

Sistema inteligente de detección y asistencia para personas con discapacidad en cruces peatonales utilizando visión por computadora

Martin Chavez-Jaramillo, Cesar Benavides-Alvarez, Israel Santoyo-Luévano

Universidad Autónoma Metropolitana, Unidad Azcapotzalco,
Departamento de Electrónica, Ciudad de México,
México

{a12163036728, cesarbenavides, iss1}@azc.uam.mx

Resumen. Los sistemas de semaforización convencionales carecen de mecanismos inclusivos para personas con movilidad reducida durante los intervalos de cruce peatonal. Esta ausencia representa un riesgo significativo en entornos de tráfico vehicular denso, especialmente para usuarios de silla de ruedas que requieren tiempos de cruce más prolongados. Estudios previos en movilidad urbana indican que la falta de adaptación en los cruces semaforizados incrementa la siniestralidad de este colectivo en intersecciones de alto flujo vehicular. En el presente trabajo se propone un sistema inteligente de detección en tiempo real mediante visión por computadora e inteligencia artificial. Para ello, se implementó el modelo YOLO11 optimizado al formato TensorFlow Lite sobre una placa Raspberry Pi 5, integrando un módulo de control de señalizaciones visuales compuesto por LEDs y una pantalla OLED, el cual prioriza automáticamente el tiempo de cruce al detectar al usuario. El modelo fue entrenado con un dataset de 1,100 imágenes etiquetadas en condiciones diurnas y nocturnas, empleando aumentación de datos para simular oclusión parcial y diferentes perspectivas. Los resultados obtenidos muestran una precisión de 82.9 % en la métrica mAP50-95 bajo condiciones controladas, con detección efectiva a distancias de hasta 4.5 metros y un tiempo de respuesta inferior a 2 segundos. Estos hallazgos demuestran la viabilidad y escalabilidad técnica de implementar soluciones de asistencia tecnológica de bajo costo y alta eficiencia para la movilidad inclusiva urbana. Como trabajo futuro, se plantea integrar sensores de profundidad y comunicación inalámbrica con semáforos existentes, así como extender la validación a usuarios con bastones o andadores.

Palabras clave: Visión por computadora, detección de objetos en tiempo real, movilidad inclusiva, sistemas embebidos, semaforización inteligente.

Intelligent Detection and Assistance System for People with Disabilities at Pedestrian Crossings Using Computer Vision

Abstract. Conventional traffic light systems lack inclusive mechanisms for people with reduced mobility during pedestrian crossing intervals. This absence represents a significant risk in dense vehicular traffic environments, especially for wheelchair users who require longer crossing times. Previous studies on urban mobility indicate that the lack of adaptation at signalized crosswalks increases the accident rate among this group at high-traffic intersections. This paper proposes an intelligent real-time detection system based on computer vision and artificial intelligence. To this end, the YOLO11 model, optimized to the TensorFlow Lite format, was implemented on a Raspberry Pi 5 board, integrating a control module for visual signals composed of LEDs and an OLED screen, which automatically prioritizes crossing time upon detecting the user. The model was trained on a dataset of 1,100 labeled images under daytime and nighttime conditions, using data augmentation to simulate partial occlusion and different viewing angles. The results obtained show a precision of 82.9% in the mAP50-95 metric under controlled conditions, with effective detection at distances of up to 4.5 meters and a response time of less than 2 seconds. These findings demonstrate the technical feasibility and scalability of implementing low-cost, high-efficiency technological assistance solutions for inclusive urban mobility. As future work, the integration of depth sensors and wireless communication with existing traffic lights is proposed, as well as extending validation to users with canes or walkers.

Keywords: Computer vision, real-time object detection, inclusive mobility, embedded systems, intelligent traffic light system.

1. Introducción

En las ciudades modernas, el derecho a la movilidad representa un pilar fundamental para la inclusión ciudadana. Sin embargo, la infraestructura vial actual no cuenta con un diseño adecuado para garantizar seguridad y autonomía a las personas en silla de ruedas, particularmente en los cruces peatonales. Esta situación constituye un problema social de relevancia, dado que la ausencia de señalización adaptada genera riesgos elevados en entornos de tráfico vehicular denso.

El sistema de semaforización tradicional en México únicamente contempla alarmas sonoras orientadas a personas invidentes, omitiendo las necesidades de usuarios en silla de ruedas, quienes requieren tiempos de cruce más prolongados [6]. Esta carencia vulnera los principios universales de accesibilidad vial y limita la integración autónoma de las personas con discapacidad en el entorno urbano.

Ante este escenario, la inteligencia artificial (IA) y, en particular, la visión por computadora, han demostrado ser herramientas eficaces para abordar problemáticas de accesibilidad y seguridad vial. Diversas investigaciones han explorado el uso de modelos de detección de objetos basados en redes neuronales convolucionales para asistir a poblaciones vulnerables en entornos urbanos [6][7][10]. No obstante, la mayoría de estas soluciones se desarrollan sobre infraestructura costosa o de difícil implementación en países en desarrollo.

En este contexto, el presente trabajo propone un sistema inteligente basado en visión por computadora e inteligencia artificial con el objetivo de desarrollar un prototipo a escala capaz de detectar usuarios en silla de ruedas en tiempo real, mediante una arquitectura embebida de bajo costo sustentada en una placa Raspberry Pi 5. El sistema integra el modelo YOLO11, versión más reciente de la familia *You Only Look Once*, optimizado para dispositivos de cómputo reducido, contribuyendo así a la autonomía, seguridad vial y accesibilidad urbana de personas con movilidad reducida [5][12].

2. Estado del arte

La visión por computadora es un subcampo de la inteligencia artificial que permite a los sistemas computacionales interpretar y analizar información visual del mundo real a partir de imágenes o secuencias de video. Su potencial radica en el uso del aprendizaje profundo (*Deep Learning*), subcapa del aprendizaje automático (*Machine Learning*), que entrena modelos para simular la percepción visual humana en entornos complejos. El componente fundamental de este enfoque son las redes neuronales convolucionales (RNC), estructuras compuestas por capas de filtros capaces de extraer características visuales progresivamente, desde formas geométricas simples hasta escenas de alta complejidad. Sobre esta base técnica se construyen los modelos de detección de objetos en tiempo real, siendo la familia YOLO (*You Only Look Once*) uno de los enfoques más adoptados en la literatura reciente [1][9].

Diversas investigaciones han explorado el uso de modelos YOLO para asistir a personas con discapacidad o movilidad reducida en entornos urbanos. Louis et al. [7] desarrollaron un sistema avanzado de asistencia al conductor de silla de ruedas eléctrica, basado en detección, estimación de profundidad y seguimiento de objetos mediante YOLOv3 y una cámara Intel RealSense, con el objetivo de construir mapas semánticos tridimensionales del entorno inmediato. En una línea similar, Ainandafiq et al. [2] propusieron un sistema de navegación autónoma para sillas de ruedas en interiores empleando YOLOv8 con módulos de atención geográfica, donde el mecanismo de Atención de Coordenadas (AC) alcanzó una precisión del 99.5 %, superando al Módulo de Atención de Bloques Convolucionales (MABC) con un 96.15 %. Por su parte, Shisei et al. [10] abordaron la monitorización de usuarios de silla de ruedas en habitaciones privadas mediante imágenes térmicas de baja resolución y YOLOv8 en su variante *nano*, priorizando la eficiencia en latencia y consumo de hardware.

Otros trabajos han centrado su atención en la detección de peatones en situaciones de riesgo vial. Lee et al. [6] implementaron un algoritmo YOLOv5 específicamente orientado a la identificación de peatones que utilizan dispositivos de asistencia, como sillas de ruedas y muletas, en pasos de cebra, dada su mayor vulnerabilidad ante accidentes de tráfico. Hsu y Lin [4] abordaron la problemática de las oclusiones en la detección peatonal, donde factores como la vestimenta, obstrucciones parciales o condiciones de iluminación reducida disminuyen la precisión del modelo; para ello, combinaron YOLOv4 con técnicas de segmentación y fusión adaptativa multirresolución, logrando mayor robustez en escenarios adversos. Complementando este enfoque, Tomás et al. [11] propusieron la evaluación de accesibilidad de rutas peatonales para usuarios de silla de ruedas mediante imágenes aéreas y geográficas procesadas con YOLOv8, aportando una perspectiva de planificación urbana inclusiva.

La rápida evolución de la familia YOLO ha sido documentada sistemáticamente en la literatura. Marco et al. [9] realizaron una revisión exhaustiva del algoritmo aplicado al reconocimiento de señales de tráfico, evaluando cinco versiones (YOLOv1 a YOLOv5) en términos de aplicaciones, conjuntos de datos, métricas, hardware y desafíos. En la misma dirección, Ahmed et al. [1] presentaron un *benchmark* comparativo entre YOLOv4 y YOLOv8, demostrando que es posible simplificar la arquitectura del modelo sin comprometer el rendimiento en detección en tiempo real. Khanam y Hussain [5] documentaron los avances arquitectónicos de YOLO11, la versión más reciente de la familia, destacando la introducción del bloque C3K2, los mecanismos SPPF y C2PSA, así como una reducción del 22 % en parámetros respecto a YOLOv8, manteniendo una mayor precisión media promedio (mAP) [12].

A pesar de los avances descritos, la mayoría de las soluciones revisadas se enfocan en asistencia en interiores, navegación autónoma o análisis de rutas, sin abordar de forma integral la problemática de la semaforización inclusiva en cruces peatonales urbanos. Asimismo, varias de estas implementaciones dependen de hardware especializado de alto costo, lo que limita su escalabilidad en contextos con recursos reducidos. El presente trabajo propone cubrir esta brecha mediante la implementación de YOLO11 sobre una plataforma embebida de bajo costo (Raspberry Pi 5), orientada específicamente a la detección de usuarios en silla de ruedas en tiempo real y a la activación automatizada de tiempos extendidos de cruce semafórico [5][12].

3. Metodología

El sistema inteligente propuesto se estructuró en tres módulos independientes y complementarios: el módulo de ejecución, encargado del entrenamiento e inferencia del modelo; el módulo de control, responsable de la gestión del hardware embebido; y el módulo de interfaz, que integra la visualización en tiempo real del sistema. Esta separación modular facilita el mantenimiento, la escalabilidad y la reproducibilidad del prototipo. La arquitectura general del sistema se ilustra en la Figura 1.

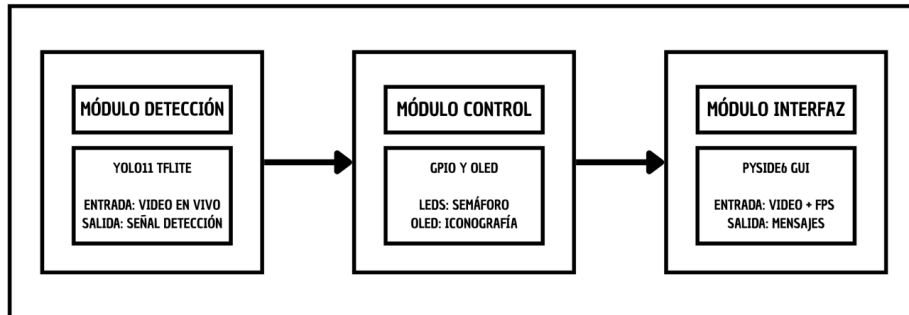


Fig. 1. Diagrama a bloques de los módulos del sistema.

3.1. Módulo de detección

Este módulo constituye el núcleo computacional del sistema, abarcando el entrenamiento del modelo de detección y su posterior optimización para su despliegue en hardware embebido. El entrenamiento se realizó en la plataforma Google Colab, aprovechando el acceso a unidades de procesamiento gráfico (GPU) de manera gratuita a través de los servidores de Google. El hardware GPU que se utilizó fue un Tesla T4 dentro del entorno de ejecución Python 3 con una VRAM de 15GB.

Conjunto de datos Se construyó un conjunto de datos compuesto por 1,100 imágenes de personas en silla de ruedas, recolectadas desde Open Images Dataset v7 de Google [3]. Las imágenes se dividieron en dos subconjuntos: 1,000 para entrenamiento y 100 para validación. Se optó por el 10% de imágenes de validación para maximizar los datos de entrenamiento dada la especificidad de la clase única, asegurando que el modelo viera la mayor cantidad de variaciones de sillas de ruedas posibles en el entorno de aprendizaje. Todas procesadas a una resolución de 320×320 píxeles para equilibrar precisión y velocidad de inferencia en la GPU virtual disponible.

Cada imagen fue etiquetada mediante la herramienta MakeSense AI [8], generando por cada muestra un archivo `.txt` compatible con el formato YOLO. Dicho archivo contiene cinco valores: el identificador de clase (*wheelchair*, asignado con valor 0), las coordenadas normalizadas del centro del *bounding box* en los ejes *X* e *Y*, y las dimensiones normalizadas del ancho y alto del *bounding box*, todas en un rango de 0 a 1. La representación visual de este formato se aprecia en las Figuras 2 y 3, donde se aprecia las dimensiones en coordenadas *X* y *Y*, alto y ancho del etiquetado de la imagen y además la clase a la que representa. Para este ejemplo se tiene a clase persona y clase corbata. En las coordenadas de formato YOLO se observa el valor que se le asignó a la respectiva clase, es decir, para persona se le asignó el valor 0 y para corbata el valor 27.

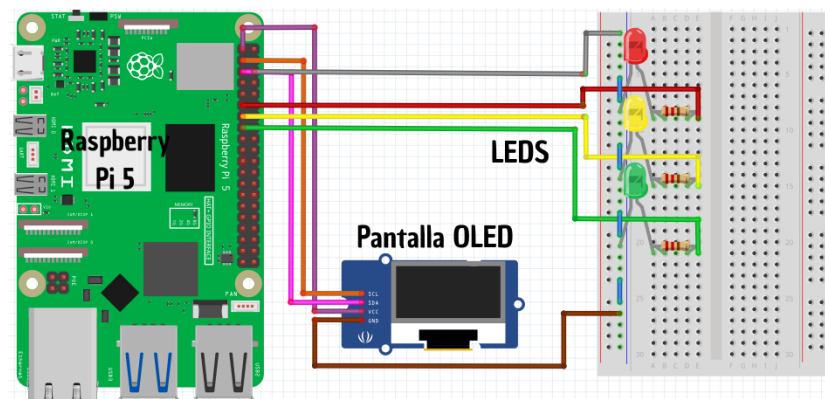


Fig. 4. Diagrama simulado de la configuración de LEDs y pantalla OLED.

- mosaic=0: aumentación desactivada para preservar la integridad espacial de cada imagen, evitando distorsiones artificiales que podrían degradar la precisión en la clase única entrenada.

Exportación del modelo Al concluir el entrenamiento, se obtuvo el modelo resultante en formato PyTorch (`best.pt`). Para su despliegue en la placa Raspberry Pi 5, el modelo fue exportado al formato TensorFlow Lite mediante el siguiente comando:

```
!yolo export model=best.pt format=tflite
```

Se seleccionó la variante `float32` (`best_float32.tflite`) por ser la que menor pérdida de precisión sufre durante el proceso de cuantificación, reduciendo el tamaño del modelo y optimizando la latencia sin impactar negativamente el rendimiento del sistema embebido.

3.2. Módulo de control

Este módulo comprende la arquitectura hardware del sistema embebido, diseñada para simular el comportamiento de un semáforo urbano inteligente. Se empleó una placa Raspberry Pi 5 conectada a tres LEDs (rojo, amarillo y verde) mediante una protoboard y cables jumpers, así como una pantalla OLED que muestra la señalización representativa del usuario en silla de ruedas y el tiempo exclusivo de espera asignado. El diagrama de la configuración física se ilustra en la Figura 4.

La lógica de control se implementó en Python, actuando como puente entre la salida de inferencia del modelo YOLO11 y los componentes físicos, a través de un entorno virtual dentro del sistema operativo de la placa Raspberry Pi 5. Al momento en que el modelo detecta un usuario en silla de ruedas con una confianza superior al umbral configurado, el sistema activa automáticamente la señalización de cruce extendido.

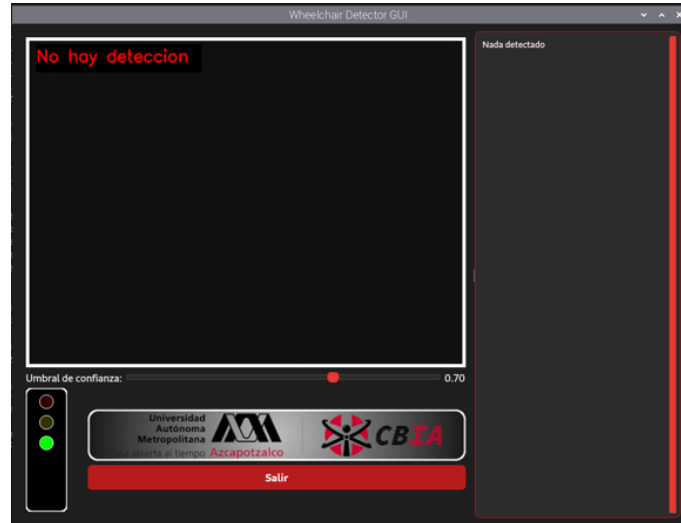


Fig. 5. Interfaz gráfica del sistema.

3.3. Módulo de interfaz

El módulo de interfaz integra todos los componentes del sistema en una aplicación de escritorio desarrollada con la librería PySide6, un framework multiplataforma de software libre para la construcción de interfaces gráficas (GUI) modernas e interactivas. La interfaz cumple las siguientes funciones:

- Visualización del video en tiempo real con indicador de fotogramas por segundo (FPS).
- Registro histórico de detecciones, con marcas de tiempo de inicio y fin de cada evento.
- Semáforo virtual sincronizado con el estado físico de los LEDs, para verificación inmediata del comportamiento del sistema.
- Barra de calibración del umbral de confianza, ajustable entre el 70% y el 90%, para adaptar el sistema a distintas condiciones ambientales como variaciones de iluminación, clima adverso o posibles oclusiones parciales.

La interfaz gráfica del sistema se aprecia en la Figura 5.

4. Resultados

En esta sección se presentan los resultados obtenidos en dos etapas: la evaluación cuantitativa del modelo durante el entrenamiento en Google Colab y la validación experimental del sistema en condiciones reales controladas.

Métrica / Condición	Validación en Colab (YOLO11n)
Precisión (P)	0.985
Recall (R)	1.000
mAP50	0.995
mAP50-95	0.829

Table 1. Resultados del entrenamiento YOLO11 en Google Colab

Cuadrante	Valor	Interpretación
Verdaderos positivos	452	Silla de ruedas detectada correctamente
Falsos positivos	32	Fondo detectado como silla de ruedas
Falsos negativos	1	Silla de ruedas presente no detectada
Verdaderos negativos	0	No aplicable (clase única)

Table 2. Matriz de confusión del modelo YOLO11 entrenado

4.1. Evaluación del entrenamiento

Al concluir el proceso de entrenamiento, el modelo generó automáticamente un conjunto de gráficas de desempeño que permiten analizar la evolución de las métricas a lo largo de las épocas. En la Figura 8 se presentan las curvas de precisión obtenidas como resultado final del entrenamiento, recall, mAP50 y mAP50-95 obtenidas.

Los valores finales de las métricas de evaluación se resumen en la Tabla 1.

La métrica de mayor relevancia para evaluar el desempeño global del modelo es mAP50-95, dado que considera múltiples umbrales de intersección sobre la unión (IoU), desde el 50 % hasta el 95 %, ofreciendo una medida integral de la precisión en distintos niveles de dificultad. El modelo alcanzó un valor de 82.9 % en esta métrica, lo que indica un desempeño robusto para la detección de una clase única bajo condiciones controladas [12].

Para evaluar cuantitativamente la capacidad de discriminación del modelo, se analizó la matriz de confusión generada sobre 485 instancias de predicción, cuyos resultados se ilustran en la Figura 9 y se detallan en la Tabla 2.

Al analizar los 32 falsos positivos registrados en la matriz de confusión, se observa que el modelo mantiene una alta fidelidad visual respecto a las etiquetas originales, véase la Figura 10. Los errores de precisión se atribuyen a una ligera sobre-sensibilidad del modelo en entornos con alta densidad de texturas o estructuras metálicas que comparten patrones geométricos con la estructura de las sillas de ruedas. Sin embargo, como se aprecia en la Figura 11, el modelo logra una superposición casi exacta de los bounding boxes respecto a la verdad de campo, lo que garantiza la fiabilidad del sistema en la activación del semáforo.

4.2. Validación en entorno real

Con el propósito de evaluar el desempeño del sistema fuera del entorno de laboratorio, se llevó a cabo una prueba experimental en las instalaciones de la

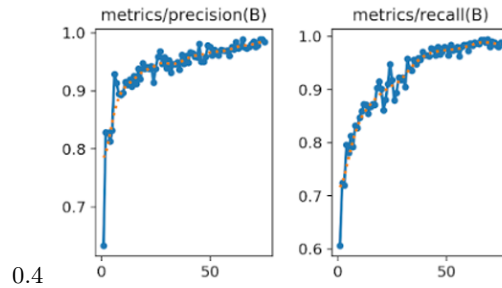


Fig. 6. Métricas de precisión y recall.

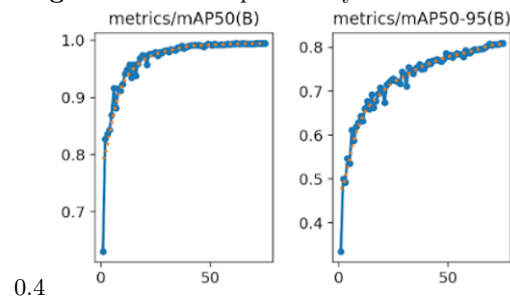


Fig. 7. mAP50 y maAP50-95.

Fig. 8. Graficas generadas al finalizar el entrenamiento del modelo YOLO.

Universidad Autónoma Metropolitana, Unidad Azcapotzalco. Se utilizó una silla de ruedas real y uno de los autores actuó como usuario simulado de movilidad reducida. la cámara se situó a una altura de 1.5 metros respecto al suelo, con un ángulo de inclinación de 10 grados, simulando la posición típica de una unidad de control en un semáforo peatonal.

Las pruebas se realizaron bajo dos configuraciones de umbral de confianza: 80 % y 90 %, con el objetivo de evaluar el alcance de detección efectiva del modelo en condiciones reales. Los resultados obtenidos se describen a continuación:

- **Umbral del 80 %:** la detección se extendió hasta una distancia de 4.5 metros, véase la Figura 12, lo que representa una distancia operativamente más relevante para entornos de semaforización real, aunque con una ligera reducción en la especificidad de las detecciones.
- **Umbral del 90 %:** el modelo logró detectar correctamente al usuario a una distancia aproximada de 2.5 metros, véase la Figura 13. Si bien esta distancia resulta limitada para una aplicación directa en cruces semafóricos urbanos, confirma que el sistema opera con alta precisión en condiciones favorables de iluminación y visibilidad.

Las distancias de 4.5 y 2.5 metros fueron seleccionadas como casos de estudio, porque para el modelo estas distancias representaban su límite de detección,

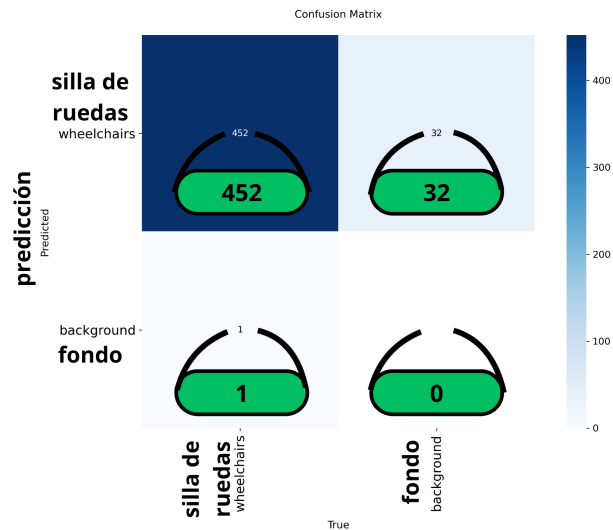


Fig. 9. Matriz de confusión del modelo YOLO11n entrenado.

pues al superarlas con sus respectivos umbrales, comenzaban los falsos positivos o falsos negativos.

Adicionalmente, el sistema demostró un tiempo de respuesta inferior a 2 segundos desde la detección del usuario hasta la activación de la señalización de cruce extendido, realizando el proceso de inmediato, superando el objetivo inicial establecido para el prototipo.

5. Conclusiones y trabajo futuro

El presente trabajo demostró la viabilidad técnica de implementar un sistema inteligente de detección en tiempo real para usuarios en silla de ruedas mediante visión por computadora y hardware embebido de bajo costo. La integración del modelo YOLO11n, optimizado al formato TensorFlow Lite sobre una placa Raspberry Pi 5, permitió alcanzar una precisión de 82.9 % en la métrica mAP50-95 bajo condiciones controladas, con detección efectiva a distancias de hasta 4.5 metros y un tiempo de respuesta inferior a 2 segundos desde la detección del usuario hasta la activación de la señalización de cruce extendido.

Si bien la detección a un umbral de confianza del 90 % se limitó a una distancia de 2.5 metros, los resultados obtenidos con un umbral del 80 % evidencian que el sistema opera de forma estable en condiciones reales, considerando las restricciones inherentes al hardware embebido, las variaciones de iluminación y la calidad de la imagen capturada. La combinación hardware-software propuesta constituye una base sólida y replicable para el desarrollo de soluciones de accesibilidad tecnológica en infraestructura vial urbana.



Fig. 10. Imágenes de validación originales.



Fig. 11. Predicción de las imágenes de validación durante el entrenamiento.

Como trabajo futuro se identifican las siguientes líneas de investigación y desarrollo:

- **Ampliación y diversificación del conjunto de datos:** incorporar imágenes capturadas en condiciones de baja iluminación, clima adverso y entornos urbanos de alta densidad, con el fin de mejorar la robustez del modelo ante escenarios no controlados.
- **Evaluación de variantes superiores de YOLO11:** explorar las versiones *small* y *medium* del modelo sobre hardware con mayor capacidad de cómputo, como la Raspberry Pi 5 con acelerador Hailo-8, para incrementar el alcance de detección sin comprometer la latencia del sistema.
- **Integración con infraestructura semafórica real:** validar el prototipo en cruces peatonales reales en coordinación con autoridades de movilidad



Fig. 12. Detección de silla de ruedas con umbral del 80 % a 4.5 metros de distancia.

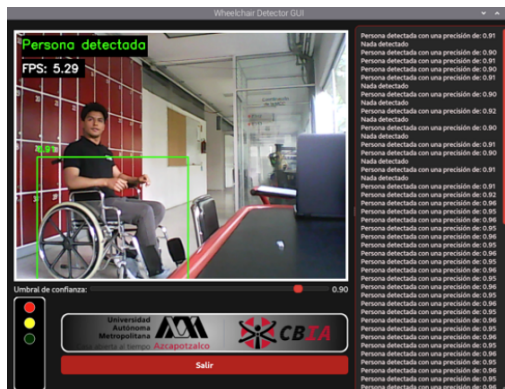


Fig. 13. Detección de silla de ruedas con umbral del 90 % a 2.5 metros de distancia.

urbana, evaluando su desempeño en condiciones de tráfico vehicular activo.

- **Detección multiclase:** extender el sistema para identificar otros perfiles de movilidad reducida, como personas con muletas, andadores o adultos mayores, ampliando el impacto social de la solución.

References

1. Ahmed, M., Israa, J., Tawfeq, P., Ahmen: Real time pedestrian and objects detection using enhanced yolo integrated with learning complexity-aware cascades. TELKOMNIKA Telecommunication Computing Electronics and Control pp. 364–369 (2024)
2. Ainandafiq, F., Muhammad, R., Corina, F.: Attention module in yolo-based object detection method for autonomous smart wheelchair room navigation system. International Conference on Robotics, Automation, and Artificial Intelligence pp. 313–316 (2024)

3. Google: Open images dataset v7 (2026), <https://storage.googleapis.com/openimages/web/index.html>, accedido: 25-03-2026
4. Hsu, W.Y., Lin, W.Y.: Adaptive fusion of multi-scale yolo for pedestrian detection. *Open Access Journal* pp. 3–10 (2021)
5. Khanam, R., Hussain, M.: An overview of the key architectural enhancements. *Arxiv* pp. 3–4 (2024)
6. Lee, K., Na, H., Kwak: A study on traffic vulnerable detection using object detection-based ensemble and yolov5. *Journal of The Korea Society of Computer and Information* pp. 62–67 (2024)
7. Louis, R., Nicolas, B., Romain, N., Ertau: Deep learning-based object detection, localisation and tracking for smart wheelchair healthcare mobility. *International Journal of Environmental Research and Public Health* pp. 5–13 (2021)
8. MakeSense: Makesense ai (2026), <https://www.makesense.ai/>, accedido: 25-03-2026
9. Marco, C., Diego, J., Bryan, B.D., Juan, D., María, J.: Traffic sign detection and recognition using yolo object detection algorithm: A systematic review. *Mathematics* pp. 5–24 (2024)
10. Shisei, M., Miwa, Y., Yuichi, H.: Detecting falls and slips of wheelchair users using low-resolution thermal image analysis. *IEEE Systems, Man and Cybernetics Society Section* pp. 136883–136891 (2025)
11. Tomás, Agustín, S., Pierpaolo: Urban pedestrian routes' accessibility assessment using geographic information system processing and deep learning-based object. *Sensors* pp. 6–16 (2024)
12. Ultralytics: Yolo11 overview (2024), <https://docs.ultralytics.com/es/models/yolo11/#overview>, accedido: 25-03-2026