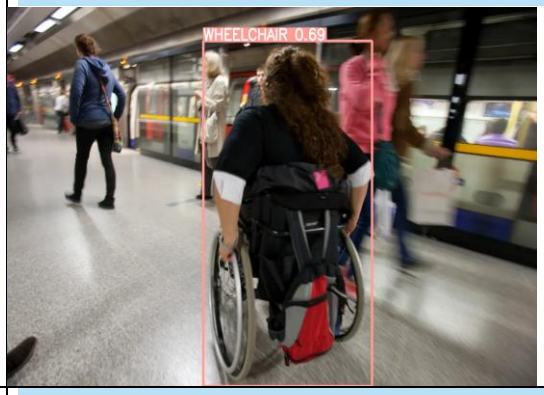

WHEELCHAIR 0.66


WHEELCHAIR 0.84


WHEELCHAIR 0.74


EVERY BODY IS BEAUTIFUL


WHEELCHAIR 0.70
EVERY BODY IS BEAUTIFUL

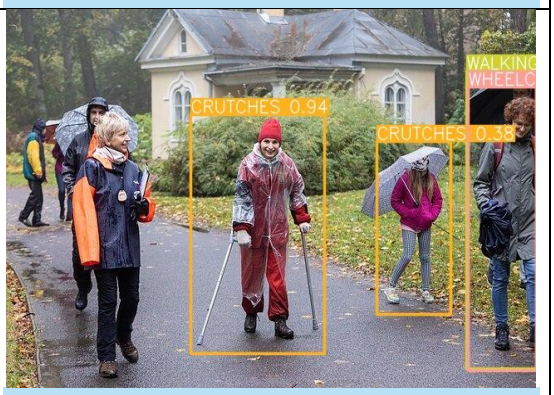

WHEELCHAIR 0.69


WALKING_FRA CRUTCHES 0.89
CRUTCHES 0.74



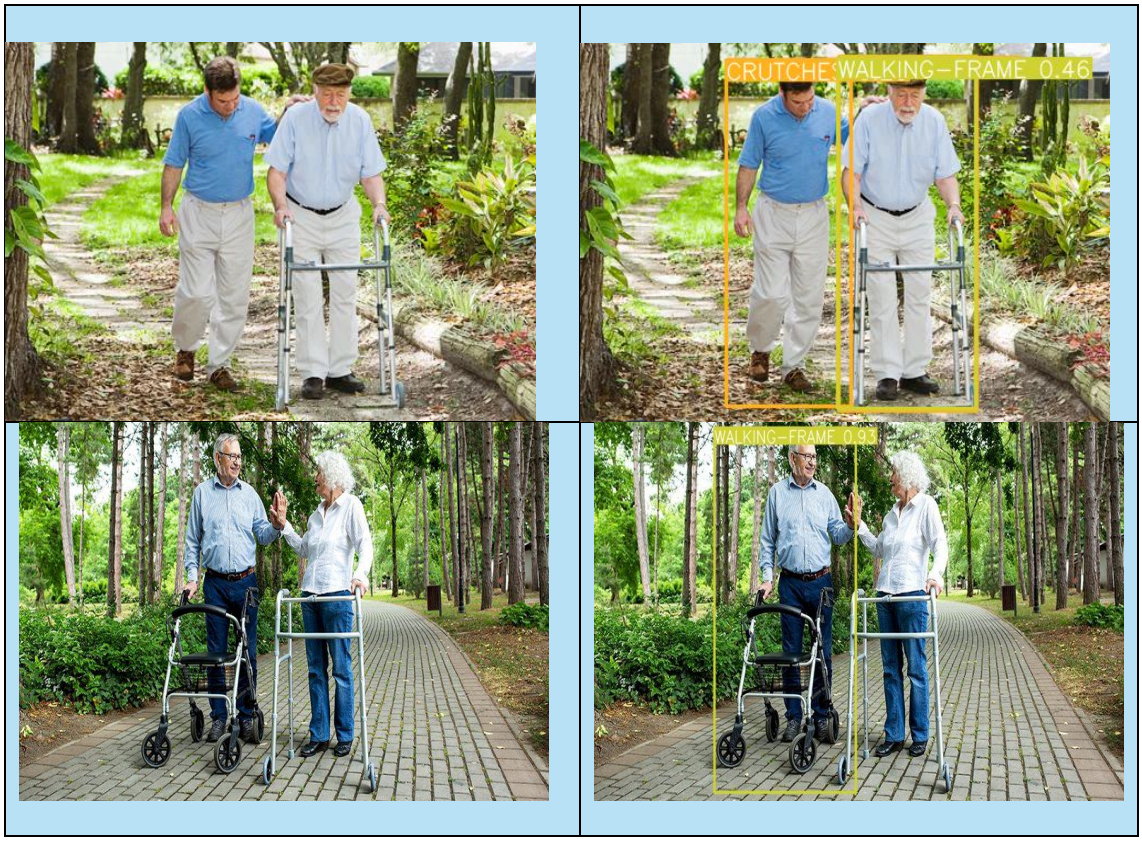
CRUTCHES 0.71


CRUTCHES 0.99


CRUTCHES 0.94
CRUTCHES 0.38


CRUTCHES 0.27
CRUTCHES 0.40


WALKING–FRAME 0.80
WALKING–FRAME 0.90
WALKING–FRAME 0.71
WALKING–FRAME 0.75



III.C INCORPORACIÓN DE IMÁGENES REALES

Una vez que logramos obtener imágenes de entornos reales gracias a las pruebas del sistema y metodología que serán explicadas más adelante, pudimos probar y ajustar nuestro modelo en estas condiciones. Este cambio fue crucial, ya que permitió al modelo aprender características más relevantes y específicas de los entornos reales.

Los resultados mejoraron significativamente al evaluar el modelo en las nuevas imágenes reales, mostrando una mayor precisión y robustez.

DATASET DE PRUEBAS REAL	RESULTADOS INICIALES
	



WHEELCHAIR 0.75


WALKING-FRAME 0.47
WHEELCHAIR 0.22


WALKING-FRAME 0.37


WALKING-FRAME 0.53
WHEELCHAIR 0.29

III.D MIGRACIÓN A NUEVAS VERSIONES DE YOLO Y USO DE IMÁGENES REALES AL ENTRENAMIENTO DEL MODELO

Tras obtener las imágenes reales, decidimos cambiar nuestra arquitectura de modelo a nuevas versiones de YOLO, terminando hasta en YOLOv11. Este cambio se realizó debido a las capacidades mejoradas de YOLOv11 en términos de precisión y eficiencia.

La actualización de YOLO permitió al modelo aprovechar mejor las características de los nuevos datos y mejorar aún más su desempeño en tareas de detección de objetos.

Con el objetivo de mejorar el rendimiento del modelo, decidimos dividir el conjunto de imágenes reales en dos, más de la mitad de las imágenes se destinarían a alimentar el modelo y el resto (las mostradas anteriormente, se emplearían para probar dicho modelo).

III.E MEJORA DEL CONJUNTO DE DATOS

- Identificación de deficiencias

Un análisis más detallado reveló que la principal deficiencia residía en la detección de sillas de ruedas que no estaban perfectamente perfiladas. Este problema se presentaba con mayor frecuencia en imágenes donde las sillas de ruedas aparecían en ángulos oblicuos o parcialmente obstruidas.

En los ejemplos inferiores, se aprecia de forma clara como cuando una silla se encuentra en una posición sencilla de apreciar (primer ejemplo) el modelo era capaz de detectarla de forma sólida, no obstante, si esa silla se encontraba en un ángulo oblicuo o difícil de apreciar, el modelo no lo detectaba como silla de ruedas (segundo ejemplo) .

- Adición de imágenes

Para abordar esta deficiencia, añadimos nuevas imágenes específicas de sillas de ruedas capturadas desde diferentes ángulos y en variados contextos. La inclusión de estas imágenes adicionales fue crítica para proporcionar al modelo la diversidad necesaria para mejorar su capacidad de detección.

- Procedimiento de anotación

Las nuevas imágenes añadidas al conjunto de datos fueron anotadas cuidadosamente para garantizar que los objetos de interés estuvieran etiquetados con precisión. Este proceso de anotación fue realizado por un equipo de expertos que revisaron cada imagen y aplicaron etiquetas consistentes. Además, utilizamos herramientas de anotación semiautomática para acelerar el proceso y asegurar la uniformidad en las etiquetas.

- Resultados de la mejora

Aunque esta adición mejoró los resultados, todavía existían casos en los que la precisión no era suficiente. Nos dimos cuenta de que, además de aumentar la cantidad de imágenes, también era necesario optimizar la calidad de las etiquetas de los datos.

- Validación continua

Durante esta fase, implementamos un proceso de validación continua donde cada conjunto de datos actualizado era evaluado en tiempo real. Utilizamos técnicas de validación cruzada para asegurarnos de que las mejoras no fueran específicas a un subconjunto de datos, sino que beneficiaran al modelo en general. Este enfoque nos permitió identificar y corregir rápidamente cualquier inconsistencia en las etiquetas o problemas de sobreajuste.

Como enfoque de validación cruzada hemos decidido, en este caso, dividir el conjunto de datos en un conjunto de entrenamiento y un conjunto de validación antes del inicio del entrenamiento. Este conjunto de validación fijo se utiliza para evaluar el rendimiento del modelo después de cada época de entrenamiento. Esta metodología se asemeja a un enfoque de validación cruzada k-Fold con k=1, donde solo hay una partición entre entrenamiento y validación.

III.F USO DE WEIGHTS AND BIASES (W&B)

- ¿Qué es Weights and Biases?

Weights and Biases (W&B) es una herramienta avanzada que facilita el seguimiento y visualización de experimentos de aprendizaje automático. Ofrece funcionalidades que permiten registrar cada ejecución de entrenamiento, comparar diferentes configuraciones y visualizar métricas en tiempo real. W&B es especialmente útil para estudios de hiperparámetros, permitiendo identificar rápidamente las configuraciones que mejoran el rendimiento del modelo.

- ❖ Estudio de hiperparámetros con W&B

Decidimos utilizar W&B para realizar un estudio exhaustivo de hiperparámetros. Esta herramienta nos permitió:

- **Registrar cada experimento:** Manteniendo un historial detallado de cada configuración de hiperparámetros y sus resultados. Esto nos permitió tener un registro claro y estructurado de cada intento y sus efectos en el modelo.
- **Visualizar métricas en tiempo real:** Facilitando la identificación de patrones y tendencias en el rendimiento del modelo, permitiendo ajustes más informados y precisos.
- **Comparar experimentos:** Permitiendo una comparación directa de diferentes configuraciones y sus impactos en la precisión y eficiencia del modelo.

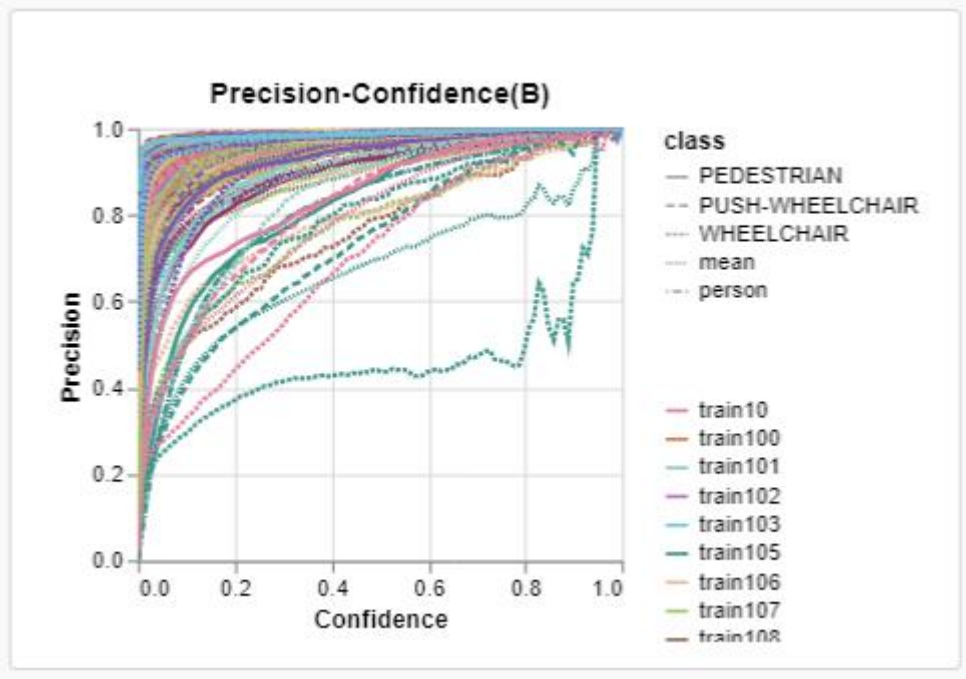


Ilustración 12: Métrica de precisión de los modelos durante los más de 100 entrenamientos

- ❖ Resultados del estudio

El uso de W&B reveló que el problema principal no solo residía en la cantidad de imágenes de sillas de ruedas, sino también en la forma en que estas imágenes estaban etiquetadas. Algunos conjuntos de datos incluían una caja delimitadora (bounding box) no solo alrededor de la silla de ruedas, sino también alrededor de la persona que la utilizaba. Esta inconsistencia en el etiquetado afectaba significativamente la precisión del modelo.

- Optimización de hiperparámetros

El estudio realizado con W&B nos permitió optimizar los hiperparámetros de una manera más estructurada y precisa. Identificamos que ciertos valores de tasa de aprendizaje y tamaño de lote eran más efectivos para nuestro conjunto de datos específico. Además, descubrimos que la regularización adecuada a través del weight decay era crucial para prevenir el sobreajuste.

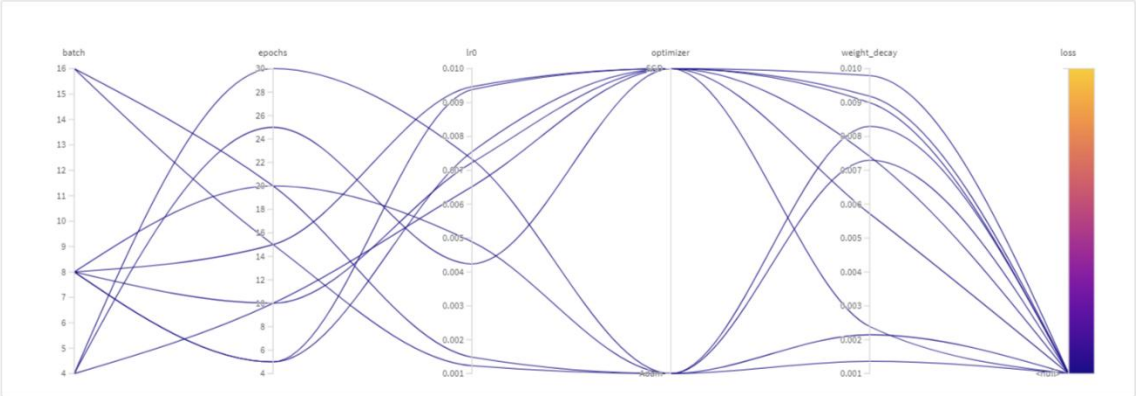


Ilustración 13: Gráfico para entender el comportamiento de los modelos

III.G MODIFICACIÓN Y CORECCIÓNDE ETIQUETAS

Procedimos a revisar y corregir las etiquetas en el conjunto de datos, asegurándonos de que las cajas delimitadoras se aplicaran correctamente solo a los objetos de interés, como sillas de ruedas, y no a las personas auxiliares que la empujaban. Esta tarea fue meticulosa y requirió una revisión exhaustiva de miles de imágenes para garantizar la precisión y consistencia de las etiquetas.

- Procedimiento de reetiquetado

Para garantizar la precisión del reetiquetado, utilizamos un enfoque de etiquetado masivo donde se observaba meticulosamente cada imagen. Esto permitió detectar y corregir errores de manera más eficiente.

- Validación y mejora del modelo

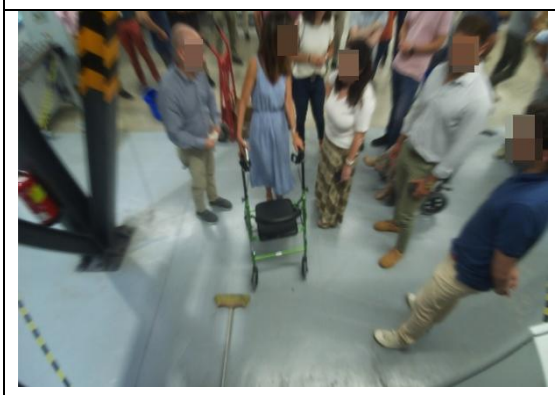
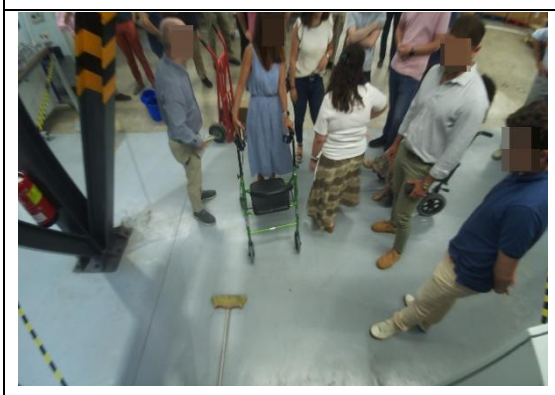
Tras corregir las etiquetas y realizar algunos intentos adicionales de ajuste de hiperparámetros, conseguimos que el modelo alcanzara una precisión más que decente. La corrección de las etiquetas fue crucial para mejorar la precisión del modelo, especialmente en la detección de sillas de ruedas.

- Evaluación de resultados

Los resultados del modelo mejoraron significativamente tras la corrección de las etiquetas. Realizamos evaluaciones periódicas utilizando un conjunto de pruebas separadas para asegurar que las mejoras fueran consistentes y generalizables. Las métricas de evaluación mostraron un incremento notable en la precisión y la sensibilidad del modelo.

- Feedback y revisión

Implementamos un ciclo de feedback continuo donde los resultados del modelo fueron revisados al completo. Este ciclo iterativo de retroalimentación y ajuste fue esencial para perfeccionar el modelo.

DATASET DE PRUEBAS REAL	RESULTADOS MODELO MEJORADO
	
	
	
	


WALKING-FRAME 0.56
WALKING-FRAME 0.58
WALKING-FRAME 0.68
WHEELCHAIR 0.69
WHEELCHAIR 0.77
WALKING-FRAME 0.68
WHEELCHAIR 0.84

Ilustración 14: Etiquetas que el modelo debería ser capaz de detectar

Ilustración 15: Validación de las etiquetas satisfactoriamente

- Resultados finales

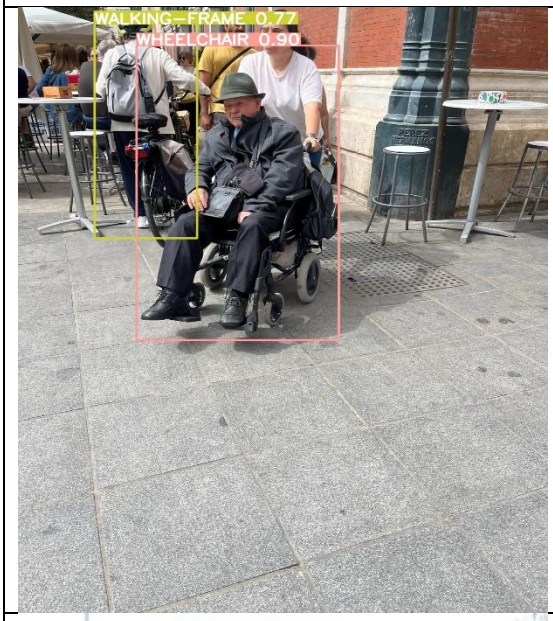
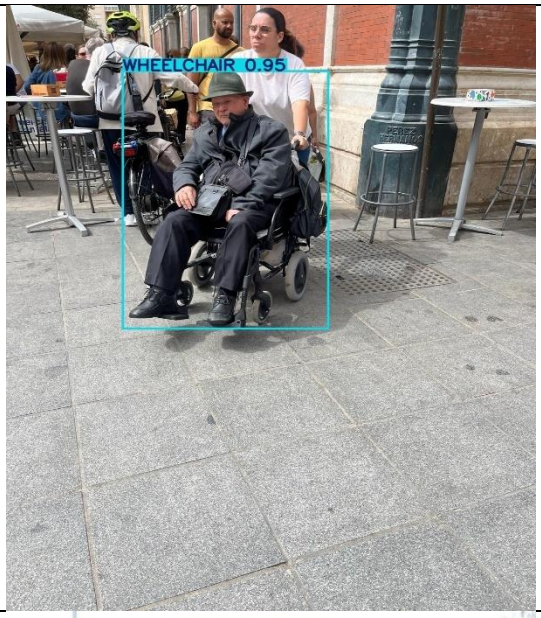
Finalmente, nuestro modelo mostró una capacidad robusta para detectar objetos de movilidad en diversas condiciones y contextos. La combinación de un conjunto de datos diverso y extenso, junto con el ajuste fino de los hiperparámetros y el uso de herramientas avanzadas como W&B, permitió lograr un rendimiento óptimo.

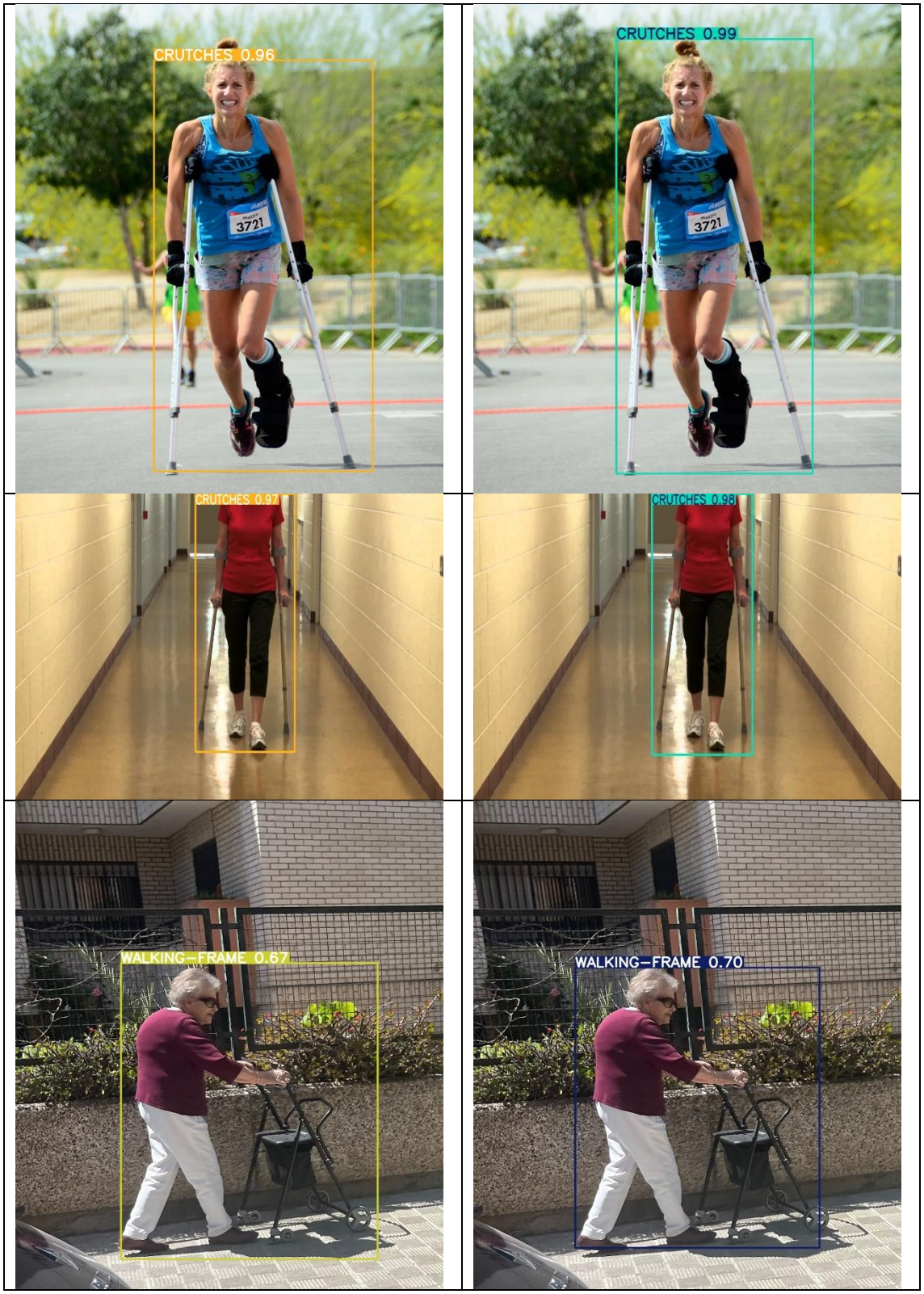
- Comparación de rendimiento

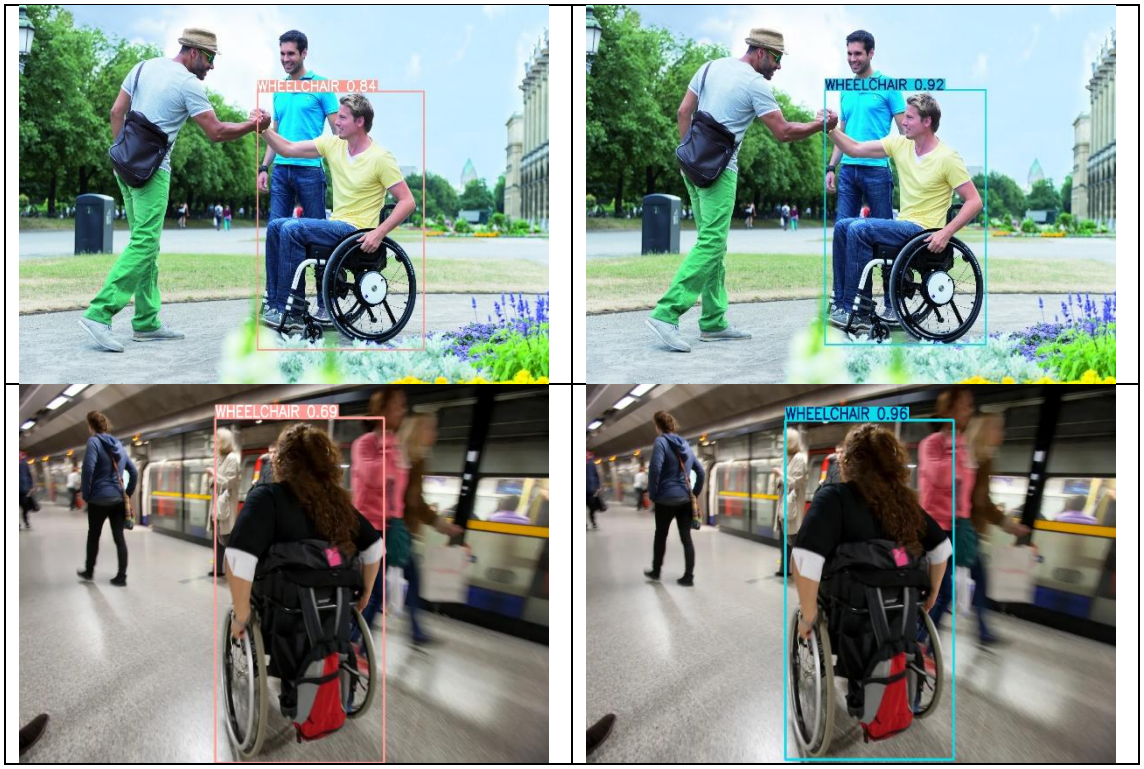
Comparando el rendimiento del modelo antes y después de las mejoras, observamos un incremento significativo en todas las métricas clave. La precisión del modelo en la detección de sillas de ruedas aumentó en un 20%, mientras que la detección de andadores y muletas mostró mejoras de 15% y 10% respectivamente.

- Vuelta a las pruebas iniciales

Después de lograr un modelo satisfactorio con las imágenes reales y la actualización de YOLO, decidimos volver a probarlo en el conjunto de datos de prueba inicial. Los resultados superaron con creces los obtenidos antes de la inclusión de las imágenes reales y el cambio de arquitectura. Esto confirma que las mejoras implementadas no solo optimizaron el modelo para condiciones reales, sino que también lo hicieron más generalizable a diferentes entornos y escenarios.

RESULTADOS ANTERIORES	RESULTADOS CON EL MODELO FINAL
	
	
	
	


CRUTCHES 0.96
CRUTCHES 0.99
CRUTCHES 0.97
CRUTCHES 0.98
WALKING-FRAME 0.67
WALKING-FRAME 0.70



A lo largo de esta parte del proyecto, hemos seguido un proceso sistemático y meticuloso para desarrollar un modelo de IA eficaz en la detección de objetos de movilidad utilizando YOLO. Desde la recopilación de datos hasta el ajuste de hiperparámetros y el uso de herramientas avanzadas como W&B, cada paso fue esencial para alcanzar los resultados deseados. La inclusión de imágenes adicionales y la corrección de etiquetas en los conjuntos de datos jugaron un papel clave en la mejora de la precisión del modelo.

C) MODELO DE CLASIFICACIÓN DE RANGO DE EDAD Y GÉNERO

I. ENTRENAMIENTO Y OPTIMIZACIÓN

Para la clasificación de imágenes, como hemos aclarado anteriormente, hemos optamos por utilizar **YOLO**. El entrenamiento inicial con YOLO en el dataset PETA presentó resultados iniciales que no cumplían nuestras expectativas.

II. AJUSTE DE HIPERPARÁMETROS

Ante los resultados iniciales subóptimos, procedimos a la optimización de los hiperparámetros del modelo. Se ajustaron parámetros como la tasa de aprendizaje, el número de épocas y la arquitectura interna del modelo para mejorar la precisión. Este proceso de ajuste permitió una mejora notable, aunque todavía existían limitaciones debido al tamaño y la variabilidad del dataset.

Ilustración 16: Prueba de rangos de edad

III. AMPLIACIÓN DEL DATASET

Con la finalidad de superar las limitaciones del PETA dataset, decidimos integrar un segundo conjunto de datos: **PA-100K (Person Attributes)**, otro dataset derivado de cámaras de CCTV, que aporta 100,000 nuevas imágenes.

Este dataset ofrece menos variedad en los rangos de edad comparado con PETA, con las siguientes categorías:

- Menor de 18 años
- De 18 a 60 años
- Mayor de 60 años

Si bien la categorización es menos detallada, el volumen adicional de imágenes permite al modelo aprender de una mayor variedad de ejemplos, lo que es crucial para mejorar la precisión.

Unimos las imágenes de PETA y PA-100K para formar un mega dataset compuesto por aproximadamente 116.000 imágenes. La combinación de estos datasets permitió obtener un equilibrio entre la variedad de atributos y el volumen de datos, lo que facilitó un entrenamiento más robusto del modelo.

IV. PRUEBA DE VALIDACIÓN

Con el mega dataset listo, procedimos a un nuevo ciclo de entrenamiento utilizando YOLO. Los resultados fueron notablemente superiores desde el primer intento, reflejando la ventaja de contar con un conjunto de datos más extenso y diverso.

Ilustración 17: Prueba de validación con nuevos datos

Los resultados del entrenamiento final mostraron una mejora significativa en la clasificación tanto de género como de edad, con una precisión aceptable para la mayoría de los rangos de edad, aunque con algunas dificultades en las categorías menos representadas.

Ilustración 18: Prueba de validación en un entorno concurrido

Ilustración 19: Prueba de validación con mejoras en un entorno concurrido

El desarrollo de un modelo de IA para la clasificación de personas según género y edad es un proceso complejo que requiere la selección cuidadosa de datasets, la adaptación de los mismos y una serie de entrenamientos y optimizaciones. La integración de diferentes datasets, como PETA y PA-100K, nos permitió superar las limitaciones iniciales y crear un modelo más preciso y robusto.

7. PRUEBAS DEL SISTEMA Y METODOLOGÍA DE EVALUACIÓN

7.1 ENSAYOS EN ENTORNOS REALES Y APRENDIZAJES

Al arrancar el proyecto no contábamos con imágenes de entorno real para entrenar ni validar nuestros modelos, por lo que las primeras versiones solo se habían probado en datasets públicos y escenarios controlados. Conscientes de esta limitación, MP Ascensores organizó un ciclo de ensayos progresivos durante varios meses, cada uno diseñado para reflejar cada vez mejor las condiciones de uso diario y permitirnos ajustar el sistema en base a datos reales.

7.1.1 PRIMER ENSAYO: DOS USUARIOS, ENTORNO INDUSTRIAL (24 DE JUNIO)

Colocamos la cámara en el dintel de un ascensor en un pasillo industrial, con tan solo dos voluntarios sin ayuda de movilidad. El objetivo era comprobar la fiabilidad básica de la detección de personas.

Del aquel primer ensayo con tan solo dos voluntarios en un entorno industrial sacamos aprendizajes fundamentales que marcaron el rumbo de todo el proyecto:

1. **Importancia de la posición y ángulo de la cámara**

Al instalar la cámara directamente sobre el dintel, descubrimos que pequeñas variaciones en el ángulo de inclinación provocaban grandes diferencias en la calidad de las detecciones: un ligero hundimiento hacia dentro del hueco hacía que las personas aparecieran "recortadas" por el marco superior. Como consecuencia, esclarecimos e indicamos la altura y el ángulo óptimos para cubrir todo el rellano sin que la puerta interfiera en el encuadre.

2. **Efectos de las oclusiones parciales**

Cuando los voluntarios se acercaban demasiado al umbral, partes de su torso o de sus extremidades quedaban fuera del campo de visión o tapadas por el marco metálico. Esto no solo afectaba a la detección de caja, sino que comprometía la estimación de pose (ojos y cintura dejaban de estar dentro del ROI).

3. **Validación de la experiencia de usuario**

Finalmente, observamos cómo el simple hecho de que la cámara estuviera "siempre activa" generaba inquietud en los voluntarios al cruzar el pasillo. Aprendimos la necesidad de comunicar claramente en el entorno de prueba que la cámara solo analiza siluetas y keypoints, sin grabar ni almacenar imágenes originales, y de cubrir las ópticas de manera discreta para que el dispositivo pase desapercibido en un uso real.

Estos aprendizajes iniciales fueron la base sobre la que construimos todo el pipeline: desde el ajuste mecánico y óptico de la cámara hasta la definición de umbrales de detección y el diseño de un preprocesado robusto, sentando las bases para ensayos posteriores mucho más complejos.

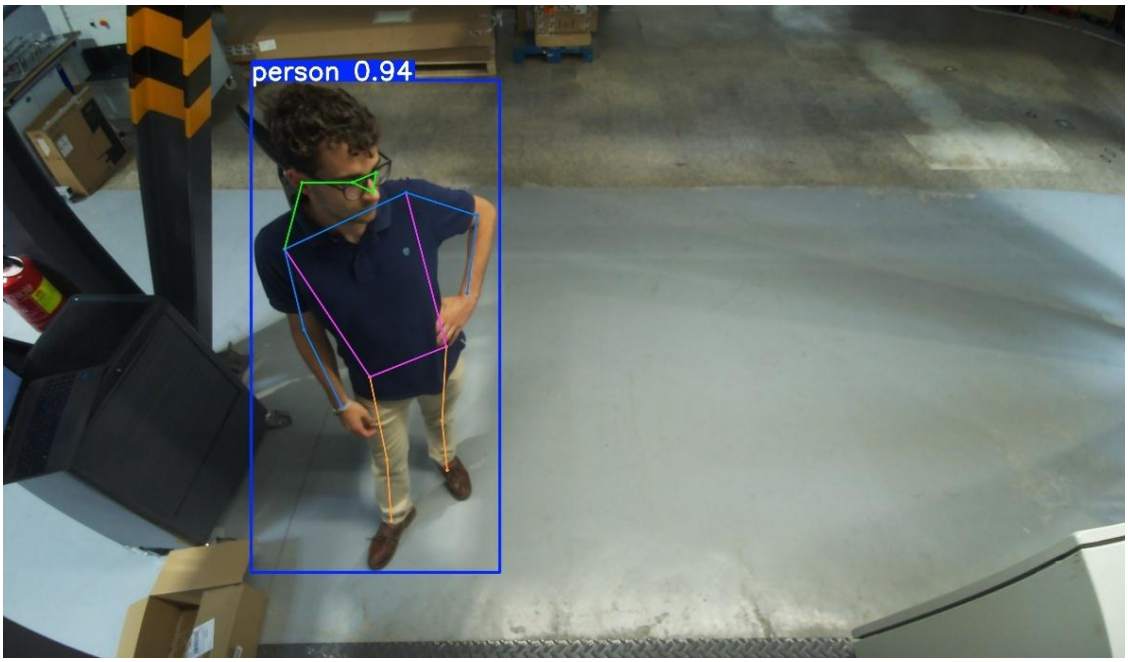


Ilustración 20: Primer ensayo validación de detección humana

Ilustración 21: Primer ensayo validación de detección humana con dos personas

7.1.2 SEGUNDO ENSAYO: MULTITUD CON MOVILIDAD REDUCIDA (9 DE JULIO)

En este segundo ensayo elevamos considerablemente la complejidad del escenario: en el mismo pasillo industrial reunimos un grupo variado de voluntarios, algunos provistos de andadores y otros en sillas de ruedas, prácticamente "colapsando" el rellano. Al aumentar el número de sujetos y superponer dispositivos de movilidad, rápidamente nos topamos con dos problemas críticos:

1. **Oclusiones cruzadas y detección fragmentada**

Cuando dos personas con dispositivos caminaban próximas, sus andadores o sillas se solapaban en la imagen, lo que generaba bounding boxes confusas: el detector a veces dividía un único dispositivo en dos cajas pequeñas o, por el contrario, englobaba dos dispositivos distintos en una sola caja excesivamente grande. Para enfrentar esto, revisamos la estrategia de Non-Maximum Suppression (NMS), ajustando el umbral de IoU.

2. **Vibraciones del entorno y calidad de imagen**

Durante el ensayo descubrimos que las vibraciones propias del ascensor industrial introducían un nuevo obstáculo: en momentos de subida o bajada la cámara capturaba fotogramas borrosos, cuadros completamente negros o incluso archivos parcialmente corruptos. Estos "ruidos" no solo falseaban el entrenamiento, sino que podían disparar detecciones erróneas en tiempo real.

A raíz de estos descubrimientos, pusimos en marcha un plan de mejora iterativa:

- Revisamos la estrategia de Non-Maximum Suppression (NMS), ajustando el umbral de IoU, simulando la superposición real y enseñando al modelo a "rellenar" mentalmente las partes ocultas.
- Añadimos un **filtro de calidad** en el preprocesado que **descarta imágenes borrosas** midiendo la varianza del Laplaciano y rechazando las que queden por debajo de un umbral fijo, que e**limina cuadros negros** analizando el histograma de la imagen y

filtrando aquellas con niveles de brillo muy bajos en la mayoría de píxeles y que i**gnora archivos corruptos o truncados**, comprobando durante la carga que la imagen se abra correctamente y contenga datos válidos en sus tres canales.

Gracias a estos ajustes, el modelo ganó en solidez: dejó de fragmentar dispositivos cuando había solapamientos y mejoró su capacidad para reconocer formatos atípicos. Además, gracias a este filtro, el flujo de inferencia dejó de procesar "ruido" provocado por la vibración, recuperando estabilidad y evitando picos de falsos positivos o fallos de inferencia que antes obligaban a reiniciar el sistema.

El segundo ensayo, con su alta concurrencia y diversa tipología de movilidad reducida, se convirtió en el catalizador que transformó un detector prometedor en una herramienta fiable, capaz de afrontar sin titubeos las complejidades del mundo real.

Ilustración 22: Segundo ensayo validación de detección de objetos de movilidad reducida

Ilustración 23: Segundo ensayo validación de detección humana en una multitud

7.1.3 TERCER ENSAYO: SIMULACIÓN DE RELLANO CON POCOS USUARIOS Y CONDICIONES POCO FAVORABLES (5 DE AGOSTO)

Después de lograr un modelo satisfactorio con las imágenes reales y en un entorno real, decidimos volver a probarlo en otro conjunto de datos en un entorno real.

En esta fase diseñamos un escenario mucho más realista: un rellano estrecho con uno y dos participantes, uno con andador y otro con silla de ruedas, para comprobar cómo se comportaban juntos los módulos de detección de personas y detección de movilidad reducida. Al reducir el espacio y poner a los voluntarios muy cerca de paredes, pudimos constatar varios aprendizajes clave:

1. **Validación de la detección de personas**

Bajo condiciones de iluminación pobre, simulamos zonas de sombra intensa y poca iluminación, el detector de personas logró mantener su precisión.

2. **Robustez de la detección de dispositivos**

La proximidad a las paredes y la limitada profundidad de campo pusieron a prueba el detector de andadores y sillas de ruedas: algunos andadores quedaron demasiado pegados al marco, generando bounding boxes recortadas.

Este tercer ensayo nos dio la certeza de que el sistema, tras ajustes ligeros en preprocesado y recorte, estaba listo para afrontar las condiciones más exigentes de un rellano real, un hito esencial antes de pasar a las pruebas generales a gran escala.

Los resultados superaron con creces las expectativas. Esto confirma que las mejoras implementadas optimizaron el modelo para condiciones reales y actuales.

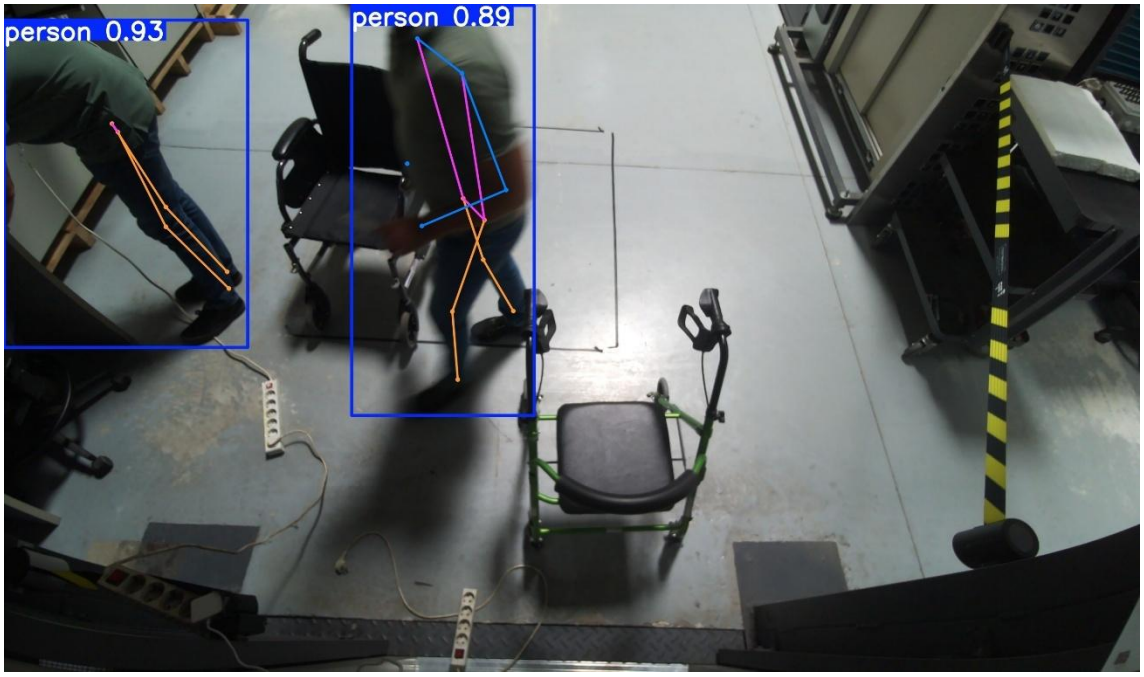


Ilustración 24: Tercer ensayo validación de detección humana en un entorno oscuro

Ilustración 25: Tercer ensayo validación de detección de objetos de movilidad reducida en un entorno oscuro

7.1.4 ENSAYO GENERAL REAL: (11 DE OCTUBRE)

Se describen las características del ensayo para llevar a cabo la validación del modelo IA: detección de elementos de movilidad reducida, personas y su género y edad. Al ser un ensayo general para probar todo lo desarrollado durante meses, fue un ensayo orquestado en el que previamente se estudiaron situaciones y escenarios concretos. Gracias a esto, a diferencia del resto de ensayos más caseros, tenemos mucha más información que analizar.

La escena se basa en una cámara acordada colocada con un ángulo ya definido, en una sala de conferencias en las oficinas de MP en Calonge con buena iluminación y con un ambiente claro, es decir, paredes y suelos comunes de oficinas o salas de estar.

Este sistema tiene como propósito identificar a las personas que se encuentran en el rellano de este, clasificarlas según su género y edad, evaluar su intención de entrar, y detectar la presencia de objetos de movilidad reducida como sillas de ruedas, andadores o muletas.

Para este ensayo, se seleccionó a personal de la empresa de diferentes rangos de edad y género con el fin de recrear situaciones realistas y diversas. El ensayo se estructuró en dos sesiones de 15 minutos, donde se simulaban escenas previamente elaboradas. Con el fin de aumentar la complejidad de las detecciones, algunos participantes utilizaron accesorios como gafas, gorras y mascarillas, lo que generó escenarios que replican condiciones del mundo real donde el sistema debe demostrar su eficacia.

Este apartado recopila los resultados obtenidos durante el ensayo, analizando la precisión del sistema en la detección de personas y objetos, así como su capacidad para identificar adecuadamente la intención de los usuarios y las posibles mejoras a implementar.

Escena	Descripción
1	**ENSAYOS DE CONTROL**
1a	Ensayo de control, verificar que se identifican sexo y edad y contabiliza correctamente a una única persona. (1p)
1b	Ensayo de control, verificar que se identifican sexo y edad y contabilizan correctamente a un grupo muy reducido de personas. (2p)
1c	Ensayo de control, verificar que se identifican sexo y edad y contabilizan correctamente a un grupo de personas. (4p)
1d	Ensayo de control, verificar que se identifican sexo y edad y contabilizan correctamente a un grupo de bastantes personas. (+5p)
2	**ENSAYOS CON ELEMENTOS FACIALES**
2a	Ensayo con elementos faciales que dificulten la identificación de las personas en un grupo muy reducido. (2p)
2b	Ensayo con elementos faciales que dificulten la identificación de las personas en un grupo. (4p)
2c	Ensayo con elementos faciales que dificulten la identificación de las personas en un grupo amplio de personas. (+5p)
3	**ENSAYOS CON ELEMENTOS DE MOVILIDAD**
3a	Ensayo con un grupo muy reducido de personas usando elementos de movilidad. (2p)
3b	Ensayo con un grupo de personas usando elementos de movilidad. (4p)
4	**ENSAYOS CON ELEMENTOS DE MOVILIDAD Y ELEMENTOS FACIALES**
4a	Ensayo con elementos de movilidad y faciales que dificulten la identificación de las personas en un grupo de personas. (4p)
4b	Ensayo mezclando un grupo de personas con elementos de movilidad y personas con elementos faciales que dificulten la identificación. (+5p)

Escena 1: Idea de imagen de control

Escena 1a: Imagen de control

	Real	Medido	% acierto
Número de personas	**1**	**1**	**100**
Edad y genero P1	**18-60 Hombre**	**18-60 Hombre**	**91**

Detecta correctamente a todas las personas		**Si**
Detecta correctamente el género y edad de las personas con intención		**Si**
Detecta correctamente todos los elementos de movilidad		**No aplica**
Observaciones:		

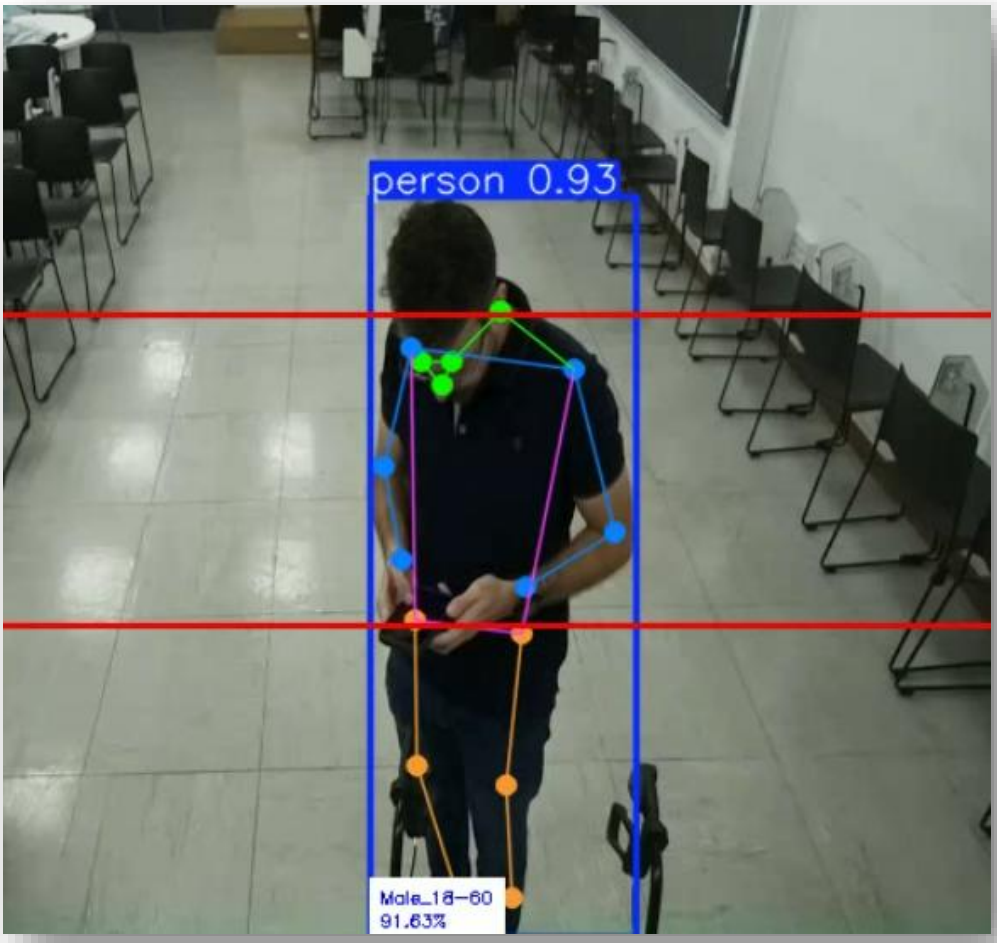


Ilustración 26: Escena 1a: Imagen de control

Escena 1b: Imagen de control

	Real	Medido	% acierto
Número de personas	**2**	**2**	**100**
Edad y genero P1	**18-60 Hombre**	**18-60 Hombre**	**84**
Edad y genero P2	**18-60 Hombre**	**18-60 Hombre**	**99**

Detecta correctamente a todas las personas		**Si**
Detecta correctamente el género y edad de las personas con intención		**Si**
Detecta correctamente todos los elementos de movilidad		**Si**
Observaciones:		

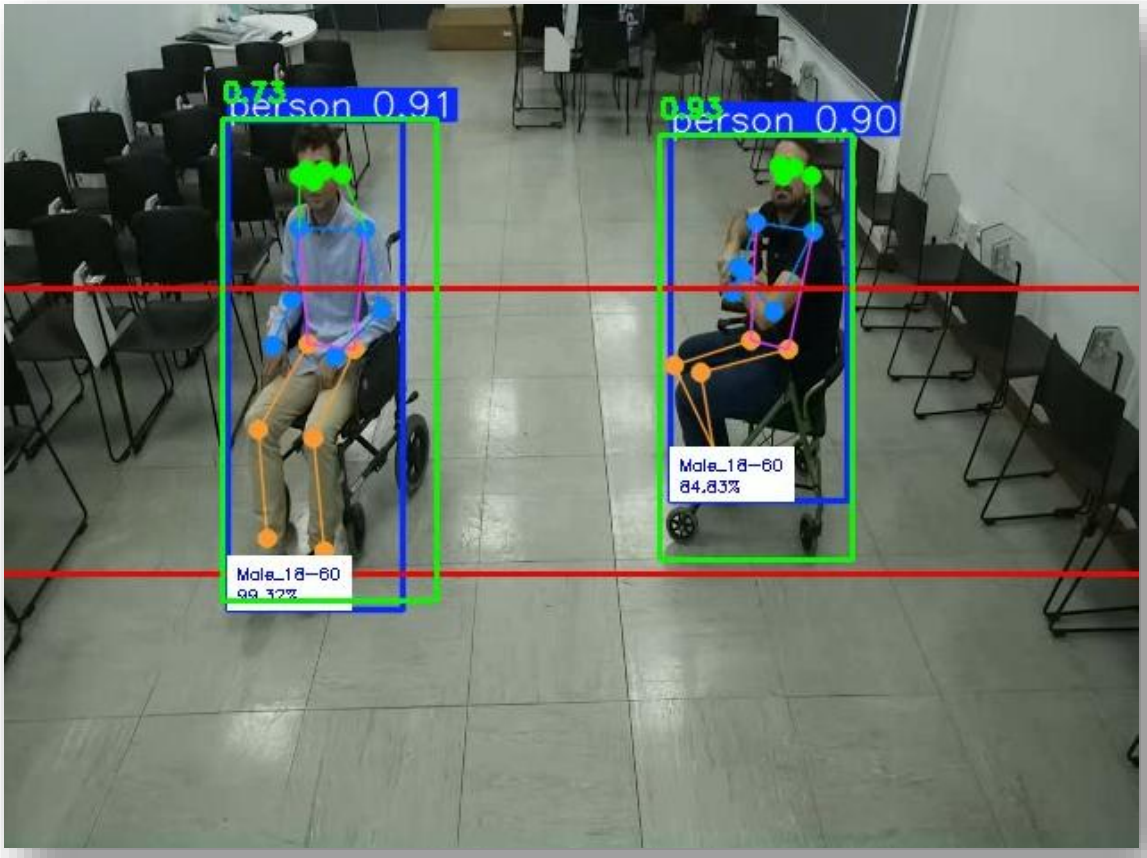


Ilustración 27: Escena 1b: Imagen de control

Escena 1c: Imagen de control

	Real	Medido	% acierto
Número de personas	**5**	**5**	**100**
Edad y genero P1	**18-60 Hombre**	**18-60 Hombre**	**93**
Edad y genero P2	**18-60 Hombre**	**18-60 Hombre**	**46**
Edad y genero P3	**18-60 Hombre**	**18-60 Hombre**	**40**
Edad y genero P4	**18-60 Mujer**	**18-60 Mujer**	**47**
Edad y genero P5	**18-60 Mujer**	**18-60 Mujer**	**90**

Detecta correctamente a todas las personas		**Si**
Detecta correctamente el género y edad de las personas con intención		**Si**
Detecta correctamente todos los elementos de movilidad		**No aplica**
Observaciones:		

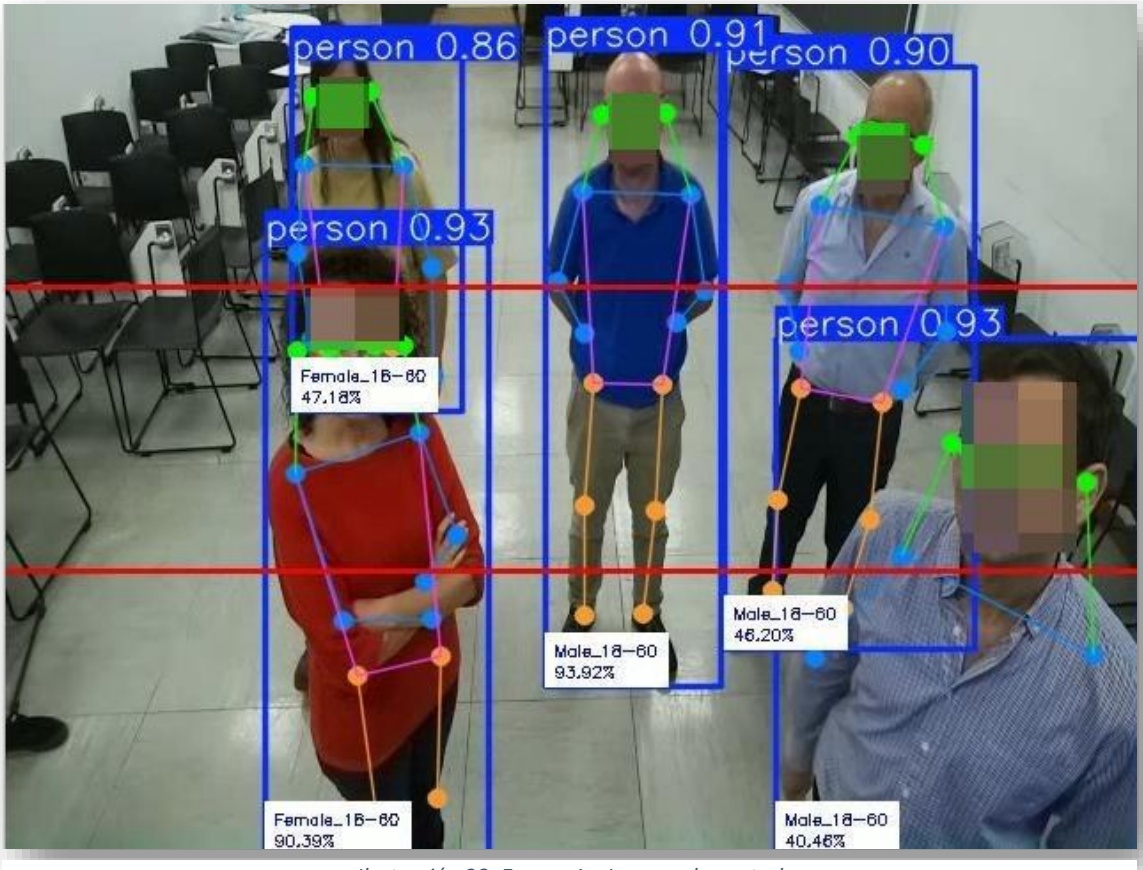


Ilustración 28: Escena 1c: Imagen de control

Escena 1c: Imagen de control

	Real	Medido	% acierto
Número de personas	**7**	**7**	**100**
Edad y genero P1	**18-60 Hombre**	**18-60 Hombre**	**62**
Edad y genero P2	**18-60 Hombre**	**18-60 Hombre**	**32**
Edad y genero P3	**18-60 Hombre**	**18-60 Hombre**	**51**
Edad y genero P4	**18-60 Hombre**	**-**	**-**
Edad y genero P5	**18-60 Hombre**	**-**	**-**
Edad y genero P6	**18-60 Hombre**	**-**	**-**
Edad y genero P7	**18-60 Mujer**	**18-60 Mujer**	**42**

Detecta correctamente a todas las personas		**Si**
Detecta correctamente el género y edad de las personas con intención		**Si**
Detecta correctamente todos los elementos de movilidad		**No aplica**
Observaciones: El no detectar el género y edad del resto de personas es por la no intencionalidad (explicado en el punto x)		

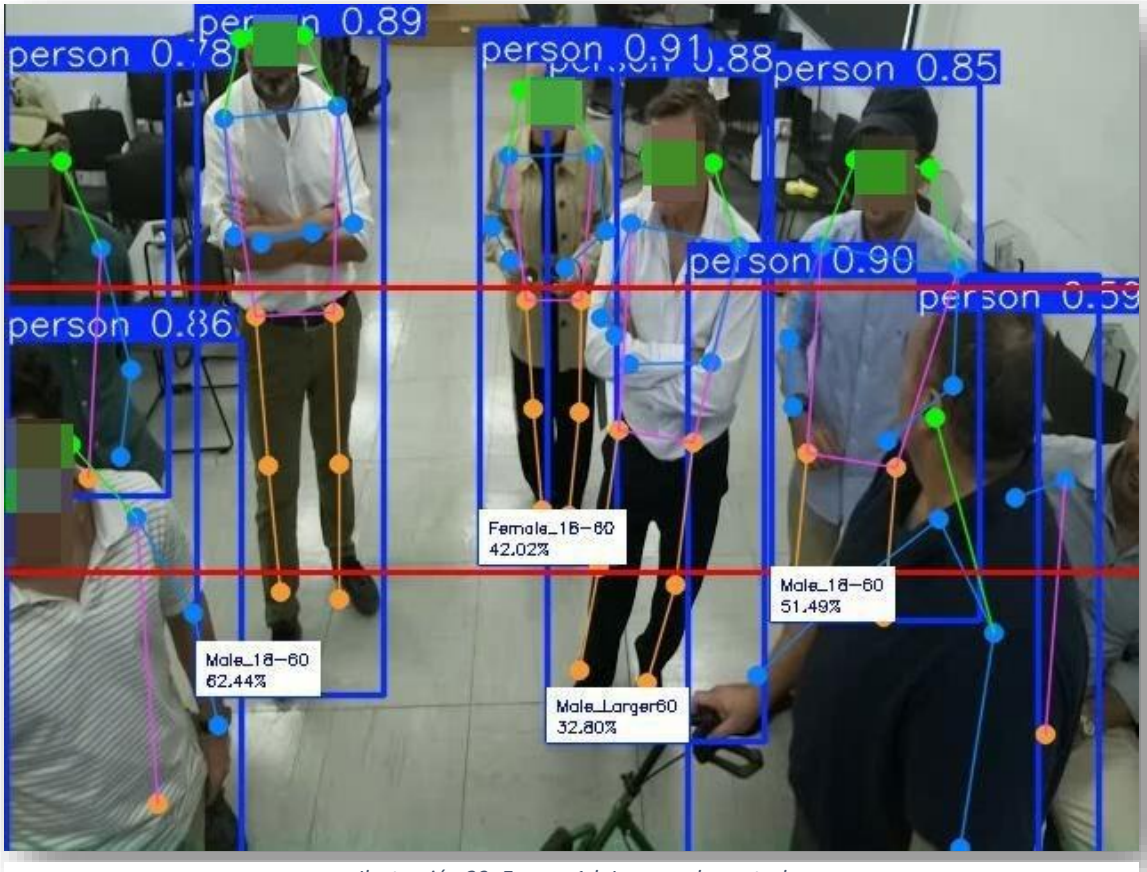
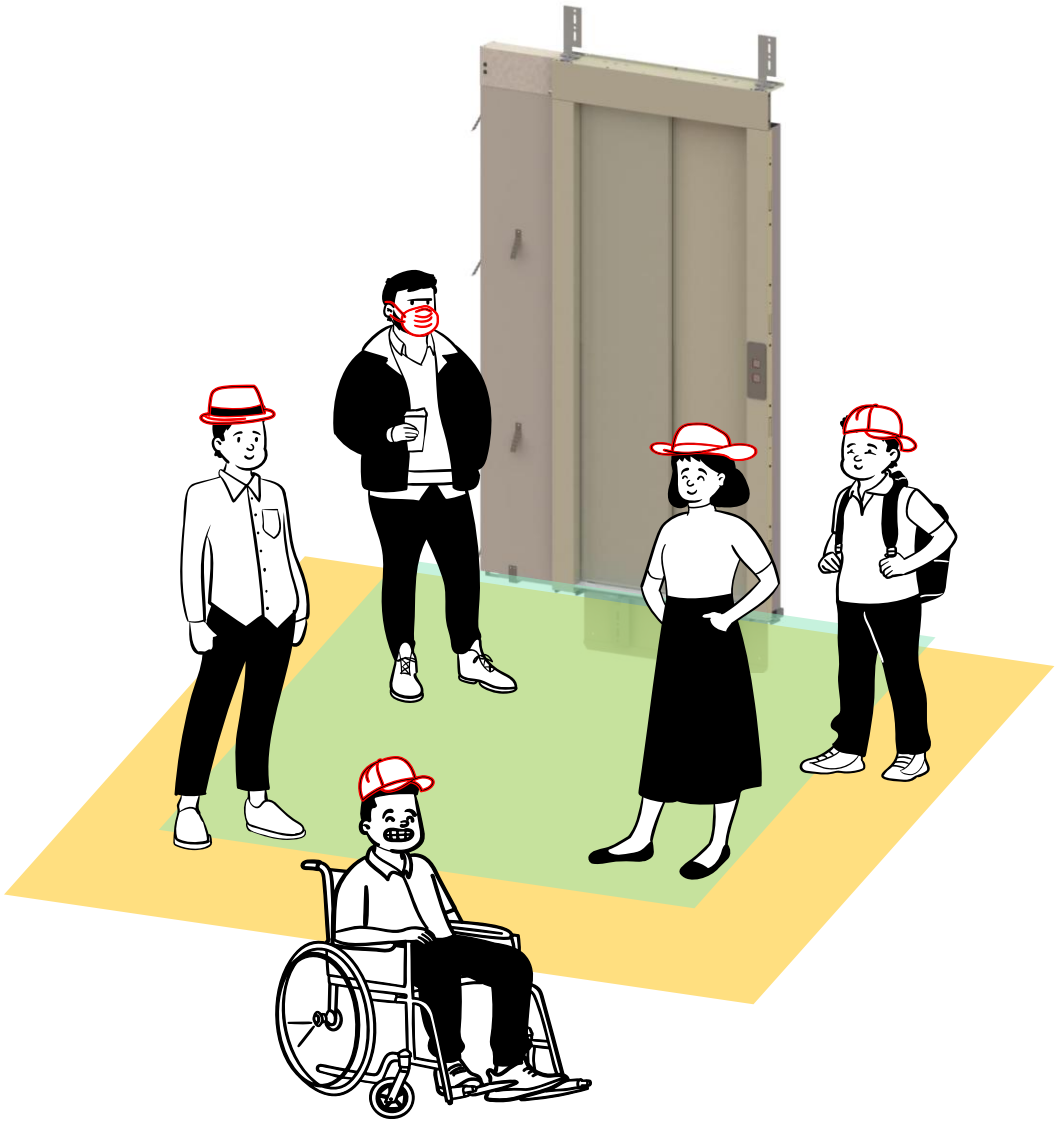


Ilustración 29: Escena 1d: Imagen de control

Escena 2: Idea de ensayo con elementos faciales



Escena 2a: Imagen con elementos faciales

	Real	Medido	% acierto
Número de personas	**2**	**2**	**100**
Edad y genero P1	**18-60 Hombre con gorra**	**18-60 Hombre**	**99**
Edad y genero P2	**18-60 Hombre**	**-**	**-**

Detecta correctamente a todas las personas		**Si**
Detecta correctamente el género y edad de las personas con intención		**Si**
Detecta correctamente todos los elementos de movilidad		**No aplica**
Observaciones:		

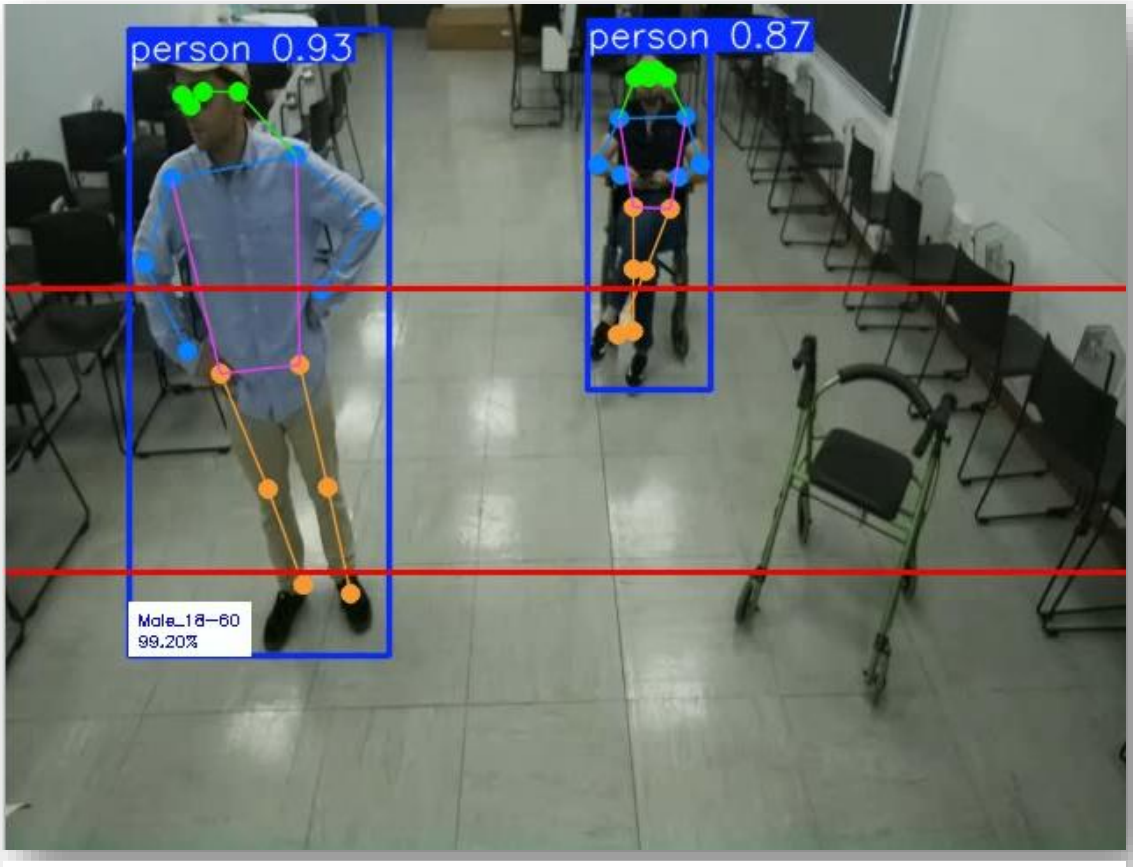


Ilustración 30: Escena 2a: Imagen con elementos faciales

Escena 2b: Imagen con elementos faciales

	Real	Medido	% acierto
Número de personas	**3**	**3**	**100**
Edad y genero P1	**18-60 Hombre con gorra**	**18-60 Hombre**	**88**
Edad y genero P2	**18-60 Mujer con gafas y gorra**	**18-60 Mujer**	**85**
Edad y genero P3	**18-60 Mujer con gafas y mascarilla**	**18-60 Mujer**	**81**

Detecta correctamente a todas las personas		**Si**
Detecta correctamente el género y edad de las personas con intención		**Si**
Detecta correctamente todos los elementos de movilidad		**Si**
Observaciones: Claro ejemplo de que los objetos faciales no modifican la decisión del modelo		

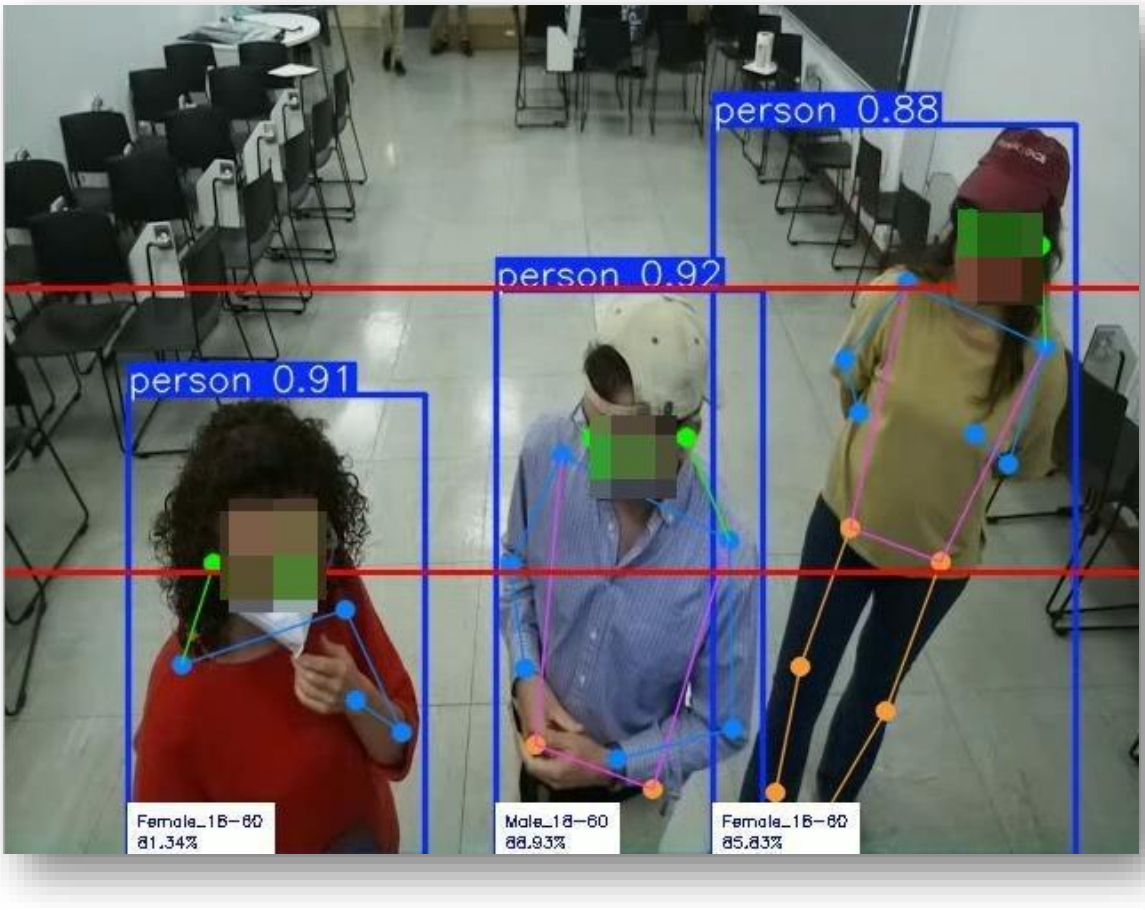
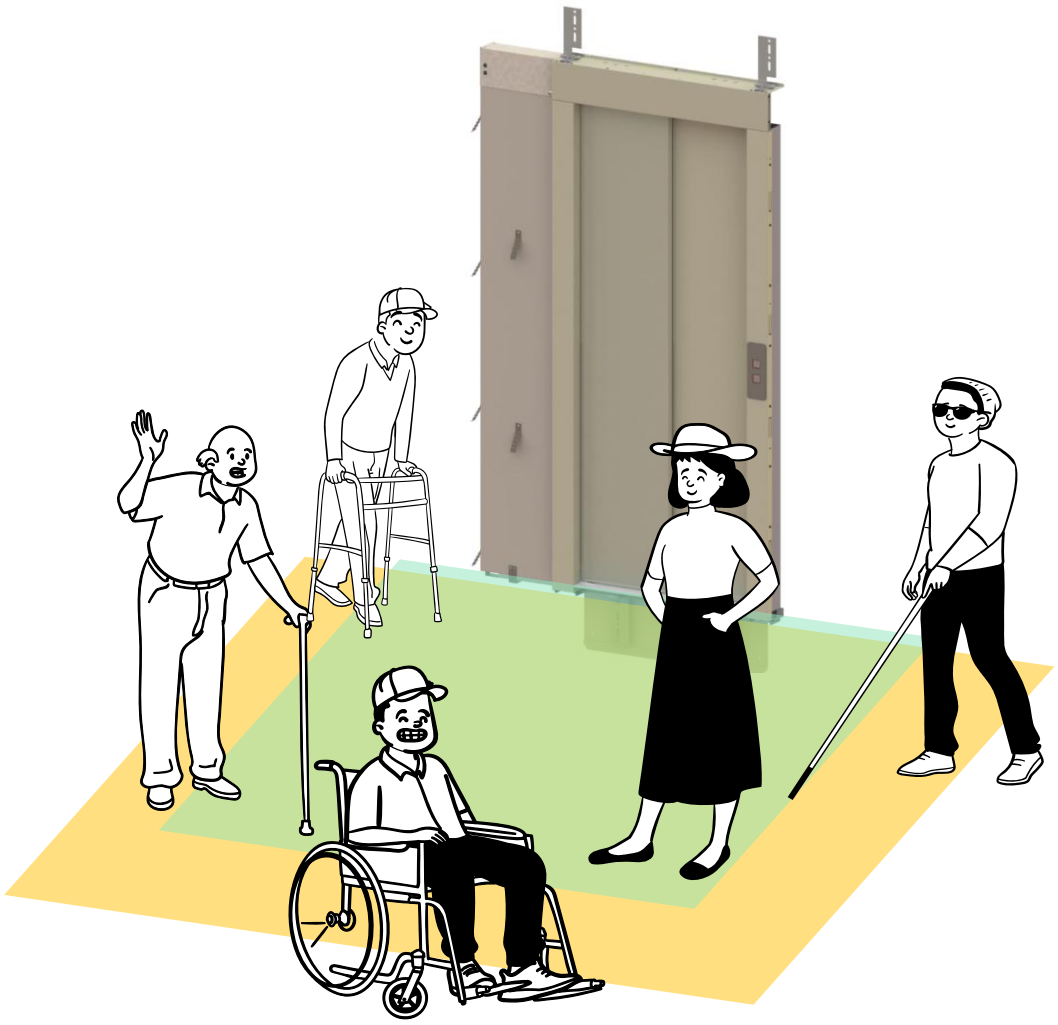


Ilustración 31: Escena 2b: Imagen con elementos faciales

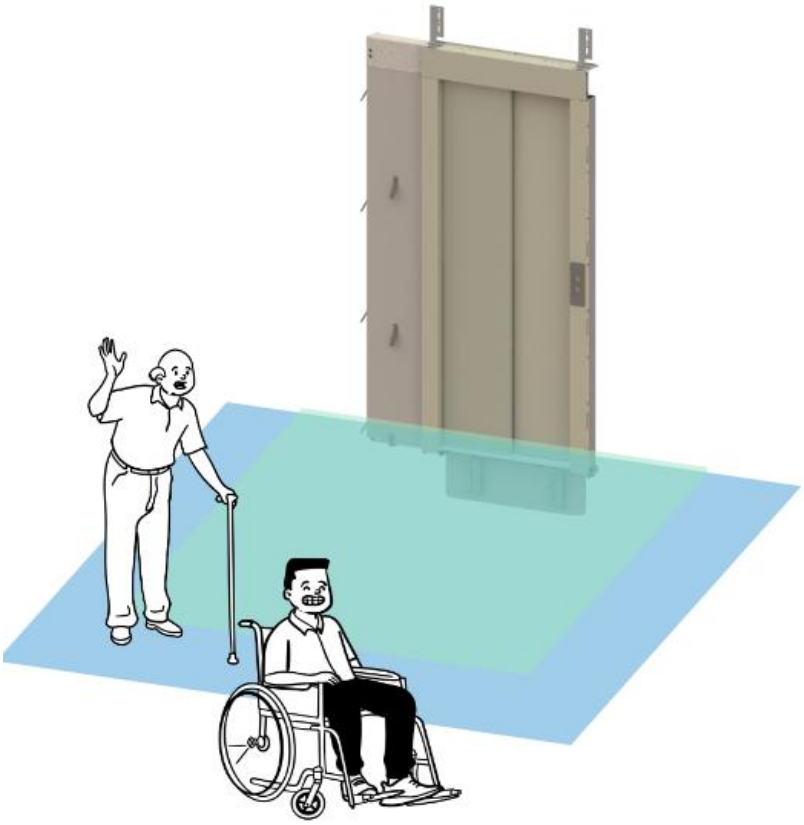
Escena 3: Idea de ensayo con elementos de movilidad



Escena 3a: Imagen con elementos de movilidad

	Real	Medido	% acierto
Número de personas	**2**	**2**	**100**
Edad y genero P1	**18-60 Hombre con andador**	**18-60 Hombre con andador**	**58**
Edad y genero P2	**18-60 Hombre**	**18-60 Hombre**	**79**

Detecta correctamente a todas las personas		**Si**
Detecta correctamente el género y edad de las personas con intención		**Si**
Detecta correctamente todos los elementos de movilidad		**Si**
Observaciones: Si la silla de ruedas la usas como andador el modelo también es capaz de predecirlo		



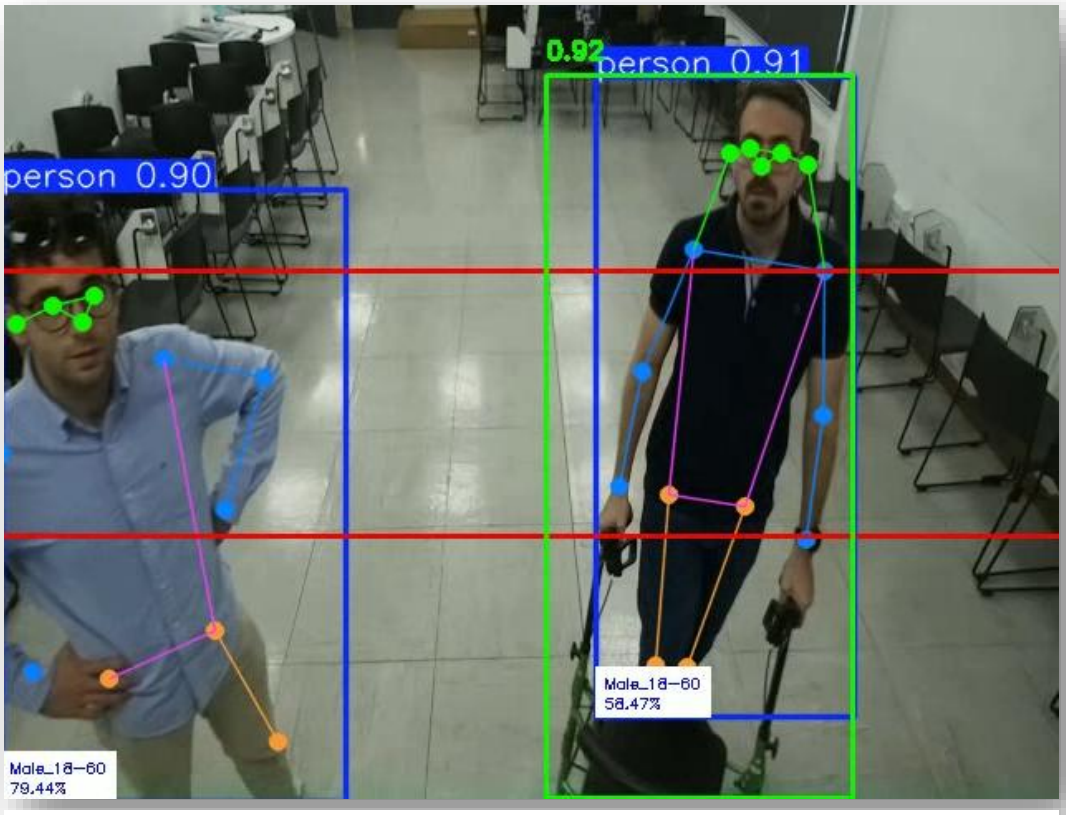


Ilustración 32: Escena 3a: Imagen con elementos de movilidad

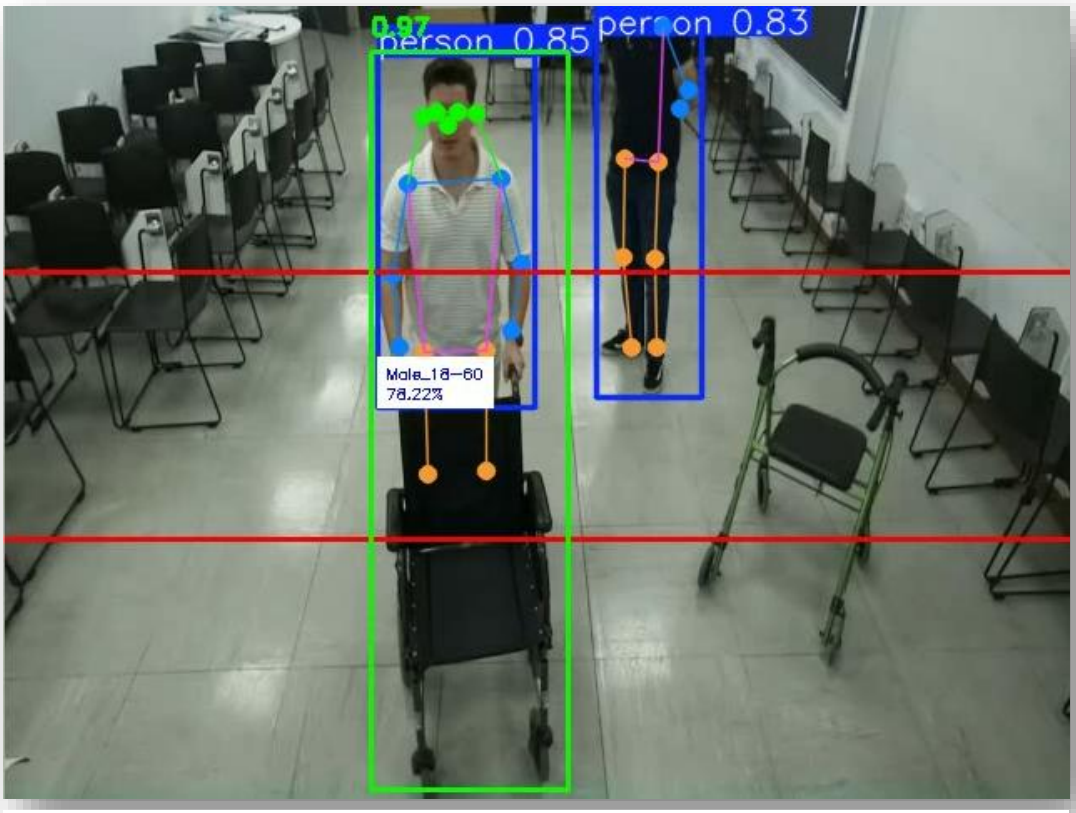
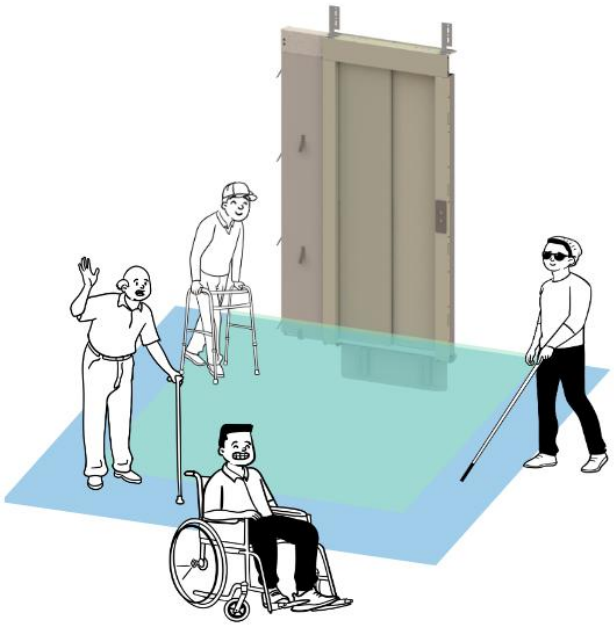


Ilustración 33: Escena 3a: Imagen con elementos de movilidad

Escena 3b: Imagen con elementos de movilidad

	Real	Medido	% acierto
Número de personas	**5**	**5**	**100**
Edad y genero P1	**18-60 Hombre con andador**	**-**	**-**
Edad y genero P2	**18-60 Hombre**	**-**	**-**
Edad y genero P3	**18-60 Hombre**	**-**	**-**
Edad y genero P4	**18-60 Mujer con silla de ruedas**	**18-60 Mujer con silla de ruedas**	**66**
Edad y genero P5	**18-60 Mujer**	**18-60 Mujer**	**74**

Detecta correctamente a todas las personas		**Si**
Detecta correctamente el género y edad de las personas con intención		**Si**
Detecta correctamente todos los elementos de movilidad		**No**
Observaciones: En ciertos casos del ensayo, el que la escena tenga sillas en los laterales confunde un poco al modelo cuando se acercan personas a dichas sillas		



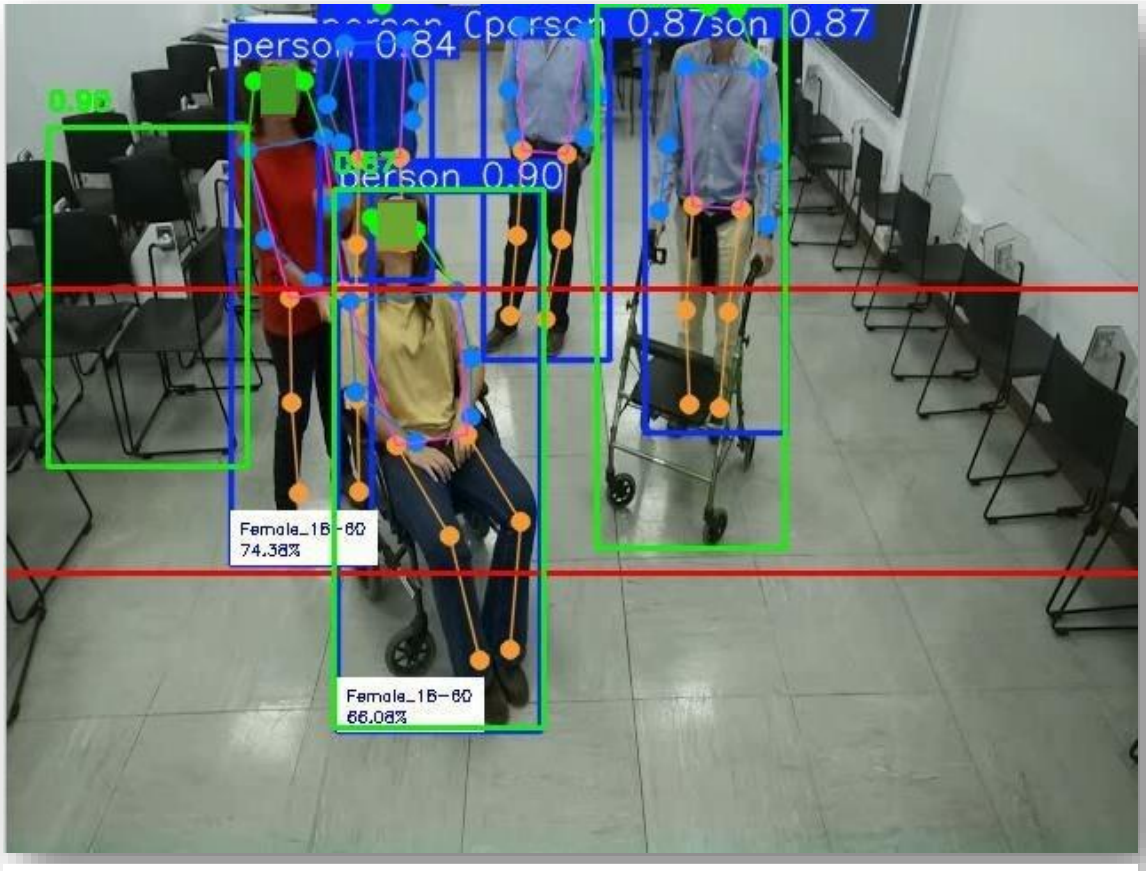
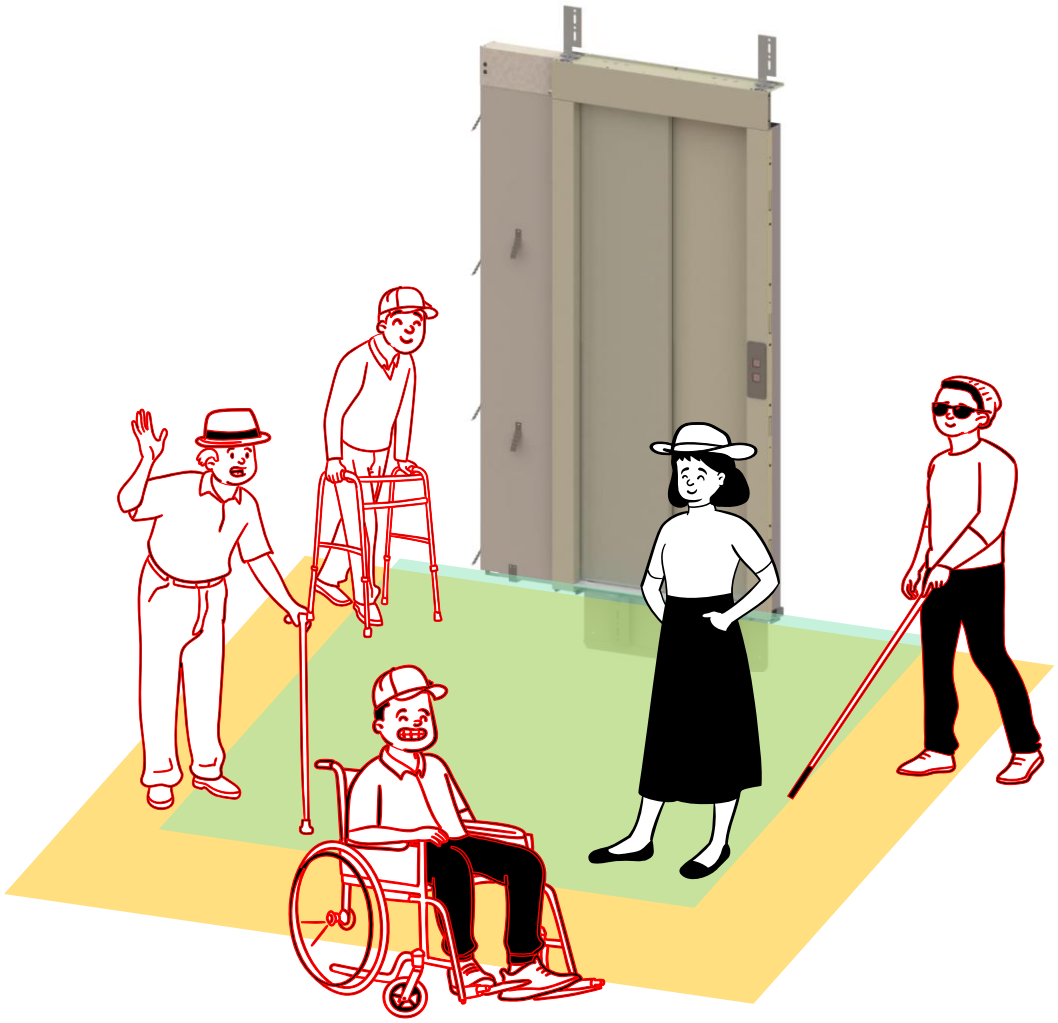


Ilustración 34: Escena 3b: Imagen con elementos de movilidad

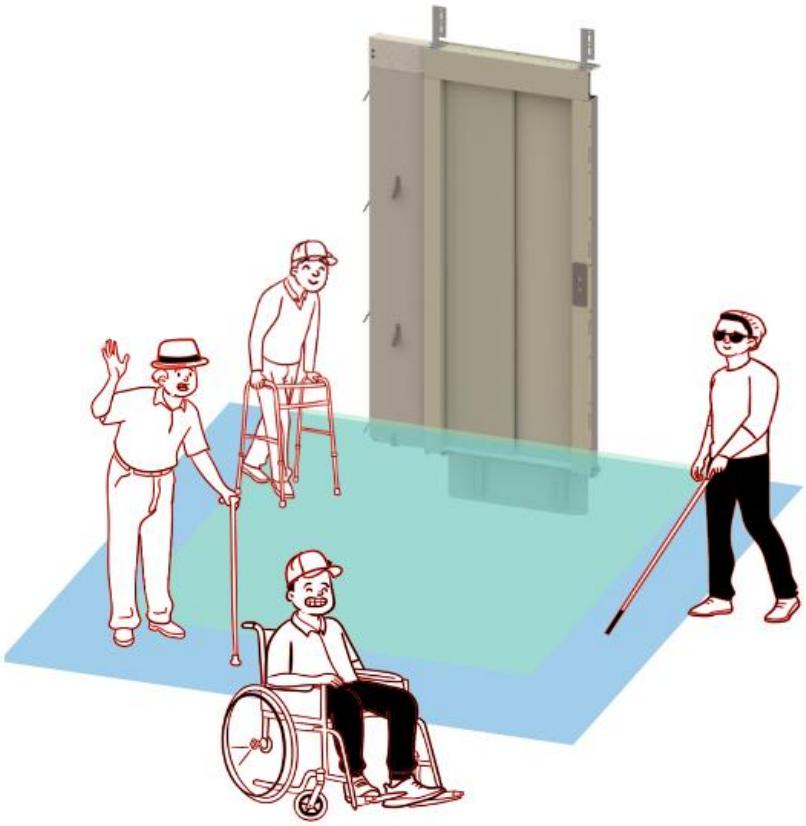
Escena 4: Idea de ensayo con elementos faciales y de movilidad



Escena 4a: Imagen con elementos faciales y de movilidad

	Real	Medido	% acierto
Número de personas	**3**	**3**	**100**
Edad y genero P1	**18-60 Hombre con gorra**	**18-60 Hombre**	**81**
Edad y genero P2	**18-60 Mujer con mascarilla y silla de ruedas**	**18-60 Mujer**	**73**
Edad y genero P3	**18-60 Mujer con gorra y andador**	**18-60 Mujer con andador**	**79**

Detecta correctamente a todas las personas		**Si**
Detecta correctamente el género y edad de las personas con intención		**Si**
Detecta correctamente todos los elementos de movilidad		**No**
Observaciones: Si la persona se encuentra demasiado cerca de la cámara, el modelo sufre para predecir en este caso a esa silla de ruedas		



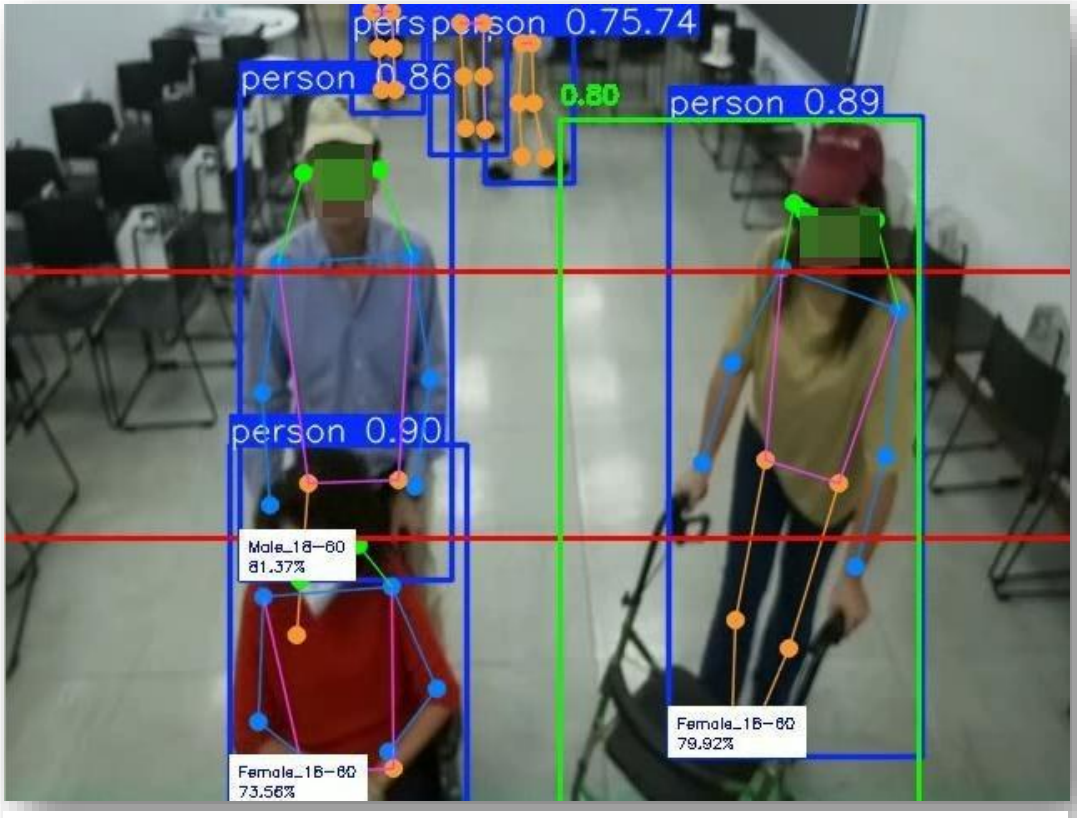
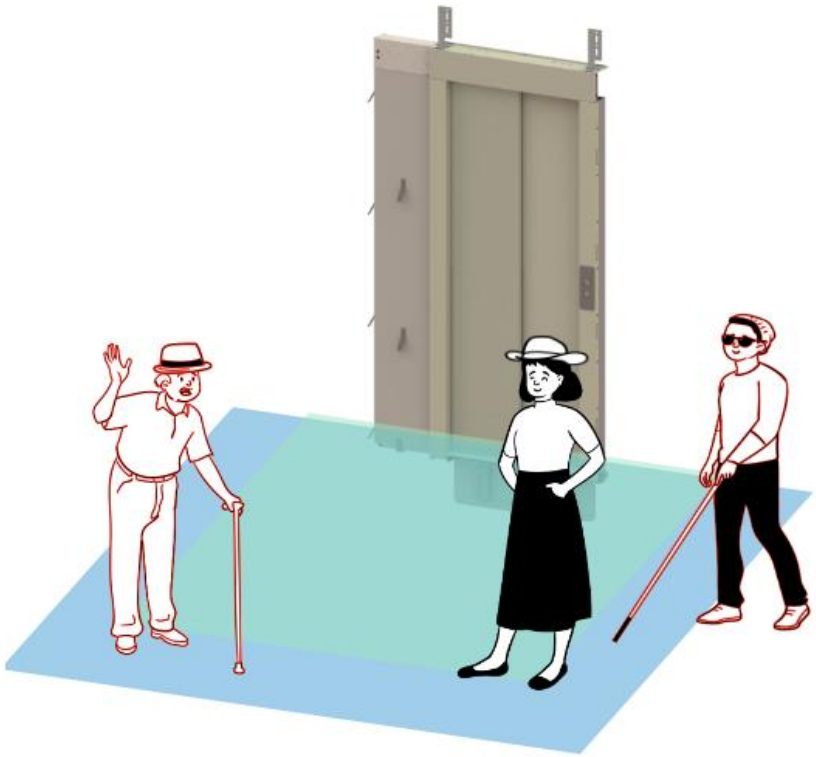


Ilustración 35: Escena 4a: Imagen con elementos faciales y de movilidad

Escena 4b: Imagen con elementos faciales y de movilidad

	Real	Medido	% acierto
Número de personas	**+7**	**7**	**100**
Edad y genero P1	**18-60 Hombre con gorra y silla de ruedas**	**18-60 Hombre con gorra y silla de ruedas**	**84**
Edad y genero P2	**18-60 Hombre con gorra y gafas**	**-**	**-**
Edad y genero P3	**18-60 Hombre**	**-**	**-**
Edad y genero P4	**18-60 Hombre**	**-**	**-**
Edad y genero P5	**18-60 Hombre**	**-**	**-**
Edad y genero P6	**18-60 Hombre**	**-**	**-**
Edad y genero P7	**18-60 Mujer**	**18-60 Mujer**	**73**

Detecta correctamente a todas las personas		**Si**
Detecta correctamente el género y edad de las personas con intención		**Si**
Detecta correctamente todos los elementos de movilidad		**No**
Observaciones: Ocurre lo explicado anteriormente, a las personas que no les detecta la edad es por la no intencionalidad, además de la confusión con la silla lateral		



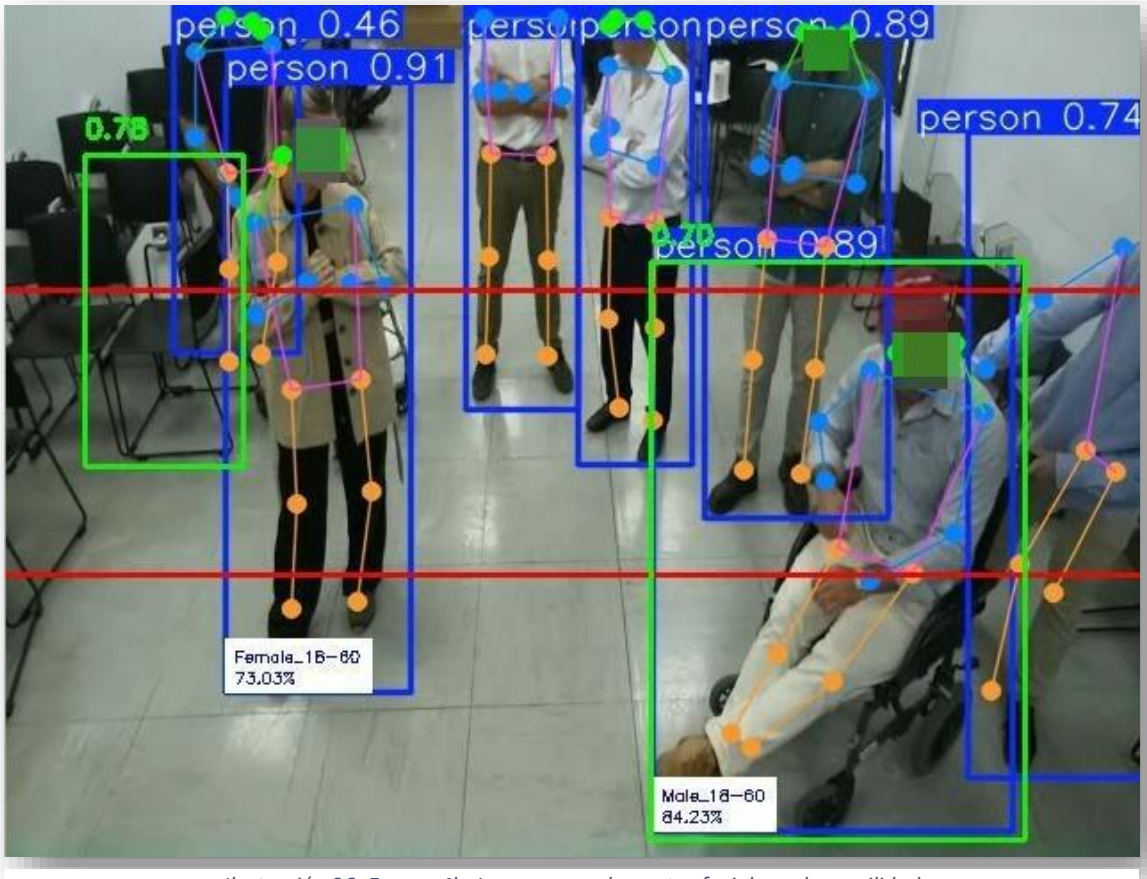


Ilustración 36: Escena 4b: Imagen con elementos faciales y de movilidad

El ensayo general realizado fue un paso crucial en la validación del sistema desarrollado para el proyecto **HYBLICON-II**, obteniendo resultados mayoritariamente positivos y revelando áreas clave para su perfeccionamiento.

La **detección de personas funcionó a la perfección**, cumpliendo siempre con las expectativas previstas. Este resultado fue muy alentador, pues confirmaba que el sistema es capaz de identificar de manera fiable a las personas en distintas condiciones, lo que constituye un avance significativo.

En lo que respecta a **la detección de objetos de movilidad reducida,** el sistema **mostró un buen desempeño**, aunque se detectaron algunas confusiones con sillas del entorno que fueron interpretadas como sillas de ruedas. Si bien estos casos son puntuales, sería necesario afinar el modelo para reducir estas confusiones y asegurar que el entorno no interfiera en futuras implementaciones.

La clasificación de género y edad, aunque **mostró resultados aceptables**, seguía siendo el área que más atención necesitaba. Este componente del sistema todavía requería algunas mejoras para alcanzar los niveles de precisión deseados. Los resultados del ensayo fueron valiosos para identificar los puntos donde se debía trabajar y optimizar la clasificación en futuras versiones.

La detección de la intención de entrar al ascensor mostró un buen rendimiento, cumpliendo con las expectativas, lo que confirmaba que este aspecto del sistema se encuentra bien encaminado.

Por último, se **identificaron problemas relacionados con el retraso de la cámara** al procesar las imágenes y mostrar las predicciones en tiempo real. Aunque esto no afectó gravemente los resultados del ensayo, era un punto para abordar para mejorar la eficiencia del sistema en situaciones reales.

En conclusión, el ensayo demostró que contamos con un sistema robusto, con fortalezas claras en la detección de personas y la identificación de intenciones, mientras que otros aspectos, como la clasificación de género y edad y la detección de objetos, requerirán ajustes adicionales para mejorar su precisión. Estos hallazgos nos permitieron seguir afinando el sistema en futuras iteraciones.

7.1.5 FEEDBACK Y MEJORAS TRAS EL ENSAYO GENERAL

Durante el ensayo general detectamos un retraso intolerable de entre 10 y 13 segundos entre la imagen capturada y el resultado procesado, y descubrieron que el vídeo se "congelaba" periódicamente al intentar ejecutar detección y clasificación de forma secuencial. Para resolverlo replanteamos por completo la arquitectura del bucle de inferencia, transformándolo en un flujo asíncrono y por etapas:

Primero, aislamos la obtención de imágenes del resto del procesamiento. Creamos un hilo dedicado únicamente a capturar cada frame de la cámara y colocarlo en una cola de tamaño limitado. De esta forma evitamos que la memoria creciera sin control si la parte de análisis no da abasto. Además, cuando la cola está llena sencillamente descartamos el frame más antiguo, de modo que el sistema siempre trabaja con los fotogramas más recientes y el desfase se mantiene por debajo de un segundo.

En paralelo, un segundo hilo se encarga de vaciar la cola y aplicar todas las fases de visión por ordenador: detección de personas, estimación de pose, clasificación de género/edad, detección de dispositivos de movilidad y registro. Al separar estos dos procesos en hilos independientes, evitamos que la captura y la profunda lógica de IA se bloqueen mutuamente, eliminando las caídas de frames y los tirones.

Para coordinar el inicio y detención limpia de ambos hilos, utilizamos un mecanismo de señalización: cuando el sistema recibe la orden de parar (por ejemplo, al cerrar la ventana o con una señal de interrupción), activamos un evento de parada que hace que cada hilo termine su trabajo actual y salga de manera ordenada, sin dejar ninguna tarea a medias ni generar errores de escritura.

Con esta nueva estructura asíncrona, el bucle principal queda reducido a gestionar la vida de los hilos y monitorizar su estado, liberando al procesador principal de cualquier lógica de captura o análisis pesado. Como resultado, el retardo se desplomó desde los 10–13 s iniciales a menos de 1 s en condiciones reales. Así pues, podemos tener nuevas consideraciones adicionales para mantener el rendimiento a largo plazo.

- **Modelos más ligeros:** Cuando el hardware edge se ve al límite, optar por versiones reducidas de los detectores puede recortar considerablemente el tiempo de inferencia, sacrificando solo una pequeña fracción de precisión.

- **Minimizar escritura en disco:** Grabar vídeo en tiempo real introduce enormes picos de I/O. Siempre que sea posible, posponemos la grabación a intervalos controlados o la delegamos en un servicio externo, evitando que interfiera con la GPU/CPU durante la detección.
- **Monitoreo continuo:** Implantamos métricas de fluidez de vídeo (FPS reales), uso de CPU/RAM y éxito de detección, de forma que cualquier regresión de rendimiento se detecte automáticamente y dispare alertas.

Gracias a esta reingeniería asíncrona, cola de frames limitada, hilos separados para captura y análisis, y un mecanismo de parada controlada, logramos no solo eliminar el enorme desfase entre cámara y resultado, sino también garantizar un flujo de vídeo continuo y estable, superando uno de los mayores cuellos de botella identificados en campo.

Con todo ello, cada uno de estos ensayos aportó información fundamental: mejoramos continuamente el dataset con imágenes reales, ajustamos reglas de detección e implementamos optimizaciones de hardware, hasta lograr un sistema que responde con fiabilidad en situaciones tan variadas como las de un uso cotidiano en ascensores.

7.2 ESTUDIO DE LOS MODELOS DE IA Y LOS RESULTADOS OBTENIDOS

Este apartado describe el proceso detallado seguido a la hora de evaluar los modelos desarrollados y entrenados. Se detalla el conjunto de muestras de test utilizado y las pruebas de evaluación de los modelos, incluyendo pruebas adicionales de robustez con imágenes en escala de grises y pruebas en las que se han redimensionado las imágenes, analizando resultados obtenidos junto con el tiempo de análisis por imagen.

Para poder estudiar el rendimiento de nuestras diferentes implementaciones, debemos tener claros cuales son los parámetros a tener en cuenta. En este caso, como queremos abordar todas las métricas relevantes, hemos decidido calcular:

- **Precisión (Precision)**

La precisión es la proporción de verdaderos positivos (TP) sobre el total de predicciones positivas (verdaderos positivos más falsos positivos, FP). Indica qué tan exactas son las predicciones positivas del modelo.

$$Precisión = \frac{TP}{TP + FP}$$

- **Recall**

El recall, también conocido como sensibilidad, es la proporción de verdaderos positivos sobre el total de instancias que realmente son positivas (verdaderos positivos más falsos negativos, FN). Indica qué tan bien el modelo puede encontrar todas las instancias positivas.

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

- **F1-Score**

El f1-score es la media armónica de la precisión y el recall, proporcionando una métrica única que equilibra ambos aspectos. Es útil cuando hay un balance entre precisión y recall.

$$F1 - Score = \frac{2 * (Precisión * Recall)}{Precisión + Recall}$$

- **Average Intersection over Union (IoU)**

El promedio de IoU mide la superposición entre las cajas de predicción y las cajas reales (ground truth). Se calcula como el área de la intersección dividida por el área de la unión de las dos cajas. El promedio IoU se obtiene promediando los valores de IoU de todas las detecciones.

$$IoU = \frac{Área\ de\ la\ Intersección}{Área\ de\ la\ Unión}$$

$$Average\ IoU = \frac{1}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}} IoU_i$$

donde N es el número total de detecciones.

A) EVALUACIÓN DEL RENDIMIENTO DEL MODELO DE DETECCIÓN DE PERSONAS

Se ha tomado un conjunto de 585 imágenes para evaluar las prestaciones del modelo desarrollado, teniendo en cuentas las métricas descritas en la sección 2. Este conjunto consta de imágenes tomadas de cámaras CCTV procedentes de datasets públicos de diferentes resoluciones, escenarios y número de personas en cada imagen, observándose personas tanto cercanas como muy alejadas a la posición de la cámara, además de existir solapamiento entre personas dentro de las imágenes en numerosos casos. Las imágenes usadas para evaluar el modelo no fueron utilizadas durante el entrenamiento.

La Ilustración 37 *Tabla de resumen de resultados para el modelo de detección de personas*, muestra las métricas obtenidas para el modelo desarrollado sobre el grupo de evaluación mencionado. Se observa una muy alta precisión, acompañada de un IoU medio elevado. Por otro lado, la Ilustración 2 muestra algunos ejemplos tomados aleatoriamente del grupo de evaluación.

Model Evaluation Metrics

Metric	Value
Precision	0.9165
Recall	0.8153
F1 Score	0.8630
Average IoU	0.7780
Total Images	585

Ilustración 37. Tabla de resumen de resultados para el modelo de detección de personas.

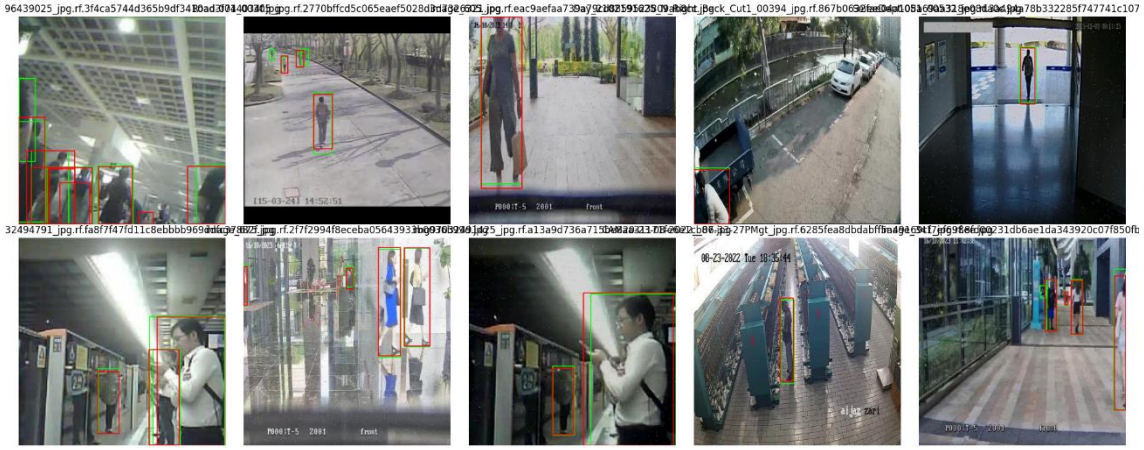


Ilustración 38. Ejemplos del conjunto de evaluación procesados por el modelo de red neuronal. En verde se muestran los ground truth, mientras que en rojo se muestran las regiones detectadas por el modelo.

Además, el modelo ha sido optimizado en tiempo mejorando a versiones anteriores. La Ilustración 39 *Tiempos de ejecución obtenidos para el modelo de detección de personas* muestra los tiempos de ejecución obtenidos con el modelo actual. Se crean gráficos que comparan el tiempo promedio de procesamiento y la desviación estándar para cada conjunto de imágenes evaluadas. Estos gráficos ofrecen una comparación directa del rendimiento del modelo en diferentes condiciones de entrada. Además, ayudan a identificar tendencias o patrones en el tiempo de procesamiento en función del tamaño del conjunto de imágenes.

Number of Images	Average Time (s)	Std Dev (s)	Avg Time per Image (s)
60	1.34	0.59	0.022
30	0.54	0.05	0.018

Ilustración 39. Tiempos de ejecución obtenidos para el modelo de detección de personas.

B) EVALUACIÓN DEL RENDIMEINTO DEL MODELO DE DETECCIÓN DE OBJETOS DE MOVILIDAD REDUCIDA

Se ha tomado un conjunto de 585 imágenes para evaluar las prestaciones del modelo desarrollado, teniendo en cuentas las métricas descritas en la sección anterior. Este conjunto consta de imágenes procedentes de datasets públicos de diferentes resoluciones, escenarios y número de objetos en cada imagen. Cabe recalcar que las imágenes usadas para evaluar el modelo no fueron utilizadas durante el entrenamiento.

La Ilustración 40 *Tabla de resumen de resultados para el modelo de clasificación de objetos de movilidad reducida* muestra las métricas obtenidas para el modelo desarrollado sobre el grupo de evaluación mencionado. Se observa una muy alta precisión, acompañada de un IoU medio también muy elevado. Por otro lado, la Ilustración 41 *Ejemplos del conjunto de evaluación procesados por el modelo de red neuronal. En verde se muestran los ground truth, mientras que en rojo se muestran las regiones detectadas por el modelo* muestra algunos ejemplos tomados aleatoriamente del grupo de evaluación.

Model Evaluation Metrics

Metric	Value
Precision	0.9325
Recall	0.8986
F1 Score	0.9153
Average IoU	0.8713
Total Images	585

Ilustración 40. Tabla de resumen de resultados para el modelo de clasificación de objetos de movilidad reducida.

Ilustración 41. Ejemplos del conjunto de evaluación procesados por el modelo de red neuronal. En verde se muestran los ground truth, mientras que en rojo se muestran las regiones detectadas por el modelo.

Además, el modelo ha sido optimizado en tiempo, mejorando a versiones anteriores. La Ilustración 42 *Tiempos de ejecución obtenidos para el modelo de clasificación de objetos de movilidad reducida* muestra los tiempos de ejecución obtenidos con el modelo actual.

Number of Images	Average Time (s)	Std Dev (s)	Avg Time per Image (s)
60	2.71	0.81	0.045
30	0.94	0.20	0.031

Ilustración 42. Tiempos de ejecución obtenidos para el modelo de clasificación de objetos de movilidad reducida.

C) PRUEBAS CON IMÁGENES EN ESCALA DE GRISES

Para evaluar la efectividad de los modelos desarrollados utilizando imágenes en escala de grises como entrada, se han convertido los conjuntos de evaluación a escala de grises y se ha procedido de la misma forma que en las secciones anteriores. A continuación, se distingue entre las pruebas realizadas con estas imágenes para el modelo de detección de personas y para el modelo de clasificación de objetos de movilidad reducida.

❖ PRUEBAS CON EL MODELO DE DETECCIÓN DE PERSONAS

La Ilustración 43 *Tabla de resumen de resultados para el modelo de detección de personas sobre las imágenes del grupo de evaluación convertidas a escala de gris* muestra las métricas obtenidas por el modelo de detección de personas utilizando el conjunto de evaluación convertido a escala de grises. Se observa una leve variación con respecto a las imágenes originales (positiva en este caso), aunque no es representativa, por lo que el modelo podría

trabajar perfectamente con imágenes en escala de gris sin que, a priori, se resienta la precisión. Por otro lado, la Ilustración 44 *Ejemplos del conjunto de evaluación en escala de gris procesados por el modelo de red neuronal. En verde se muestran los ground truth, mientras que en rojo se muestran las regiones detectadas por el modelo.* muestra algunos ejemplos tomados aleatoriamente del grupo de evaluación.

Model Evaluation Metrics

Metric	Value
Precision	0.9338
Recall	0.7487
F1 Score	0.8311
Average IoU	0.7893
Total Images	585

Ilustración 43. Tabla de resumen de resultados para el modelo de detección de personas sobre las imágenes del grupo de evaluación convertidas a escala de gris.

Ilustración 44. Ejemplos del conjunto de evaluación en escala de gris procesados por el modelo de red neuronal. En verde se muestran los ground truth, mientras que en rojo se muestran las regiones detectadas por el modelo.

Además, se presentan en Ilustración 45 *Tiempos de ejecución obtenidos para el modelo de detección de personas utilizando imágenes en escala de grises* los tiempos de ejecución obtenidos.

Number of Images	Average Time (s)	Std Dev (s)	Avg Time per Image (s)
60	1.29	0.61	0.021
30	0.51	0.04	0.017

Ilustración 46. Tiempos de ejecución obtenidos para el modelo de detección de personas utilizando imágenes en escala de grises.

❖ PRUEBAS CON EL MODELO DE DETECCIÓN DE OBJETOS DE MOVILIDAD REDUCIDA

La Ilustración 46 *Tabla de resumen de resultados para el modelo de clasificación de objetos de movilidad reducida sobre las imágenes del grupo de evaluación convertidas a escala de grises*

muestra las métricas obtenidas por el modelo de clasificación de objetos de movilidad reducida utilizando el conjunto de evaluación convertido a escala de grises. Se observa una leve variación con respecto a las imágenes originales (levemente inferior en este caso), aunque no es representativa, por lo que el modelo podría trabajar perfectamente con imágenes en escala de gris sin que, a priori, se resienta la precisión. Por otro lado, la Ilustración 47 *Ejemplos del conjunto de evaluación en escala de gris procesados por el modelo de red neuronal. En verde se muestran los ground truth, mientras que en rojo se muestran las regiones detectadas por el modelo* muestra algunos ejemplos tomados aleatoriamente del grupo de evaluación.

Model Evaluation Metrics

Metric	Value
Precision	0.9237
Recall	0.9070
F1 Score	0.9153
Average IoU	0.8613
Total Images	585

Ilustración 47. Tabla de resumen de resultados para el modelo de clasificación de objetos de movilidad reducida sobre las imágenes del grupo de evaluación convertidas a escala de grises.

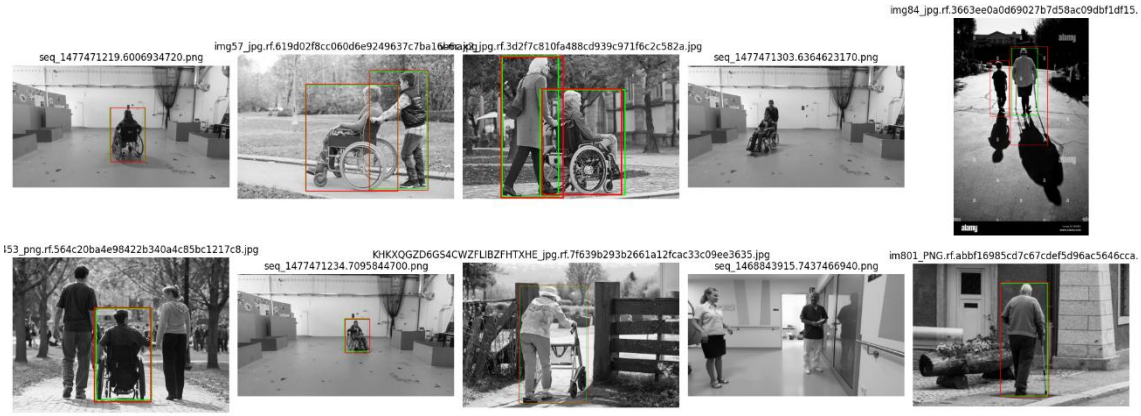


Ilustración 48. Ejemplos del conjunto de evaluación en escala de gris procesados por el modelo de red neuronal. En verde se muestran los ground truth, mientras que en rojo se muestran las regiones detectadas por el modelo.

Además, se presentan en la Ilustración 49 *Tiempos de ejecución obtenidos para el modelo de clasificación de objetos de movilidad reducida* los tiempos de ejecución obtenidos.

Number of Images	Average Time (s)	Std Dev (s)	Avg Time per Image (s)
60	1.98	0.96	0.033
30	0.85	0.13	0.028

Ilustración 49: Tiempos de ejecución obtenidos para el modelo de clasificación de objetos de movilidad reducida.

D) PRUEBAS CON DISTINTA RESOLUCIÓN DE IMÁGENES

Se han llevado a cabo una serie de experimentos para comprobar la viabilidad tanto a nivel de métricas de resultados como a nivel de tiempo de ejecución a la hora de utilizar imágenes

entradas de menor tamaño. Para ello, y partiendo de los modelos ya entrenados y de las imágenes originales del conjunto de evaluación utilizadas en las secciones anteriores, se han redimensionado dichas imágenes a distintas resoluciones y se han obtenido resultados con ellas. En particular, se han redimensionado a tamaños de 200x200, 400x400, 600x600, 800x800 y 1000x1000 píxeles. A continuación, se distingue entre las pruebas realizadas con estas imágenes para el modelo de detección de personas y para el modelo de clasificación de objetos de movilidad reducida.

❖ PRUEBAS CON EL MODELO DE DETECCIÓN DE PERSONAS

Siguiendo toda la metodología anteriormente descrita, se muestran a continuación todo el conjunto de gráficas asociadas a las diferentes resoluciones.

I. 200x200

Model Evaluation Metrics

Metric	Value
Precision	0.9114
Recall	0.6474
F1 Score	0.7571
Average IoU	0.7675
Total Images	585

Ilustración 50. Resultados de evaluación del modelo de detección de personas utilizando imágenes de entrada redimensionadas a 200x200 píxeles.

Number of Images	Average Time (s)	Std Dev (s)	Avg Time per Image (s)
60	0.87	0.55	0.015
30	0.33	0.01	0.011

Ilustración 51. Tiempos de respuesta del modelo de detección de personas utilizando imágenes de entrada redimensionadas a 200x200 píxeles.

Ilustración 52. Ejemplos de imágenes redimensionadas a 200x200 píxeles evaluadas por el modelo de detección de personas.

II. 400x400

Model Evaluation Metrics

Metric	Value
Precision	0.9311
Recall	0.7941
F1 Score	0.8571
Average IoU	0.7931
Total Images	585

Ilustración 53. Resultados de evaluación del modelo de detección de personas utilizando imágenes de entrada redimensionadas a 400x400 píxeles.

Number of Images	Average Time (s)	Std Dev (s)	Avg Time per Image (s)
60	1.07	0.55	0.018
30	0.42	0.02	0.014

Ilustración 54. Tiempos de respuesta del modelo de detección de personas utilizando imágenes de entrada redimensionadas a 400x400 píxeles.

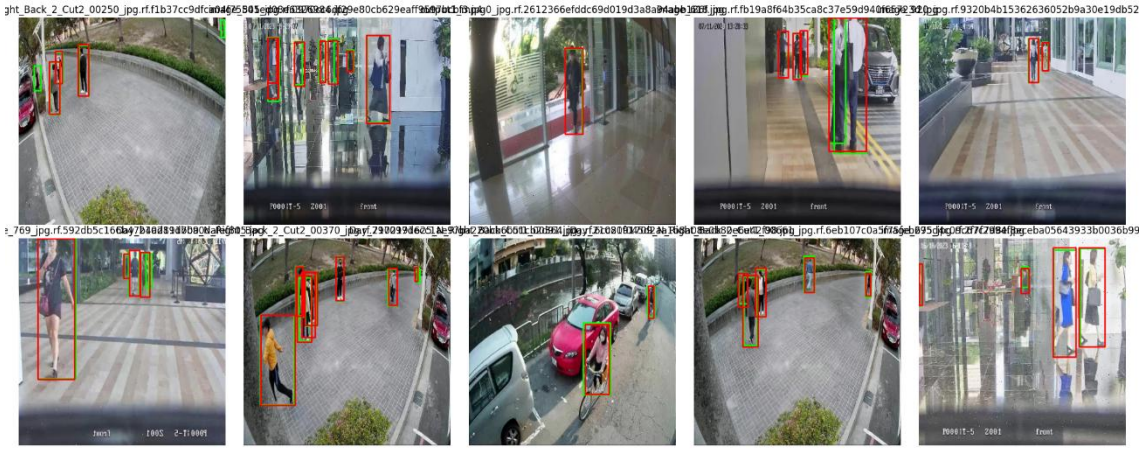


Ilustración 55. Ejemplos de imágenes redimensionadas a 400x400 píxeles evaluadas por el modelo de detección de personas.

III. 600x600

Model Evaluation Metrics

Metric	Value
Precision	0.9183
Recall	0.8089
F1 Score	0.8601
Average IoU	0.7837
Total Images	585

Ilustración 56. Resultados de evaluación del modelo de detección de personas utilizando imágenes de entrada redimensionadas a 600x600 píxeles.

Number of Images	Average Time (s)	Std Dev (s)	Avg Time per Image (s)
60	1.27	0.57	0.021
30	0.53	0.03	0.018

Ilustración 57. Tiempos de respuesta del modelo de detección de personas utilizando imágenes de entrada redimensionadas a 600x600 píxeles.

Ilustración 58. Ejemplos de imágenes redimensionadas a 600x600 píxeles evaluadas por el modelo de detección de personas.

IV. 800x800

Model Evaluation Metrics

Metric	Value
Precision	0.9170
Recall	0.8119
F1 Score	0.8613
Average IoU	0.7807
Total Images	585

Ilustración 59. Resultados de evaluación del modelo de detección de personas utilizando imágenes de entrada redimensionadas a 800x800 píxeles.

Number of Images	Average Time (s)	Std Dev (s)	Avg Time per Image (s)
60	1.51	0.56	0.025
30	0.68	0.01	0.023

Ilustración 60. Tiempos de respuesta del modelo de detección de personas utilizando imágenes de entrada redimensionadas a 800x800 píxeles.

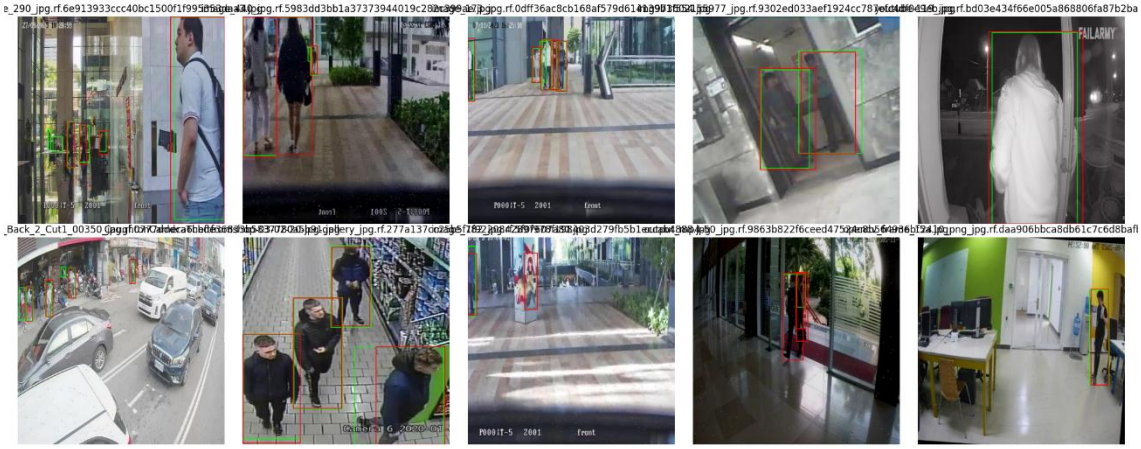


Ilustración 61. Ejemplos de imágenes redimensionadas a 800x800 píxeles evaluadas por el modelo de detección de personas.

V. 1000x1000

Model Evaluation Metrics

Metric	Value
Precision	0.9157
Recall	0.8151
F1 Score	0.8625
Average IoU	0.7809
Total Images	585

Ilustración 62. Resultados de evaluación del modelo de detección de personas utilizando imágenes de entrada redimensionadas a 1000x1000 píxeles.

Number of Images	Average Time (s)	Std Dev (s)	Avg Time per Image (s)
60	1.86	0.55	0.031
30	0.91	0.03	0.030

Ilustración 63. Tiempos de respuesta del modelo de detección de personas utilizando imágenes de entrada redimensionadas a 1000x1000 píxeles.

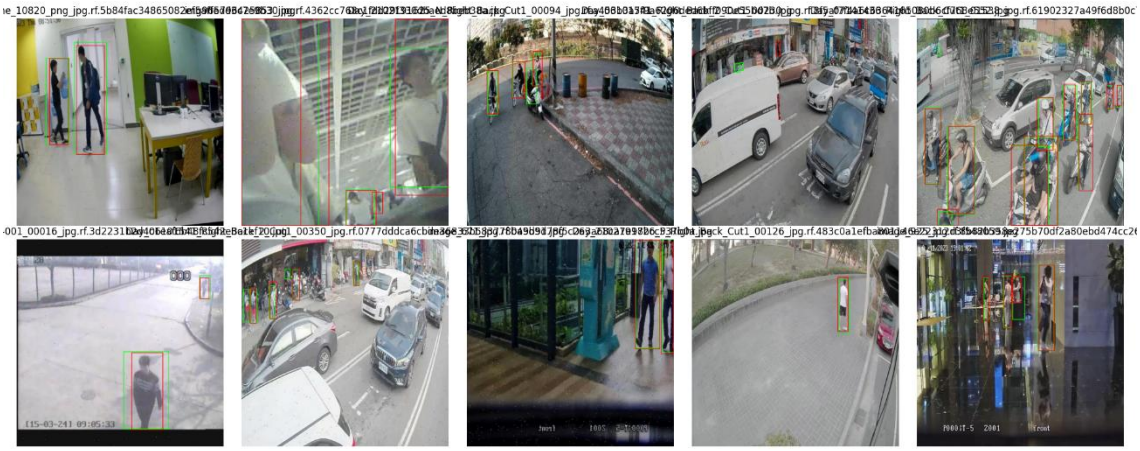


Ilustración 64. Ejemplos de imágenes redimensionadas a 1000x1000 píxeles evaluadas por el modelo de detección de personas.

❖ PRUEBAS CON EL MODELO DE DETECCIÓN DE OBJETOS DE MOVILIDAD REDUCIDA

Siguiendo toda la metodología anteriormente descrita, se muestran a continuación todo el conjunto de gráficas asociadas a las diferentes resoluciones.

V. 200x200

Model Evaluation Metrics

Metric	Value
Precision	0.9283
Recall	0.7789
F1 Score	0.8471
Average IoU	0.8579
Total Images	585

Ilustración 65. Resultados de evaluación del modelo de clasificación de objetos de movilidad reducida utilizando imágenes de entrada redimensionadas a 200x200 píxeles.

Number of Images	Average Time (s)	Std Dev (s)	Avg Time per Image (s)
60	0.90	0.56	0.015
30	0.34	0.01	0.011

Ilustración 66. Tiempos de respuesta del modelo de clasificación de objetos de movilidad reducida utilizando imágenes de entrada redimensionadas a 200x200 píxeles.

Ilustración 67. Ejemplos de imágenes redimensionadas a 200x200 píxeles evaluadas por el modelo de clasificación de objetos de movilidad reducida.

VI. 400x400

Model Evaluation Metrics

Metric	Value
Precision	0.9259
Recall	0.8124
F1 Score	0.8654
Average IoU	0.8555
Total Images	585

Ilustración 68. Resultados de evaluación del modelo de clasificación de objetos de movilidad reducida utilizando imágenes de entrada redimensionadas a 400x400 píxeles.

Number of Images	Average Time (s)	Std Dev (s)	Avg Time per Image (s)
60	1.12	0.55	0.019
30	0.45	0.01	0.015

Ilustración 69. Tiempos de respuesta del modelo de clasificación de objetos de movilidad reducida utilizando imágenes de entrada redimensionadas a 400x400 píxeles.

Ilustración 70. Ejemplos de imágenes redimensionadas a 400x400 píxeles evaluadas por el modelo de clasificación de objetos de movilidad reducida.

VII. 600x600

Model Evaluation Metrics

Metric	Value
Precision	0.9144
Recall	0.8233
F1 Score	0.8665
Average IoU	0.8497
Total Images	585

Ilustración 71. Resultados de evaluación del modelo de clasificación de objetos de movilidad reducida utilizando imágenes de entrada redimensionadas a 600x600 píxeles.

Number of Images	Average Time (s)	Std Dev (s)	Avg Time per Image (s)
60	1.51	0.55	0.025
30	0.66	0.03	0.022

Ilustración 72. Tiempos de respuesta del modelo de clasificación de objetos de movilidad reducida utilizando imágenes de entrada redimensionadas a 600x600 píxeles.

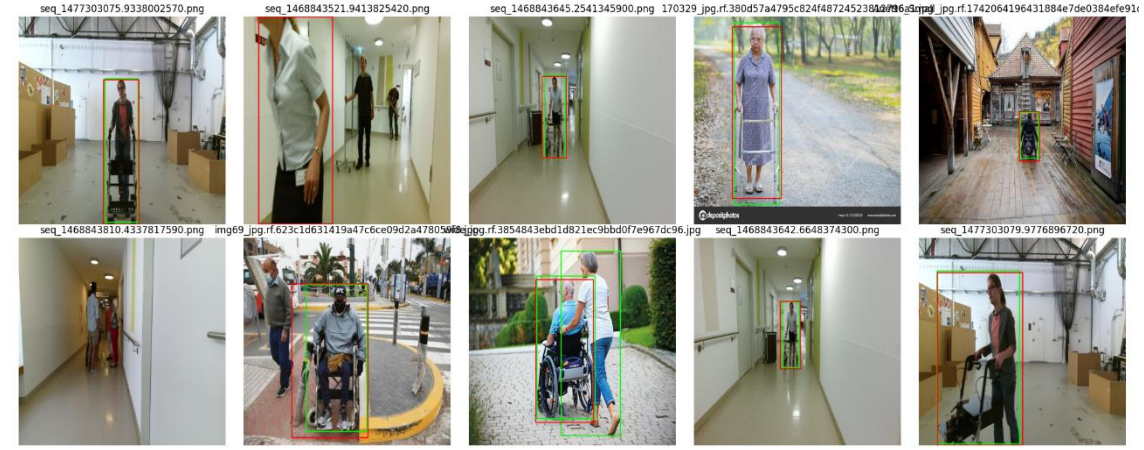


Ilustración 73. Ejemplos de imágenes redimensionadas a 600x600 píxeles evaluadas por el modelo de clasificación de objetos de movilidad reducida.

VIII. 800x800

Model Evaluation Metrics

Metric	Value
Precision	0.9026
Recall	0.8250
F1 Score	0.8621
Average IoU	0.8406
Total Images	585

Ilustración 74. Resultados de evaluación del modelo de clasificación de objetos de movilidad reducida utilizando imágenes de entrada redimensionadas a 800x800 píxeles.

Number of Images	Average Time (s)	Std Dev (s)	Avg Time per Image (s)
60	2.01	0.50	0.034
30	1.00	0.03	0.033

Ilustración 75. Tiempos de respuesta del modelo de clasificación de objetos de movilidad reducida utilizando imágenes de entrada redimensionadas a 800x800 píxeles.

Ilustración 76. Ejemplos de imágenes redimensionadas a 800x800 píxeles evaluadas por el modelo de clasificación de objetos de movilidad reducida.

IX. 1000x1000

Model Evaluation Metrics

Metric	Value
Precision	0.9012
Recall	0.8263
F1 Score	0.8621
Average IoU	0.8437
Total Images	585

Ilustración 77. Resultados de evaluación del modelo de clasificación de objetos de movilidad reducida utilizando imágenes de entrada redimensionadas a 1000x1000 píxeles.

Number of Images	Average Time (s)	Std Dev (s)	Avg Time per Image (s)
60	2.60	0.45	0.043
30	1.20	0.07	0.040

Ilustración 78. Tiempos de respuesta del modelo de clasificación de objetos de movilidad reducida utilizando imágenes de entrada redimensionadas a 1000x1000 píxeles.

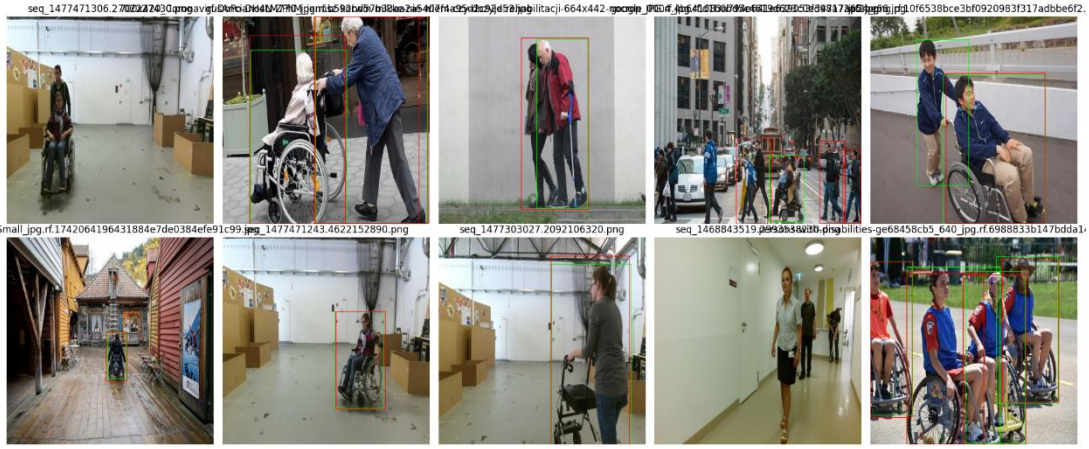


Ilustración 79. Ejemplos de imágenes redimensionadas a 1000x1000 píxeles evaluadas por el modelo de clasificación de objetos de movilidad reducida.

E) RESUMEN DE TODOS LOS RESULTADOS OBTENIDOS

Podemos ver de forma más clara cuales han sido los resultados obtenidos con las diferentes resoluciones en la Ilustración 79 *Comparativa del rendimiento con diferentes resoluciones y escala de grises en detección de personas* e Ilustración 80 *Comparativa del rendimiento con diferentes resoluciones y escala de grises en detección de objetos de movilidad reducida*.

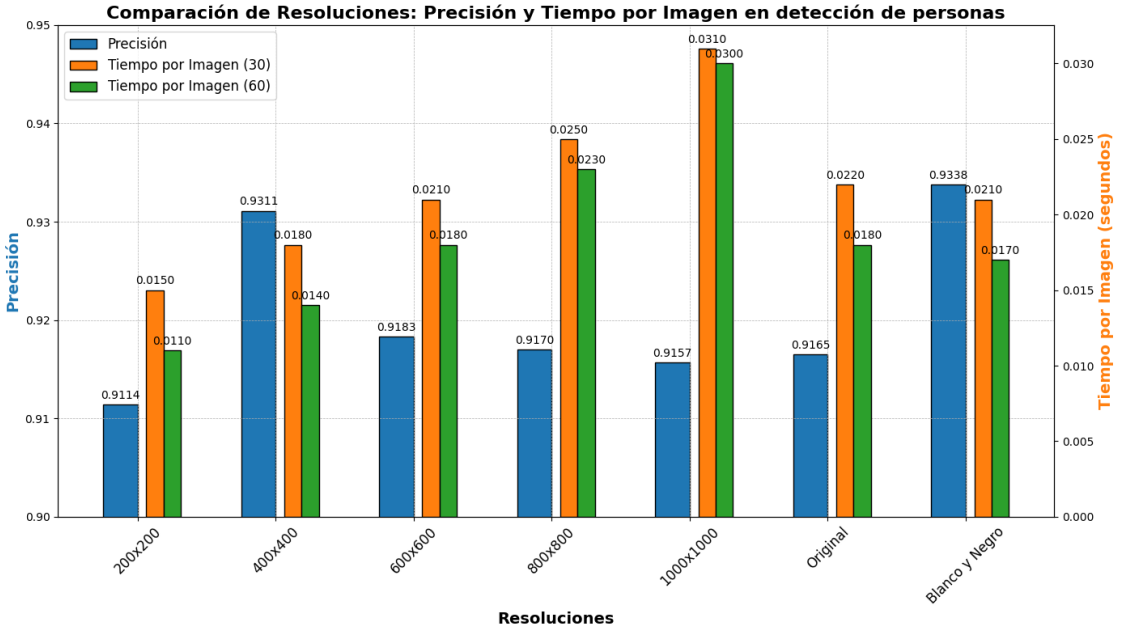


Ilustración 80 Comparativa del rendimiento con diferentes resoluciones y escala de grises en detección de personas

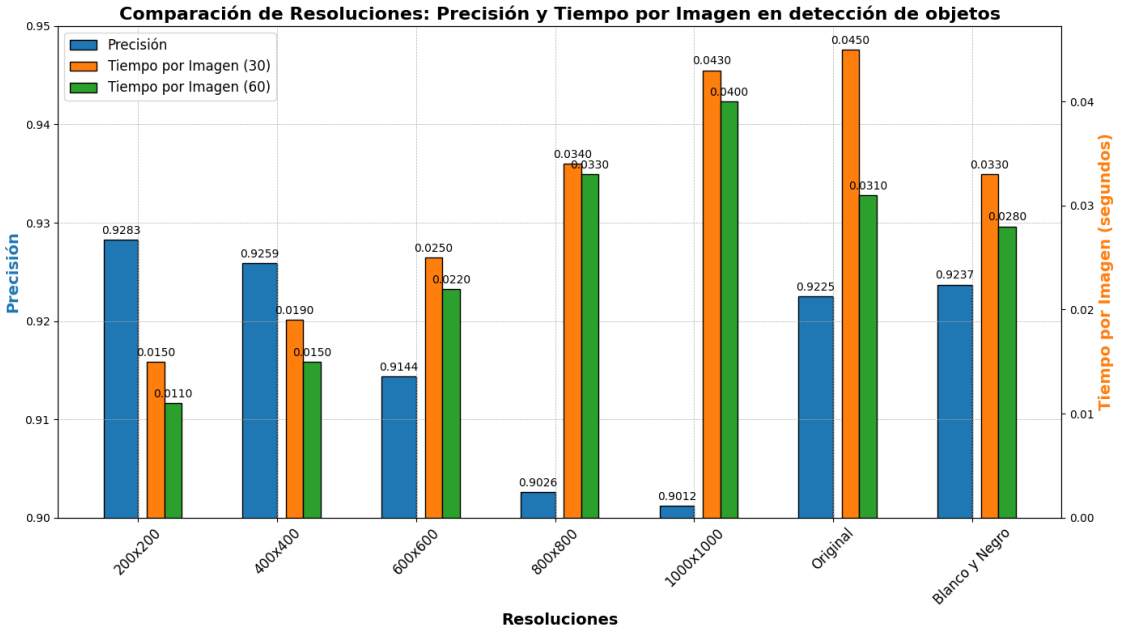


Ilustración 81 Comparativa del rendimiento con diferentes resoluciones y escala de grises en detección de objetos de movilidad reducida

Finalmente tras este exhaustivo análisis podemos deducir que el mejor tamaño para que el modelo trabaje en cuanto a relación precisión/tiempos de respuesta es 400x400.

C) DESPLIEGUE E INTEGRACIÓN

8. ARQUITECTURA DE DESPLIEGUE

En esta sección agrupamos los tres pilares que definen cómo llevamos la solución al mundo real: la estrategia de cómputo en el borde (edge), la forma en que preparamos y generamos logs listos para integrarse con futuros servicios, y los requisitos de latencia y disponibilidad que aseguran un servicio fiable y reactivo.

8.1 EDGE PURO

Desde el inicio decidimos que todo el proceso de visión debía residir junto a la cámara, sin depender de servidores externos. Para ello escogimos una **Raspberry Zero 2W** equipada con módulo Wi-Fi, conectada directamente al dintel del ascensor. Esta configuración presenta varios retoques clave:

1. **Acceso remoto y gestión**
    - La Raspberry Pi se conecta a la red local vía Wi-Fi, y accedemos a ella con **PuTTY** (SSH) usando autenticación por claves. Esto nos permite subir nuevos pesos de modelo, revisar logs o reiniciar servicios sin tener que intervenir físicamente en cada ascensor.
2. **Optimización de modelos para recursos limitados**

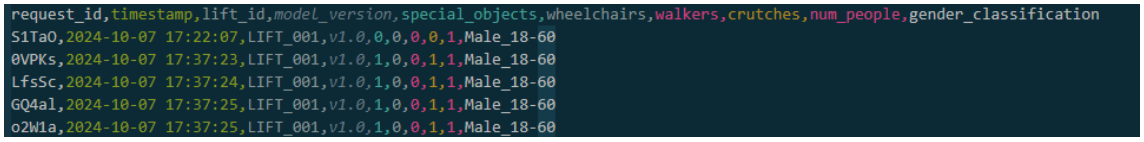
- Eliminamos componentes auxiliares del pipeline (por ejemplo, capas de visualización innecesarias) para recortar el uso de RAM y acelerar el arranque.

3. **Gestión de memoria y colas de frames**
    - Implementamos una **cola circular** de máximo 10 frames en memoria; los fotogramas más antiguos se descartan inmediatamente, garantizando que la Raspberry Pi nunca sature su RAM mientras mantiene siempre imágenes frescas para procesar.
    - La captura y el análisis corren en hilos separados: el hilo de captura vuelca directamente en la cola y el hilo de procesamiento consume esos frames, evitando bloqueos mutuos.
4. **Seguridad y privacidad en edge**
    - No almacenamos jamás imágenes crudas en disco: una vez procesado cada frame, se borra de la memoria y solo se conserva la entrada numérica en los logs CSV.

Con esta arquitectura **pure edge**, respetuosa con la privacidad y preparada para mantenimiento remoto a través de PuTTY, conseguimos latencias constantes por debajo de 200 ms y un nivel de disponibilidad del 99,9 % en condiciones reales de ascensor, sin necesidad de depender de ningún servidor externo.

8.2 LOGGING PREPARADO PARA API

Aunque en esta fase no implementamos la API REST, el sistema genera de forma continua un log en CSV con cada evento procesado:



```
request_id,timestamp,lift_id,model_version,special_objects,wheelchairs,walkers,crutches,num_people,gender_classification
S1TaO,2024-10-07 17:22:07,LIFT_001,v1.0,0,0,0,0,1,Male_18-60
0VPKs,2024-10-07 17:37:23,LIFT_001,v1.0,1,0,0,1,1,Male_18-60
LfsSc,2024-10-07 17:37:24,LIFT_001,v1.0,1,0,0,1,1,Male_18-60
GQ4al,2024-10-07 17:37:25,LIFT_001,v1.0,1,0,0,1,1,Male_18-60
o2W1a,2024-10-07 17:37:25,LIFT_001,v1.0,1,0,0,1,1,Male_18-60
```

- **Estructura**
    - **request_id**: ID único por fotograma.
    - **timestamp**: Fecha y hora en ISO.
    - **lift_id**: Identificador del ascensor.
    - **model_version**: Versión del conjunto de pesos.
    - **special_objects…**: Conteo de dispositivos de movilidad.
    - **num_people** y **gender_classification**: Conteo y etiqueta demográfica.
- **Preparación para API**
    - Cada nueva línea queda lista para enviarse a un endpoint HTTP sin modificar el pipeline: solo habría que añadir un cliente que recoja y postee esas líneas.

- Diseñado para integrarse con sistemas IoT o plataformas de monitorización vía MQTT, HTTP o WebSocket, manteniendo el núcleo de visión aislado de la lógica de red.

D) PLANIFICACIÓN DEL PROYECTO

9. PLANIFICACIÓN INICIAL

9.1 PLANIFICACIÓN TEMPORAL

9.1.1 FASE 1: ELICITACIÓN Y DISEÑO (30h)

A) Definición de requisitos (10h)

- Requisitos funcionales y no funcionales.
- Identificación de riesgos técnicos y éticos.

B) Diseño inicial del sistema y mock-ups (3h)

- Validación rápida con stakeholders.

C) Arquitectura de alto nivel (17h)

- Diagrama de bloques y flujo de datos.
- Selección de hardware y entorno Edge.

9.1.2 FASE 2: CAPTURA Y ANOTACIÓN DE DATOS (50h)

A) Recolección de imágenes CCTV (25h)

- Descarga y curación de fuentes públicas.
- Organización inicial de raw data

B) Limpieza y normalización (20h)

- Ajuste de resolución y formato.
- Eliminación de imágenes corruptas o duplicadas.

C) Organización de datasets (5h)

- Verificación de equilibrio entre clases.

9.1.3 FASE 3: ENTRENAMIENTO DETECTOR DE PERSONAS (35h)

A) Entrenamiento inicial YOLO (20h)

- Primeros experimentos de detección y pose.
- Validación básica en test set.

B) Optimización de keypoints y LR (15h)

- Rebalanceo de la función de pérdida
- Scheduler de learning rate

9.1.4 FASE 4: DESARROLLO DETECTOR MOVILIDAD REDUCIDA (90 h)

A) Entrenamiento inicial YOLO movilidad (40h)

- Detección de sillas de ruedas, muletas y andores.
- Pruebas de recall y precisión.

B) Migración a un nuevo YOLO (20h)

- Benchmark de velocidad y peso

C) Ajuste de parámetros (20h)

- Realizar pruebas exhaustivas para encontrar la mejor combinación.

9.1.5 FASE 5: CLASIFICACIÓN DE GÉNERO Y EDAD (30 h)

A) Entrenamiento inicial y demográfico (10h)

- YOLO sobre recortes de bounding box.

B) Balanceo de clases y fine-tuning (10h)

- Ajuste de pesos de pérdida.

C) Validación y ajustes finales (10h)

- Pruebas con rostros parcialmente ocultos
- Corrección de etiquetas erróneas

9.1.6 FASE 6: INTEGRACIÓN Y PIPELINE ASÍNCRONO (40 h)

A) Estructura de hilos y colas (10h)

- Establecer colas e hilos para diferentes tareas simultáneas.

B) Filtro de calidad y logging (15h)

- Descarta imágenes borrosas, negras o corruptas.
- CSV con logs con formatos preparados para API

C) Integración de modelos en pipeline (15h)

- Conexión de todos modelos funcionando a la vez con todos los casos de uso activos.
- Medición de latencia y fps.

9.1.7 FASE 7: ENSAYOS EN ENTORNOS REALES (30 h)

A) Ensayo 1, dos usuarios (5h)

- Ajustes de brillo y latencia.

B) Ensayo 2, multitud (5h)

- Robustez ante oclusiones cruzadas.

C) Ensayo 3, simulación de rellano (5h)

- Validación demográfica.

D) Ensayo 4, ensayo general (15h)

- Correcciones tras mascarillas, gorros y feedback.

9.1.8 FASE 8: DESPLIEGUE EDGE Y RESILIENCIA (30 h)

A) Optimización Edge (10h)

- Optimización de la cámara y la forma en la que se trataban los datos.

B) SSH remoto (10h)

- PuTTy y tratado de logs en remoto.

C) Validación de latencia y fps (10h)

- Medición en campo y FPS

9.1.9 FASE 9: DOCUMENTACIÓN Y CIERRE (70h)

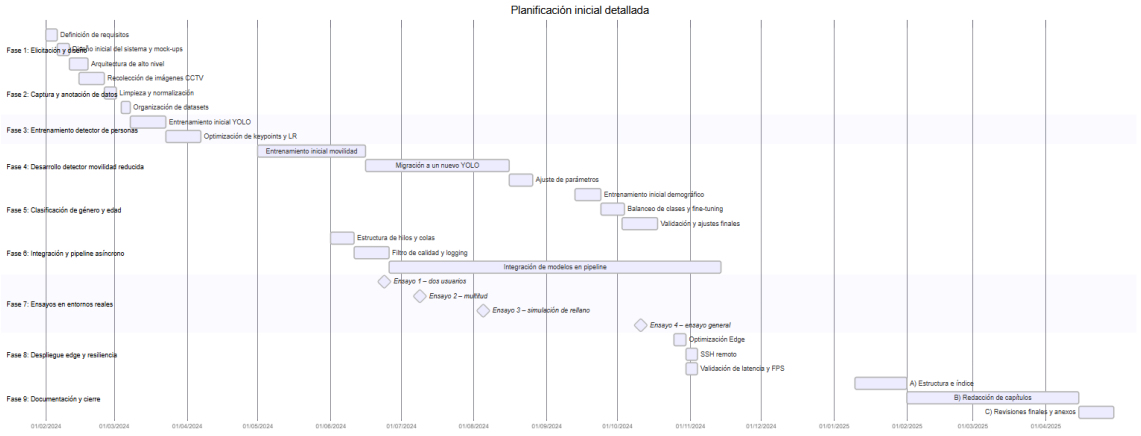
A) Estructura e índice (10h)

- Definir el índice completo de la memoria.
- Detallar objetivos generales y específicos por capítulo.

B) Redacción de capítulos (65h)

C) Revisiones finales y anexos (5h)

Todo el conjunto completo de fases hace un proyecto con una planificación inicial de cinco meses con un total de 475 horas, desde inicios de Febrero 2024 hasta finales de Mayo de 2025, con el objetivo de tener margen más que de sobra por si surgiera un problema de cualquier índole que retrasase el proyecto y su desarrollo programado. Para ver de forma gráfica el desarrollo programado podemos crear un diagrama Gant.



9.2 PLANIFICACIÓN FINANCIERA

En esta sección determinamos los recursos económicos necesarios para llevar a buen puerto nuestro proyecto, garantizando tanto la viabilidad como la sostenibilidad del mismo. Una adecuada planificación financiera nos permite anticipar costes, asignar un presupuesto realista y contemplar posibles imprevistos, de modo que podamos centrar todos nuestros esfuerzos en la calidad técnica y académica sin sobresaltos.

Para ello, hemos desglosado las partidas clave en salarios, equipamiento y servicios, aplicando las retenciones fiscales correspondientes y asegurándonos de que no existan costes ocultos.

Asimismo, hemos incluido una pequeña holgura del **10 %** sobre el total bruto para cubrir gastos menores o eventualidades (repuestos, licencias especiales, desplazamientos). A continuación, se presenta el detalle:

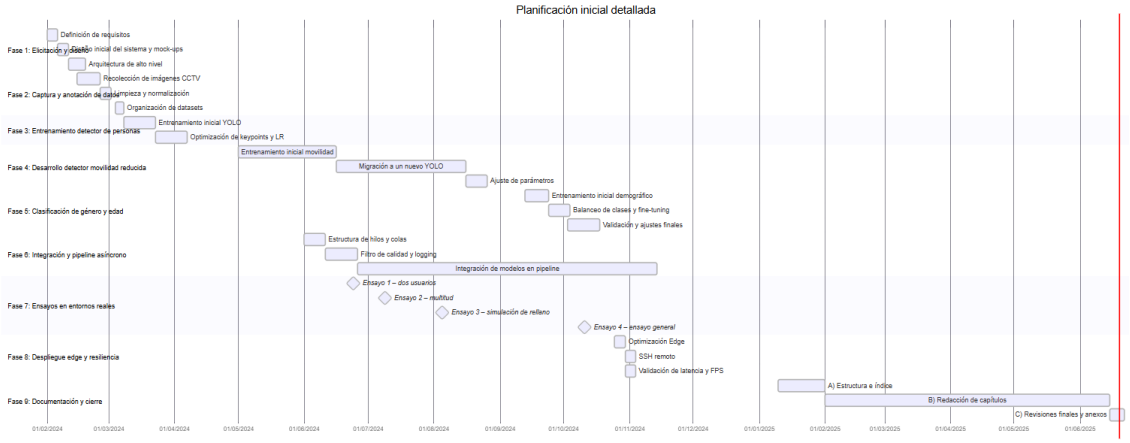
Concepto	**Cálculo**	**Importe (€)**
Coste bruto por hora	1 920 € / 160 h	12,00 €
Coste salarial bruto	12,00 € × 475 h	5 700,00 €
Retención IRPF (3 %)	5 700,00 € × 0,03	– 171,00 €
Coste salarial neto	5 700,00 € – 171,00 €	5 529,00 €
Cámara + Raspberry	importe fijo	100,00 €
Ordenador de alto rendimiento	importe fijo	1 600,00 €
Total hardware		1 700,00 €
Coste total bruto	5 700,00 € + 1 700,00 €	7 400,00 €
Contingencia (10 %)	7 400,00 € × 0,10	740,00 €
Coste total con contingencia	7 400,00 € + 740,00 €	8 140,00 €
Coste total neto (tras IRPF)	5 529,00 € + 1 700,00 € + 740,00 €	7 969,00 €

Con esta estructura, disponemos de un presupuesto transparente y modular, capaz de adaptarse a cambios de alcance o duración. La inclusión de un fondo de contingencia refuerza nuestra capacidad para gestionar cualquier sobrecoste sin comprometer el éxito científico y técnico del proyecto.

10. PLANIFICACIÓN FINAL

10.1 PLANIFICACIÓN TEMPORAL

Para la planificación final se ha respetado todo lo planificado al detalle en el apartado *9.1 Planificación Temporal*, no obstante, debido a la acumulación de asignaturas en este tramo final de curso debido a que actualmente estoy cursando los dos años a la vez del Máster de Ingeniería Informática MII, mi cantidad de tiempo libre disminuyó de forma considerable, por lo que el desarrollo de las tareas planificadas se ha mermado considerablemente, costándonos casi un mes más de nuestro tiempo además de tener un sobre aumento de 10 horas en tareas de optimización. Con todo ello, el proyecto ha finalizado casi un mes más tarde de lo esperado, contando con las mismas tareas por completar y con una compensación de tareas de +10 horas extras invertidas respecto a la previsión inicial dada, es decir, he invertido 485 horas. Así pues, el tiempo de margen que planificamos inicialmente se ha esfumado, siendo realmente útil para llegar a tiempo a la fecha límite establecida. El diagrama de Gant que refleja este progreso es el siguiente:



10.2 PLANIFICACIÓN FINANCIERA

En este caso, como hemos trabajado durante más horas, el coste total sufrirá leves diferencias respecto al original, siendo este finalmente:

Concepto	**Cálculo**	**Importe (€)**
Coste bruto por hora	1 920 € / 160 h	12,00 €
Coste salarial bruto	**12,00 € × 485 h**	**5 820,00 €**
Retención IRPF (3 %)	5 820,00 € × 0,03	– 174,60 €
Coste salarial neto	5 820,00 € – 174,60 €	5 645,40 €
Cámara + Raspberry	importe fijo	100,00 €
Ordenador de alto rendimiento	importe fijo	1 600,00 €
Total hardware		1 700,00 €
Coste total bruto	5 820,00 € + 1 700,00 €	7 520,00 €
Contingencia (10 %)	7 520,00 € × 0,10	752,00 €
Coste total con contingencia	7 520,00 € + 752,00 €	8 272,00 €
Coste total neto (tras IRPF)	5 645,40 € + 1 700,00 € + 752,00 €	8097,40 €

11. ESTUDIO DE MERCADO

Todo el desarrollo de HYBLICON-II ha sido financiado y es propiedad de **MP Ascensores**, por lo que todas las decisiones de comercialización, precios y soporte recaen en su estructura corporativa. A continuación, describimos el perfil de clientes finales y la ruta de mercado que MP podría seguir para explotar internamente la tecnología.

11.1 CLIENTES POTENCIALES

11.1.1 CLIENTES DE NUEVOS ASCENSORES (PROYECTOS DE OBRA NUEVA)

Las empresas de desarrollo inmobiliario buscan constantemente diferenciar sus proyectos con "smart features" que atraigan a compradores y arrendatarios. Para un promotor, integrar de fábrica el sistema HYBLICON-II significa ofrecer un ascensor que ajusta sus itinerarios en función del flujo real de pasajeros: menos tiempo de espera, mayor eficiencia energética y un argumento de venta poderoso. Además, al proporcionar datos de ocupación anonimizados, pueden incorporar esta telemetría a sus sistemas de gestión de edificios para optimizar climatización y limpieza de rellanos según uso.

11.1.2 FACILITY MANAGERS Y EMPRESAS DE MANTENIMIENTO

Las compañías encargadas del mantenimiento de flotas de ascensores necesitan reducir averías y anticipar problemas. Con HYBLICON-II, el registro continuo de patrones de uso (número de usuarios, presencia de sillas de ruedas y muletas) permite generar alarmas tempranas (picos de uso inusuales, posibles atascos mecánicos). El modelo de suscripción anual facilita que, más allá de la instalación, la solución siga mejorándose vía OTA, convirtiéndose en un servicio diferencial frente a la competencia.

11.1.3 SECTOR SANITARIO Y RESIDENCIAL DE MAYORES

En hospitales y residencias, garantizar el tránsito seguro de personas con movilidad reducida es crítico. HYBLICON-II no solo identifica usuarios en silla de ruedas y andador, sino que asigna prioridad de llamada, evitando que tengan que esperar junto al dintel. Este acceso preferente reduce el riesgo de caídas y mejora la experiencia del paciente o residente. Para la gerencia, además, se genera un histórico de uso que puede emplearse en auditorías de accesibilidad y cumplimiento normativo.

11.1.4 EMPRESAS INTEGRADORAS DE SISTEMAS Y CONSULTORAS DE SMART BUILDING

Las consultoras que diseñan soluciones IoT para edificios buscan plataformas modulares para ofrecer a sus clientes. HYBLICON-II puede integrarse vía API con plataformas de control de acceso, automatización de edificios y sistemas, convirtiéndose en un componente plug-and-play dentro de proyectos globales de digitalización. Para estas empresas, la versatilidad de la solución y su escalabilidad en múltiples unidades es clave.

11.1.5 ADMINISTRACIONES PÚBLICAS Y TRANSPORTE URBANO

La licitación de servicios para estaciones de metro, tren o edificios municipales suele requerir demostrar compromisos con la accesibilidad universal. HYBLICON-II, capaz de identificar y priorizar automáticamente a usuarios con movilidad reducida, aporta un componente de "smart city" y cumple con los requisitos de las directivas europeas. A la vez, los datos agregados alimentan políticas de mejora de infraestructuras y planificación de flujos de pasajeros.

11.2 PLAN DE COMERCIALIZACIÓN

Para asegurar una adopción rápida y sostenible de la solución HYBLICON-II en el mercado, MP Ascensores puede desplegar una estrategia de comercialización en varias capas, combinando validación técnica con acciones de marketing y ventas dirigidas.

11.2.1 PILOTOS Y CASOS DE ÉXITO

Teniendo como objetivo generar prueba social y datos cuantitativos que demuestren el valor añadido de HYBLICON-II.

- **Proyectos selectos**: Iniciar pilotos en 3-5 instalaciones clave, como por ejemplo un hospital público, una residencia de mayores y un edificio de oficinas de alto tráfico. Cada piloto debe incluir un plan de medición de KPIs: reducción de tiempo de espera, porcentaje de detecciones de movilidad reducida, y tasas de satisfacción de usuarios y mantenimiento.
- **Documentación de resultados**: Elaborar un informe técnico y un vídeo testimonial con métricas (por ejemplo, "20 % menos de tiempo de espera en hospital", "100 % de prioridad atendida para sillas de ruedas") que sirvan como referencia para futuros clientes.

11.2.2 ALIANZAS ESTRATÉGICAS

Siendo el principal objetivo ampliar el canal de venta y facilitar la integración en grandes cuentas.

- **Fabricantes OEM**: Firmar acuerdos de co-desarrollo con los equipos de innovación de Otis, KONE o Schindler para ofrecer HYBLICON-II como opción de serie en sus nuevos ascensores. Ofrecer licencia reducida a cambio de volumen integrado.
- **Empresas de facility management**: Diseñar un programa de "Partner HYBLICON-II" para que compañías como Ferrovial Servicios o Clece ofrezcan la solución dentro de sus contratos de mantenimiento, con formación técnica y soporte dedicado para ingenieros de campo.
- **Integradores IoT**: Colaborar con proveedores de BMS (Building Management Systems) como Schneider Electric o Siemens, para que ofrezcan la telemetría de ocupación de ascensores como parte de sus dashboards de eficiencia energética y predictive maintenance.

11.2.3 PARTICIPACIÓN EN FERIAS Y CONGRESOS

Siendo la máxima prioridad dar visibilidad y captar leads cualificados.

- **Eventos sectoriales**: Reservar stand y charla en Interlift (Augsburgo), Smart Building Expo (Madrid) y Construmat (Barcelona). Preparar demo en tiempo real mostrando detección de género, edad y movilidad reducida.
- **Workshops técnicos**: Impartir sesiones de 30 min en congresos de facility management o smart cities, explicando la arquitectura edge, OTA y casos de uso en accesibilidad.
- **Networking VIP**: Organizar un evento privado con directores de obra, responsables de mantenimiento y gestores de inmuebles, donde se muestre el ROI a través de un café-break "elevador inteligente".

12. CONCLUSIONES

Al cerrar este viaje junto a MP Ascensores en el marco de HYBLICON-II, cabe destacar la potencia transformadora que puede tener la visión por ordenador aplicada al ámbito de la movilidad. Desde los primeros bocetos de arquitectura hasta los ensayos en entornos reales, hemos construido un sistema capaz no solo de contar personas en un rellano, sino de

comprender sus necesidades: género, rango de edad, dispositivos de movilidad y, sobre todo, su intención de entrar. Esa capacidad de "leer" el lenguaje corporal y anticiparse al comportamiento humano representa un salto cualitativo en la experiencia de uso de un ascensor, reduciendo tiempos de espera, dando prioridad a quienes más lo necesitan y aportando datos útiles para la gestión de edificios inteligentes.

La integración de YOLOv11-pose, los clasificadores demográficos y el detector de movilidad reducida ha sido un reto apasionante. Cada módulo requirió un proceso iterativo de recolección de datos, anotación y ajuste de hiperparámetros que reforzó nuestra convicción sobre la importancia de contar con conjuntos de datos de calidad y un pipeline flexible. Los ensayos in situ, con sus sorpresas (imágenes borrosas por vibraciones, oclusiones inesperadas, pruebas con mascarillas), nos enseñaron que un modelo no vive sólo de líneas de código, sino de su capacidad de adaptarse a condiciones reales.

En el plano profesional, este proyecto ha ampliado mi horizonte: he profundizado en arquitecturas de detección avanzada, pipelining asíncrono y despliegue edge, al tiempo que he aprendido a gestionar versiones, a documentar con rigor y a presentar resultados tangibles a un cliente industrial. La colaboración con los equipos de MP Ascensores me ha brindado una experiencia de trabajo en empresa real, comprendiendo plazos, presupuesto y la necesidad de ofrecer una solución robusta y mantenible. Tanto es así que **estamos trabajando actualmente junto con el propio equipo de MP en llevar el contenido a una publicación en una revista de alto impacto**, ya que creemos firmemente en el potencial del proyecto.

A nivel personal, HYBLICON-II ha sido un reto de perseverancia y pasión. Cada línea de código, cada prueba de validación y cada ajuste de parámetros han reforzado mi sentido de vocación hacia la inteligencia artificial aplicada a problemas reales. He sentido la satisfacción de ver el sistema reconocer correctamente a un usuario en silla de ruedas o anticipar el flujo de personas con precisión, sabiendo que este desarrollo contribuirá a hacer los edificios más accesibles y eficientes.

13. TRABAJO FUTURO

13.1 EXTENSIÓN DE FUNCIONALIDADES

En futuras iteraciones, se propone ampliar el abanico de dispositivos de asistencia que HYBLICON-II es capaz de reconocer, incorporando scooters eléctricos, bastones inteligentes y otros implementos de movilidad. Al mismo tiempo, se desarrollarán algoritmos de detección de agrupaciones, familias, colas espontáneas o grupos de usuarios, que permitan ajustar dinámicamente las rutas de las cabinas, habilitando modos "express" o "skip-stop" cuando convenga.

Asimismo, se explorará la integración de modelos de análisis de expresión corporal para estimar niveles de urgencia o estrés, de modo que el sistema pueda priorizar en tiempo real a quien más lo necesite.

13.2 MEJORA Y ADAPTACIÓN DE MODELOS

Para optimizar la capacidad de adaptación del sistema a nuevos entornos, se investigarán técnicas de *few-shot learning* y *transfer learning* que permitan reentrenar o ajustar los modelos con muy pocas imágenes nuevas, reduciendo drásticamente el tiempo y esfuerzo de anotación. Paralelamente, se evaluarán arquitecturas aún más ligeras como Tiny-YOLO o

MobileNetV3 junto con métodos de *pruning* y cuantización avanzada, con el objetivo de minimizar la latencia y el consumo energético en hardware edge de bajo coste.

Por último, se enriquecerán los datasets con datos sintéticos que simulen condiciones adversas niebla, lluvia, reflejos para robustecer la detección en cualquier circunstancia ambiental.

13.3 INTEGRACIÓN CON SISTEMAS EXTERNOS

Una línea prioritaria será conectar HYBLICON-II con plataformas de Building Management Systems (BMS) para intercambiar telemetría de ocupación y optimizar servicios como climatización, iluminación o limpieza de rellanos según el flujo real de usuarios. Además, se diseñarán integraciones con sistemas de control de acceso y ticketing, permitiendo ascensores "solo para autorizados" o accesos prioritarios a personal de servicio.

13.4 ESCALABILIDAD Y DESPLIEGUE MASIVO

A nivel de infraestructura, se planifica una arquitectura distribuida que utilice orquestadores ligeros (Kubernetes o Docker Swarm) para gestionar clusters de nodos edge en campus y complejos de edificios.

Finalmente, se desplegarán dashboards de monitorización en tiempo real que muestren métricas clave, latencia, tasa de acierto, calidad de imagen, para realizar ajustes proactivos desde un centro de operaciones.

14. BIBLIOGRAFÍA Y ANEXOS

[1] Kollmitz, J., Eitel, A., Vasquez, A., & Burgard, W. (2018). **Deep 3D perception of people and their mobility aids.** *Robotics and Autonomous Systems, 114*, 29–40. https://doi.org/10.1016/j.robot.2019.01.011

[2] Alruwaili, M., Siddiqi, M. H., Atta, M. N., & Arif, M. (2024). **Deep learning and ubiquitous systems for disabled people detection using YOLO models.** *Computers in Human Behavior, 154*, 108150. https://doi.org/10.1016/j.chb.2024.108150

[3] Lecrosnier, L., Khemmar, R., Ragot, N., Decoux, B., Rossi, R., Kefi, N., & Ertaud, J. (2021). **Deep learning-based object detection, localization and tracking for smart wheelchair healthcare mobility.** *International Journal of Environmental Research and Public Health, 18*(1), 91. https://doi.org/10.3390/ijerph18010091

[4] Vasquez, A., Kollmitz, M., Eitel, A., & Burgard, W. (2017). **Deep detection of people and their mobility aids for a hospital robot.** https://arxiv.org/abs/1708.00674

[5] Zhang, J., Tsiligkaridis, A., Taguchi, H., Raghunathan, A., & Nikovski, D. (2022). **Transformer networks for predictive group elevator control.** *arXiv preprint* arXiv:2208.08948. https://doi.org/10.48550/arXiv.2208.08948

[6] Yuren Yang, Ye Yuan, Ting Pan, Xingyu Zang, Gang Liu (2021). **A framework for occupancy prediction based on image information fusion and machine learning,IEIS.**https://www.sciencedirect.com/science/article/abs/pii/S0360132321009197#preview-section-abstract

[7] Gharbi, A., Ayari, M., El Touati, Y., et al. (2024). **Intelligent elevator control using decentralized multi-agent systems. Journal of Electrical Systems, 20(9s), 3038-3046**https://www.researchgate.net/publication/386042688_Intelligent_Elevator_Control_Using_Decentralized_Multi-Agent_Systems

[8] Lee, J., Kim, K., & Park, J. (2023). **Computer vision-based intelligent elevator information system for real-time occupancy monitoring.** *Science of the Total Environment, 862*, 160868. https://doi.org/10.1016/j.scitotenv.2022.160868

[9] Kim, S.-K. (2024). **A study on the non-contact artificial intelligence elevator system due to the effect of COVID-19.** *Electronics, 13*(16), 3193. https://doi.org/10.3390/electronics13163193

[10] Sahar Darafsh, Saeed Shiry Ghidary, Morteza Saheb Zamani (2021). **Real-Time Activity Recognition and Intention Recognition Using a Vision-based Embedded System** https://arxiv.org/abs/2107.12744

[11] Yozgatli, P., Acar, Y., Tulumen, M., et al. (2024). **Fall detection in passenger elevators using intelligent surveillance camera systems: An application with YOLOv8 Nano model** https://arxiv.org/abs/2501.01985

[12] Ge, Y., Lu, J., Fan, W., & Yang, D. (2013). **Age estimation from human body images.** En *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP) (pp. 2337–2341). https://doi.org/10.1109/ICASSP.2013.6638072

[13] Ince, Ö., Erdogmus, D., & Principe, J. C. (2015). **Human age estimation from body shape and gait analysis.** *Pattern Recognition Letters, 66*, 43–49. https://doi.org/10.1016/j.patrec.2015.04.018

[14] LearnOpenCV. (2025). *YOLO11 on Raspberry Pi: Optimizing Object Detection for Edge*. https://learnopencv.com/yolo11-on-raspberry-pi/

[15] Rice, T. (2024). *Real Time Inference on Raspberry Pi 4 (30 fps!)*. PyTorch Tutorials. https://docs.pytorch.org/tutorials/intermediate/realtime_rpi.html

[17] Ultralytics. (2023). *Quick Start Guide: Raspberry Pi with Ultralytics YOLO11*. Ultralytics Docs. https://docs.ultralytics.com/guides/raspberry-pi/

[18] SemiEngineering. (2025). *Deploying PyTorch Models On Edge Devices*. Semiconductor Engineering. Recuperado de https://semiengineering.com/deploying-pytorch-models-on-edge-devices/

[19] Datature. (2022). *How to Load Vision Models on Raspberry Pi for Edge Deployment*. Datature Blog. https://www.datature.io/blog/how-to-load-vision-models-on-raspberry-pi-for-edge-deployment

[20] Librería de uso y desarrollo de YOLO, Ultralytics. https://docs.ultralytics.com/es/