

RICUY: Sistema Inteligente de Asistencia al Conductor en vías urbanas de Lima y Andahuaylas

José Luis Barboza Gonzales, Paul Vladimir Ruiz Crisostomo, Diego Omar Tenorio Huarancca,
Mario Yufra Chambilla y Oscar Salas Vizcarra

Resumen—Los usuarios vulnerables de la vía en entornos urbanos del Perú se enfrentan a peligros que los conjuntos de datos estándar de visión por computadora representan mal: la mototaxi—vehículo motorizado de tres ruedas que constituye un modo de transporte masivo y, sin embargo, está ausente de los grandes repositorios internacionales—, la señalización lumínica local y marcas viales deterioradas. Este artículo presenta RICUY—del quechua *rikuy*, “ver; observar”—un Sistema Avanzado de Asistencia al Conductor (ADAS) de tiempo real que integra segmentación de instancias (YOLOv8L-seg), seguimiento multiobjeto (ByteTrack), estimación monocular de distancia mediante el modelo de cámara estenopeica (pinhole) y alertas de voz en español. El núcleo experimental del trabajo es un estudio de transferencia y localización: se parte de un punto de control YOLOv8L-seg preentrenado sobre tráfico andino de Ayacucho [30] y se adapta al tráfico costero de Lima mediante ajuste fino sobre un conjunto de datos local original de 253 imágenes anotadas con LabelMe bajo un esquema de 12 clases. Evaluados sobre una partición de validación idéntica de Lima (49 imágenes, 464 instancias), el ajuste fino con datos locales eleva el mAP50 de caja de 0.355 a 0.613 (+72.7 %) y el mAP50 de máscara de 0.318 a 0.556 (+74.8 %), con mejoras simultáneas de precisión y exhaustividad; el caso más ilustrativo es el semáforo en rojo, cuyo mAP50 de caja asciende de 0.013 a 0.711. El sistema completo se validó de forma operativa sobre secuencias de conducción naturalista de Lima y Andahuaylas, sosteniendo tasas de 14–43 fotogramas por segundo en una GPU de consumo NVIDIA RTX 3050 de 4 GB, tras un ajuste fino de apenas 41 min. Los resultados constituyen una verificación experimental directa de la brecha de localización y demuestran la viabilidad de un ADAS de segmentación de instancias para el tráfico urbano peruano sobre hardware de recursos limitados.

Palabras clave—sistema avanzado de asistencia al conductor, segmentación de instancias, aprendizaje por transferencia, ajuste fino, estimación monocular de distancia, seguimiento multiobjeto, YOLOv8, mototaxi, seguridad vial, usuarios vulnerables de la vía.

I. INTRODUCCIÓN

Los peatones constituyen el grupo más agudamente vulnerable en el tránsito vial: la Organización Mundial de la Salud informa que representan el 23 % de las muertes viales a nivel mundial y que, en conjunto, los usuarios vulnerables de la vía—peatones, ciclistas y conductores de vehículos motorizados de dos y tres ruedas—concentran más de la mitad de las muertes [1]. El Informe sobre la Situación Mundial de la Seguridad Vial 2023 registró aproximadamente 1.19 millones de muertes anuales atribuibles a siniestros de tránsito, equivalentes a una carga económica estimada del 3 % del PIB mundial [1].

Esta carga no se distribuye uniformemente: las naciones con presupuestos limitados para infraestructura vial absorben el 90 % de las muertes pese a concentrar solo el 60 % de los vehículos registrados, y una proporción desproporcionada de las muertes involucra infraestructura informal o no estandarizada que los conjuntos de datos existentes de visión por computadora son estructuralmente incapaces de representar [1], [2].

En el contexto peruano, el Observatorio Nacional de Seguridad Vial (ONSV) reportó 87,083 siniestros de tránsito durante 2023, con un saldo de 3316 fallecidos [5]. Los factores humanos explican el 70.2 % de los incidentes, con la imprudencia del conductor (28.2 %) y el exceso de velocidad (26.6 %) como causas predominantes [5], [6]. Lima, conurbación metropolitana costera, concentra el parque vehicular más grande y heterogéneo del país; las ciudades andinas intermedias como Andahuaylas (2926 m s.n.m.) presentan una alta penetración del transporte informal dominado por la mototaxi—un vehículo motorizado de tres ruedas para el transporte de pasajeros que constituye el principal modo de tránsito informal en estas ciudades—junto con motocicletas convencionales, autobuses y vehículos particulares [7].

El sistema presentado en este artículo se denomina RICUY, palabra proveniente del quechua—la principal lengua indígena de la región andina, hablada por un estimado de 8–12 millones de personas en seis países sudamericanos [3]. En el dialecto quechua sureño, *rikuy* se define como un verbo que significa “ver; observar” [4]; la forma causativa derivada *rikuchiy* (“hacer ver; mostrar”) expresa el acto de dirigir la mirada de otro hacia un peligro, dimensión semántica que captura la función anticipatoria de alerta del sistema propuesto.

Los Sistemas Avanzados de Asistencia al Conductor (ADAS) basados en detección de objetos con aprendizaje profundo han demostrado efectividad tangible en la mitigación de peligros viales [8], [9]. La segmentación de instancias extiende la detección por cajas delimitadoras al recuperar el contorno preciso a nivel de píxel de cada objeto, lo que habilita análisis espaciales inalcanzables con aproximaciones rectangulares: caracterización de los límites del objeto, evaluación de riesgo de proximidad basada en zonas y cálculo de solapamiento entre actores del tránsito e infraestructura [10], [23]. La estimación monocular de distancia mediante el modelo de cámara estenopeica ofrece una alternativa práctica e independiente de sensores para el despliegue de ADAS con recursos limitados en países en desarrollo [11], [12], [14]; su integración con el seguimiento multiobjeto ByteTrack [15] habilita además la estimación temporal de velocidad para el cálculo del Tiempo hasta la Colisión (TTC) frente a los umbrales de la NHTSA [13].

El problema de la localización. La brecha entre los conjuntos de datos internacionales y los entornos reales de despliegue ha sido cuantificada: Linardi et al. demostraron que un modelo YOLO entrenado sobre un conjunto público de señales de tránsito indonesias alcanza un mAP@50 del 97.07 % en su partición de prueba pero cae al 28.68 % al evaluarse sobre imágenes reales de vías indonesias recolectadas de forma independiente [16]. Para las ciudades peruanas esta brecha se agrava por la mototaxi como clase vehicular independiente [17], [18] y por marcas y señales viales locales ausentes de repositorios como BDD100K [25]. Un trabajo previo de nuestro grupo desarrolló QHAWAY, un ADAS de segmentación de instancias entrenado para el tráfico andino de Ayacucho [30]; el presente trabajo toma ese punto de control como modelo base y aborda de manera directa la pregunta de cuánto se degrada al transferirse a un entorno vial costero distinto (Lima) y cuánto se recupera mediante ajuste fino con datos locales.

Contribuciones. Este artículo aporta cuatro contribuciones principales. Primero, se construye un conjunto de datos local original de 253 imágenes de tráfico urbano de Lima anotadas con segmentación de instancias bajo un esquema de 12 clases alineado con el modelo base andino, incluyendo la mototaxi como clase de exclusividad local. Segundo, se realiza un estudio de transferencia y localización controlado (Ayacucho→Lima) que aísla y cuantifica, sobre una partición de validación idéntica, el efecto del ajuste fino con datos del entorno de despliegue: el mAP50 de caja asciende de 0.355 a 0.613 (+72.7 %). Tercero, se presenta RICUY, una aplicación ADAS de escritorio en tiempo real que integra segmentación de instancias, seguimiento ByteTrack, estimación monocular de distancia y alertas de voz, demostrada sobre secuencias naturalistas de Lima y Andahuaylas a 14–43 FPS en una GPU de 4 GB. Cuarto, se aporta un análisis por clase que identifica qué actores viales recupera el ajuste fino local y cuáles permanecen limitados por escasez de datos, orientando la priorización de la recolección futura.

II. TRABAJOS RELACIONADOS

II-A. Segmentación de Instancias en Aplicaciones ADAS

La detección de objetos de una sola etapa basada en la familia YOLO se ha consolidado como el estándar de facto para la percepción ADAS en tiempo real por su equilibrio entre exactitud y latencia [9], [12]. La segmentación de instancias añade una rama de máscara a nivel de píxel que resulta especialmente valiosa para la evaluación de riesgo de proximidad peatonal y la caracterización geométrica de actores e infraestructura [23], [10]. RICUY adopta YOLOv8L-seg [27] por su disponibilidad de pesos de segmentación y su despliegue eficiente sobre GPU de consumo.

II-B. Estimación Monocular de Distancia y Seguimiento

La estimación de distancia mediante el modelo estenopeico requiere únicamente una distancia focal calibrada y la altura conocida del objeto, ofreciendo una alternativa de bajo costo a los sensores LiDAR o estéreo [14], [11]. El seguimiento multiobjeto mediante ByteTrack [15] asocia detecciones entre

fotogramas y habilita la estimación temporal de velocidad necesaria para el cálculo del TTC frente a los umbrales de alerta de colisión frontal de la NHTSA [13].

II-C. La Brecha de Localización y las Clases Locales

Diversos estudios documentan la severa degradación de modelos entrenados sobre repositorios no representativos del entorno de despliegue [16]. En el contexto peruano, la mototaxi permanece ausente de los repositorios internacionales y rara vez se reporta como clase de segmentación independiente [17], [18]; de modo análogo, las marcas viales y los cruces peatonales locales se desvían de los perfiles estandarizados [19], [20], [21]. Trabajos previos de nuestro grupo abordaron el reconocimiento de señales de tránsito con alertas de voz para cascos históricos andinos [29] y la segmentación de instancias para el tráfico de Ayacucho [30]. El presente artículo se distingue al cuantificar experimentalmente la transferencia de ese modelo base andino al tráfico metropolitano costero de Lima.

III. MATERIALES Y MÉTODOS

III-A. Arquitectura del Sistema RICUY

RICUY se implementa como una aplicación de escritorio que procesa secuencias de video de cámara de tablero en tiempo real a través de una cascada de cuatro etapas (Figura 1). (i) **Percepción:** cada fotograma se redimensiona y se procesa con el modelo de segmentación de instancias YOLOv8L-seg, que produce cajas, máscaras a nivel de píxel y puntuaciones de confianza para las 12 clases del esquema (Sección III-C). (ii) **Seguimiento:** las detecciones se asocian entre fotogramas mediante ByteTrack [15], que asigna identificadores persistentes de trayectoria. (iii) **Estimación de distancia y riesgo:** para cada objeto seguido se estima la distancia longitudinal mediante el modelo de cámara estenopeica,

$$D = \frac{H_{\text{real}} \times f}{h_{\text{px}}}, \quad (1)$$

donde H_{real} es la altura física nominal de la clase, f la distancia focal calibrada en píxeles y h_{px} la altura de la caja en píxeles [14]. La variación temporal de D a lo largo de la trayectoria proporciona una estimación de la velocidad de aproximación y del TTC, con los que el sistema clasifica cada objeto en zonas de riesgo alineadas con la NHTSA (peligro: <3 m; precaución: 3–7 m) [13]. (iv) **Alerta:** un gestor de prioridad emite alertas de voz en español mediante síntesis texto-a-voz [24], restringidas a las clases de mayor criticidad (peatón, reductor de velocidad y semáforo en rojo) para evitar la saturación del conductor. La interfaz gráfica superpone máscaras de segmentación, identificadores de seguimiento, etiquetas de distancia y banners de alerta, y permite exportar registros de detección en formato CSV.

III-B. Implementación de la Aplicación RICUY

La cascada anterior se materializa en una aplicación de escritorio propia, desarrollada con una interfaz gráfica PyQt, que ejecuta la inferencia sobre secuencias de video y expone los controles operativos al usuario (Figuras 8 y 9). El panel de

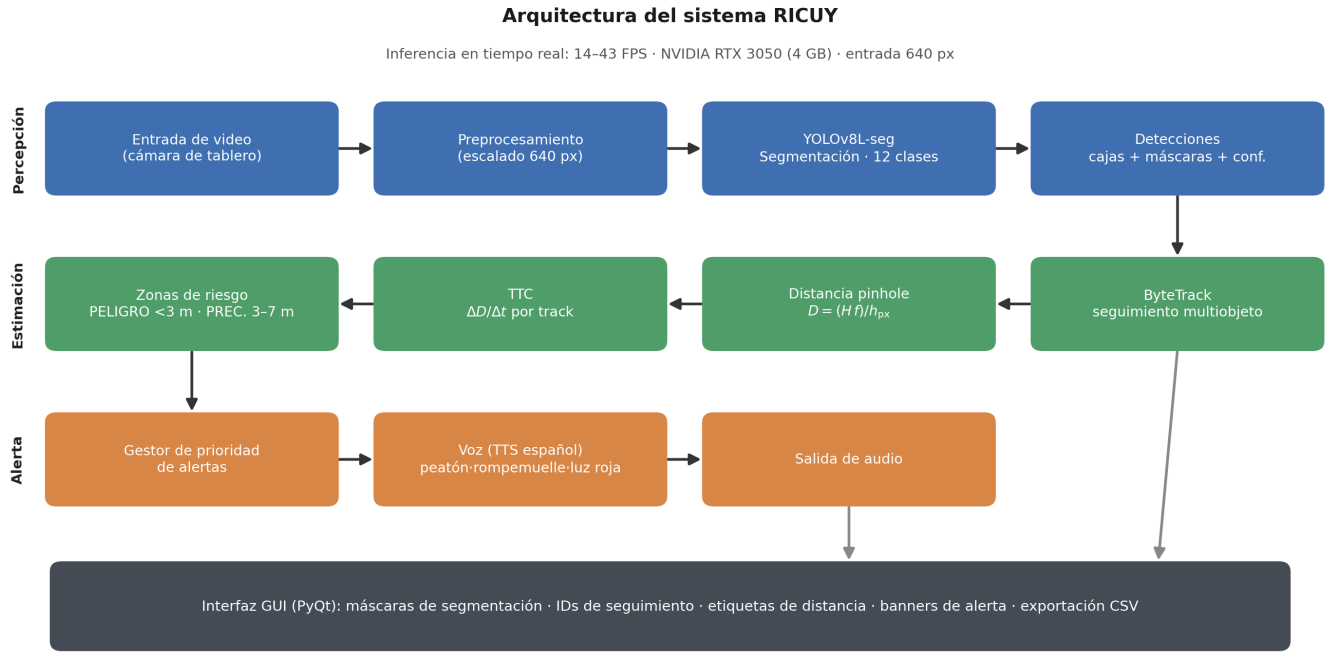


Figura 1. Arquitectura del sistema RICUY: cascada de percepción (segmentación de instancias YOLOv8L-seg), estimación (seguimiento ByteTrack, distancia monocular pinhole, TTC y clasificación en zonas de riesgo según la NHTSA) y alerta (gestor de prioridad y síntesis de voz en español), integrada en una interfaz gráfica PyQt. La inferencia opera en tiempo real (14–43 FPS) sobre una GPU de consumo NVIDIA RTX 3050 de 4 GB.

configuración permite ajustar el umbral de confianza y activar o desactivar de forma independiente el seguimiento ByteTrack, la superposición de máscaras de segmentación, las alertas por voz y la grabación del video procesado. Un panel de estadísticas en vivo reporta la tasa de fotogramas por segundo, el número de objetos detectados, el número de trayectorias activas y el desglose de detecciones por clase, mientras que un registro de alertas recientes muestra el nivel de peligro y el mensaje emitido. Cada objeto se anota en pantalla con su máscara de segmentación, su identificador de seguimiento, la clase y la confianza, y la distancia monocular estimada en metros; las situaciones de riesgo generan banners de alerta y la locución de voz correspondiente. La aplicación admite además la captura de fotogramas y la exportación de los registros de detección y seguimiento en formato CSV para su análisis posterior, y se ejecuta en tiempo real sobre una GPU de consumo NVIDIA RTX 3050 de 4 GB.

III-C. Modelo Base y Esquema de 12 Clases

El punto de partida es un modelo YOLOv8L-seg (45.9 M de parámetros, 210 GFLOPs) previamente entrenado por nuestro grupo sobre tráfico andino de Ayacucho [30], sin exposición alguna a la dinámica vial costera de Lima. Para preservar la cabeza de detección ya entrenada y realizar un verdadero ajuste fino, el conjunto de datos local de Lima se alineó al esquema exacto de 12 clases del modelo base (Tabla I). Esta alineación resultó indispensable: las anotaciones LabelMe originales empleaban sinónimos e identificadores de clase distintos que, de no haberse realineado, habrían invalidado la transferencia de la cabeza de detección.

Tabla I
ESQUEMA OFICIAL DE 12 CLASES DEL MODELO BASE, CON EL SINÓNIMO EMPLEADO EN LAS ANOTACIONES LABELME LOCALES.

ID	Clase (modelo base)	Sinónimo LabelMe
0	Reductor de velocidad	Tope de velocidad
1	Paso peatonal	Cruce peatonal
2	Línea recta	Línea recta
3	Línea recta y derecha	Línea recta a la derecha
4	Persona	Persona
5	Motocicleta	Motocicleta
6	Automóvil particular	Automóvil particular
7	Camión	Camión
8	Bus de transporte	Autobús
9	Semáforo en verde	Semáforo en verde
10	Semáforo en rojo	Semáforo en rojo
11	Mototaxi	Mototaxi

III-D. Conjunto de Datos Local de Lima

Se recolectaron y anotaron manualmente 253 pares válidos de imagen y archivo JSON de LabelMe [26] a partir de secuencias de conducción naturalista en vías urbanas de Lima. La anotación es predominantemente poligonal (2199 polígonos y 31 rectángulos), lo que permite máscaras de segmentación de instancias de alta fidelidad. Tras la alineación al esquema oficial (Sección III-C) no se registraron etiquetas desconocidas ni archivos corruptos. El conjunto se dividió de forma determinista (semilla 42) en 80 % / 20 %, resultando en 204 imágenes de entrenamiento y 49 de validación. La Tabla II resume la composición por clase y la Figura 2 ilustra la distribución de instancias de la partición de entrenamiento. La distribución es

Tabla II
COMPOSICIÓN DEL CONJUNTO DE DATOS LOCAL DE LIMA POR CLASE
(INSTANCIAS, ENTRENAMIENTO + VALIDACIÓN).

Clase	Instancias
Automóvil particular	1184
Persona	454
Bus de transporte	236
Mototaxi	90
Semáforo en rojo	69
Camión	58
Motocicleta	51
Semáforo en verde	48
Paso peatonal	33
Línea recta	4
Reductor de velocidad	3
Línea recta y derecha	0
Total	2230

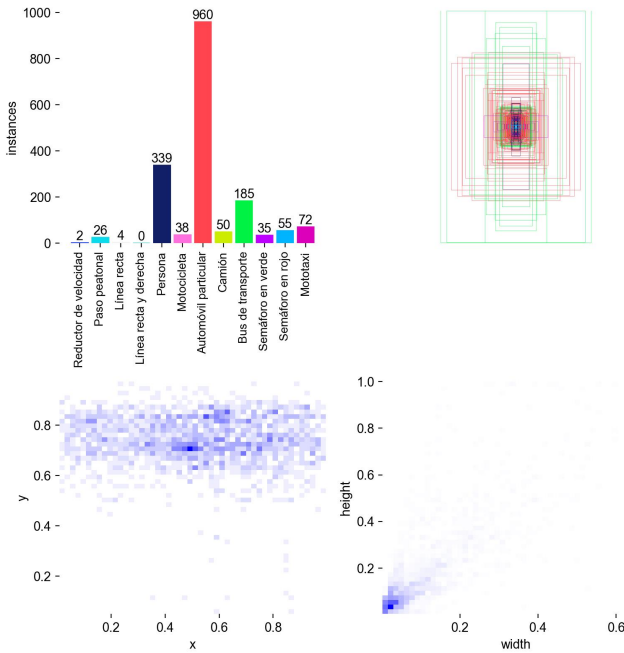


Figura 2. Distribución de instancias por clase de la partición de entrenamiento de Lima (204 imágenes) y mapas de densidad espacial y de dimensiones de las cajas. El predominio del automóvil particular y la persona, y el soporte casi nulo de las clases de marcas viales, caracterizan el desbalance del conjunto.

marcadamente desbalanceada—un reflejo fiel de la frecuencia real de los actores en el tráfico de Lima—y dominada por el automóvil particular y la persona; tres clases de marcas viales (reductor de velocidad, línea recta y línea recta y derecha) presentan un soporte casi nulo, condición cuyo impacto se analiza en la Sección V.

III-E. Estrategia de Ajuste Fino

El ajuste fino se realizó con Ultralytics sobre el punto de control base, bajo un presupuesto de cómputo deliberadamente restringido: una GPU portátil NVIDIA GeForce RTX 3050 con 4 GB de VRAM. Los hiperparámetros se seleccionaron para adaptar el modelo al dominio de Lima sin sobrescribir el

Tabla III
CONFIGURACIÓN DE AJUSTE FINO (ULTRALYTICS, YOLOv8L-SEG).

Parámetro	Valor
Modelo base	YOLOv8L-seg (Ayacucho) [30]
GPU	NVIDIA RTX 3050 (4 GB)
Resolución de entrada	640 px
Tamaño de lote	4
Optimizador	<i>auto</i> (Ultralytics)
lr_0 / lr_f	0.005 / 0.01
Momento / <i>weight decay</i>	0.937 / 0.0005
Épocas de calentamiento	3
Épocas (objetivo / efectivas)	150 / 111 (parada temprana)
Paciencia	40
Mosaico / <i>close_mosaic</i>	1.0 / 10
HSV (h , s , v)	0.015, 0.7, 0.4
Traslación / escala	0.1 / 0.5
Volteo horizontal / vertical	0.5 / 0.0
Rotación / <i>mixup</i>	0.0 / 0.0
Precisión mixta (AMP)	Sí

conocimiento andino previamente adquirido: resolución de entrada de 640 px, tamaño de lote de 4 (consumo ≈ 3.5 –4 GB, sin desbordamiento de memoria), tasa de aprendizaje inicial suave $lr_0 = 0.005$, aumento por mosaico activo con desactivación en las últimas 10 épocas (*close_mosaic*), volteo horizontal con probabilidad 0.5 y, dado el carácter del tráfico, sin volteo vertical ni rotaciones. El objetivo se fijó en 150 épocas con parada temprana (paciencia 40). El entrenamiento se reanudó una vez tras una interrupción y se detuvo por parada temprana en la época 111; el mejor modelo, seleccionado por la métrica de aptitud de Ultralytics, correspondió a la época 71. La evaluación se realizó sobre la partición de validación de Lima (49 imágenes, 464 instancias) a 640 px. El entrenamiento completo (111 épocas) tomó 41 min sobre la RTX 3050, y el mejor modelo (época 71) se alcanzó en apenas 8 min, lo que evidencia la eficiencia del ajuste fino frente a un entrenamiento desde cero. La Tabla III detalla la configuración completa en aras de la reproducibilidad.

IV. RESULTADOS

IV-A. Progreso del Entrenamiento

La Figura 3 muestra la evolución de las pérdidas y las métricas a lo largo del ajuste fino. Las pérdidas de entrenamiento de caja, máscara, clasificación y *DFL* descienden de forma monótona y estable, mientras que las métricas de validación—mAP50 y mAP50-95 de caja y de máscara—ascienden con rapidez durante las primeras 30 épocas y se estabilizan hacia la época 70, momento a partir del cual la parada temprana evita el sobreajuste.

IV-B. Estudio de Transferencia: Ajuste Fino Ayacucho→Lima

El resultado central de este trabajo es la comparación, sobre una partición de validación idéntica de Lima, entre el punto de control no localizado (línea base de Ayacucho, sin exposición alguna a datos de Lima) y el mismo punto de control tras el ajuste fino con 253 imágenes locales. La Tabla IV resume los resultados. El ajuste fino con datos locales produjo mejoras

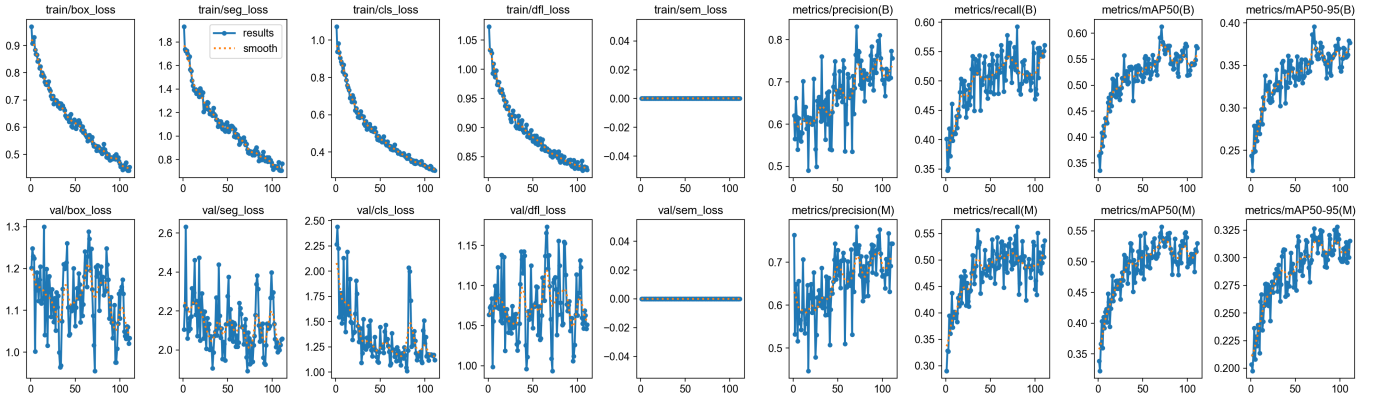


Figura 3. Curvas de entrenamiento y validación del ajuste fino sobre datos de Lima (RTX 3050, 640 px). Fila superior: pérdidas de entrenamiento y métricas de caja (B). Fila inferior: pérdidas de validación y métricas de máscara (M). El mejor modelo corresponde a la época 71.

sustanciales en todas las métricas evaluadas: el mAP50 de caja ascendió de 0.355 a 0.613 (+25.8 pp, +72.7 % relativo) y el mAP50 de máscara de 0.318 a 0.556 (+23.8 pp, +74.8 %), con incrementos simultáneos de precisión y exhaustividad que descartan un compromiso entre ambas. El efecto resulta especialmente pronunciado en clases sensibles al contexto local: la detección de semáforo en rojo se elevó de un mAP50 de caja de 0.013 a 0.711, lo que evidencia que el punto de control no localizado era prácticamente incapaz de reconocer la señalización lumínica bajo las condiciones de captura de Lima. Estos resultados constituyen una validación experimental directa de la brecha de localización cuantificada por Linardi et al. [16].

IV-C. Rendimiento por Clase: Línea Base vs. Ajustado

La Tabla V desglosa el efecto del ajuste fino por clase, comparando el punto de control no localizado de Ayacucho [30] con el modelo ajustado sobre la misma validación de Lima. El patrón es revelador: las ganancias más espectaculares se concentran en las clases cuya apariencia depende fuertemente del contexto local de captura. El semáforo en rojo asciende de 0.013 a 0.711 de mAP50 de caja (+0.698), el paso peatonal de 0.145 a 0.667 (+0.522), el camión de 0.099 a 0.607 (+0.508) y el semáforo en verde de 0.115 a 0.606 (+0.491); el modelo base era prácticamente ciego a estas clases bajo las condiciones de Lima. En contraste, las clases vehiculares de apariencia más universal ya partían de un desempeño aceptable y mejoran de forma más moderada (automóvil particular 0.699→0.731, persona 0.611→0.651, mototaxi 0.630→0.692). Promediado únicamente sobre las nueve clases con soporte de entrenamiento suficiente, el mAP50 de caja del modelo ajustado asciende a 0.68; las tres clases de marcas viales con soporte casi nulo (reductor de velocidad, línea recta y línea recta y derecha; ≤ 4 instancias) deprimen el promedio global a 0.613 y no aprenden señal útil. La Figura 4 ilustra cualitativamente esta diferencia sobre un mismo lote de validación.

IV-D. Validación Cruzada

Para verificar que el resultado de transferencia no depende de una partición particular, se evaluó el modelo sobre 3 particiones

independientes de validación (cada una retiene ≈ 51 imágenes y emplea ≈ 202 para el ajuste fino), re-ajustando el modelo base de Ayacucho en cada una con configuración idéntica y presupuesto de 70 épocas. La Tabla VI reporta la media $\pm$ desviación estándar. La métrica principal del modelo de partición única cae dentro de la banda de ± 1 DE de la validación cruzada (mAP50 de caja 0.613 frente a 0.585 ± 0.111 ; de máscara 0.556 frente a 0.529 ± 0.106) y el mAP50-95 coincide casi exactamente (0.396 frente a 0.396 ± 0.089), lo que confirma que el desempeño reportado es representativo y no un artefacto de partición. De manera consistente, las tres particiones superan ampliamente la línea base de Ayacucho (mAP50 de caja 0.355). La precisión de la partición única (0.831) resultó superior a la media de la validación cruzada (0.699 ± 0.058), lo que indica que dicha partición fue algo favorable en precisión; la exhaustividad (*recall*), en cambio, fue representativa (0.560 frente a 0.587 ± 0.100).

IV-E. Curvas Precisión–Recall y Matriz de Confusión

Las curvas precisión–recall de caja y de máscara (Figura 5) confirman el comportamiento agregado: las clases con soporte suficiente sostienen una precisión elevada en un amplio rango de exhaustividad, mientras que las clases de marcas viales deprimen el promedio global. La matriz de confusión normalizada (Figura 6) muestra una diagonal dominante para los actores vehiculares y peatonales, con las confusiones residuales concentradas en el fondo para las clases de bajo soporte.

IV-F. Validación Operativa y Rendimiento en Tiempo Real

El sistema RICUY completo se ejecutó sobre secuencias de conducción naturalista, sin escenarios escenificados, en una GPU NVIDIA RTX 3050. La Figura 7 presenta predicciones del modelo sobre lotes de validación, y las Figuras 8 y 9 muestran la aplicación en operación sobre secuencias de Lima y Andahuaylas, respectivamente, con máscaras de segmentación, identificadores de seguimiento ByteTrack, etiquetas de distancia monocular y alertas activas. En condiciones controladas, la inferencia del modelo de segmentación sobre la validación de Lima registró ≈ 28 ms por imagen (0.6 ms de preprocesamiento,

Tabla IV

ESTUDIO DE TRANSFERENCIA AYACUCHO→LIMA: EFECTO DEL AJUSTE FINO CON DATOS LOCALES SOBRE EL CONJUNTO DE VALIDACIÓN DE LIMA (49 IMÁGENES, 464 INSTANCIAS, ESQUEMA DE 12 CLASES). LÍNEA BASE: PUNTO DE CONTROL YOLOV8L-SEG ENTRENADO EN AYACUCHO [30] SIN EXPOSICIÓN A DATOS DE LIMA. AJUSTADO: EL MISMO PUNTO DE CONTROL TRAS EL AJUSTE FINO SOBRE 253 IMÁGENES LOCALES. AMBOS MODELOS SE EVALUARON A 640 PX SOBRE LA MISMA PARTICIÓN DE VALIDACIÓN.

Métrica	Línea base (Ayacucho)	Ajustado (Lima)	Δ abs. (pp)	Δ rel. (%)
Box mAP50	0.355	0.613	+25.8	+72.7
Box mAP50-95	0.232	0.396	+16.4	+70.7
Box Precisión	0.501	0.831	+33.0	—
Box Recall	0.371	0.560	+18.9	—
Mask mAP50	0.318	0.556	+23.8	+74.8
Mask mAP50-95	0.198	0.326	+12.8	+64.6
Mask Precisión	0.563	0.780	+21.7	—
Mask Recall	0.300	0.531	+23.1	—

Tabla V

RENDIMIENTO POR CLASE: LÍNEA BASE DE AYACUCHO [30] FRENTE AL MODELO AJUSTADO A LIMA, SOBRE LA MISMA VALIDACIÓN DE LIMA (49 IMÁGENES); VALORES DE mAP50. ORDENADO POR LA MEJORA DE CAJA. “INST.”: INSTANCIAS DE VALIDACIÓN; “—”: SIN INSTANCIAS DE VALIDACIÓN; Δ EN PUNTOS ABSOLUTOS.

Clase	Inst.	Box mAP50			Mask mAP50		
		Base	Ajust.	Δ	Base	Ajust.	Δ
Semáforo en rojo	14	0.013	0.711	+0.698	0.002	0.489	+0.487
Paso peatonal	7	0.145	0.667	+0.522	0.145	0.667	+0.522
Camión	8	0.099	0.607	+0.508	0.099	0.401	+0.302
Semáforo en verde	13	0.115	0.606	+0.491	0.115	0.570	+0.455
Bus de transporte	51	0.678	0.806	+0.128	0.614	0.847	+0.233
Motocicleta	13	0.556	0.678	+0.122	0.416	0.678	+0.262
Mototaxi	18	0.630	0.692	+0.062	0.583	0.692	+0.109
Persona	115	0.611	0.651	+0.040	0.521	0.533	+0.012
Automóvil particular	224	0.699	0.731	+0.032	0.681	0.717	+0.036
Reductor de velocidad	1	0.000	0.000	0.000	0.000	0.000	0.000
Línea recta	0	—	—	—	—	—	—
Línea recta y derecha	0	—	—	—	—	—	—
Todas (global)	464	0.355	0.613	+0.258	0.318	0.556	+0.238



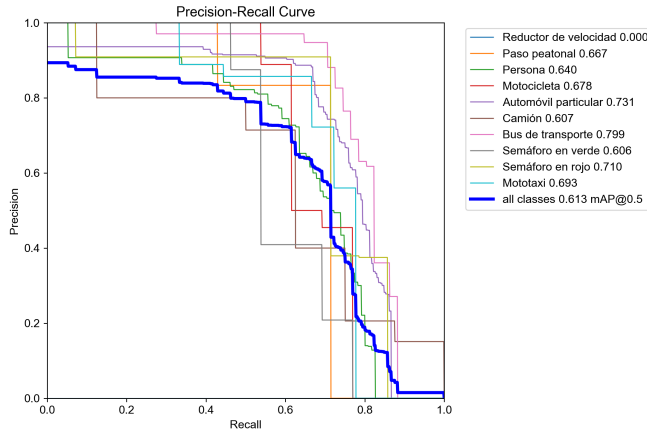
(a) Línea base de Ayacucho (sin datos de Lima).

(b) Modelo ajustado a Lima.

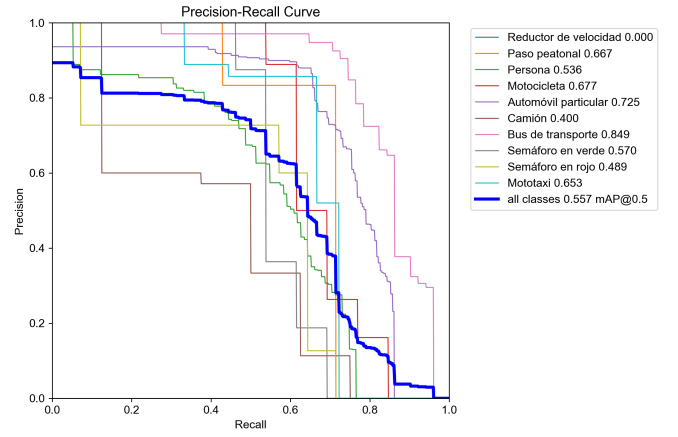
Figura 4. Comparación cualitativa sobre el mismo lote de validación de Lima: el modelo base de Ayacucho (a) omite o confunde múltiples actores y la señalización lumínica, mientras que el modelo ajustado (b) recupera detecciones y máscaras correctas para automóviles, personas, buses, mototaxis y semáforos.

28.0 ms de inferencia y 1.3 ms de posprocesamiento), equivalentes a unos 35 FPS de percepción pura en la RTX 3050. La aplicación completa—que añade el seguimiento, la estimación de distancia, la gestión de alertas y el renderizado de la interfaz—sostuvo un rango de 14–43 FPS extremo a extremo según la densidad de la escena, manteniéndose dentro de una ventana compatible con la percepción y reacción del

conductor para aplicaciones de asistencia [13]. Entre los eventos espontáneos registrados destacan la detección de peatones que cruzan de forma imprevista, la alerta de semáforo en rojo en escenas con múltiples peatones próximos y la detección simultánea de mototaxis, automóviles y personas en intersecciones de alta densidad.



(a) Curva PR de caja.



(b) Curva PR de máscara.

Figura 5. Curvas precisión–recall del modelo ajustado sobre la validación de Lima, para detección por caja (a) y segmentación por máscara (b).

Tabla VI

ROBUSTEZ SOBRE 3 PARTICIONES INDEPENDIENTES DE VALIDACIÓN (CADA UNA ≈ 51 IMÁGENES RETENIDAS), RE-AJUSTANDO EL MODELO BASE EN CADA UNA CON IDÉNTICA CONFIGURACIÓN. MEDIA $\pm$ DESVIACIÓN ESTÁNDAR; ÚLTIMA COLUMNA: MODELO DE PARTICIÓN ÚNICA (TABLA IV).

Métrica	CV (media $\pm$ DE)	Part. única
Box mAP50	0.585 $\pm$ 0.111	0.613
Box mAP50-95	0.396 $\pm$ 0.089	0.396
Box Precisión	0.699 $\pm$ 0.058	0.831
Box Recall	0.587 $\pm$ 0.100	0.560
Mask mAP50	0.529 $\pm$ 0.106	0.556
Mask mAP50-95	0.324 $\pm$ 0.054	0.326
Mask Precisión	0.673 $\pm$ 0.075	0.780
Mask Recall	0.544 $\pm$ 0.093	0.531

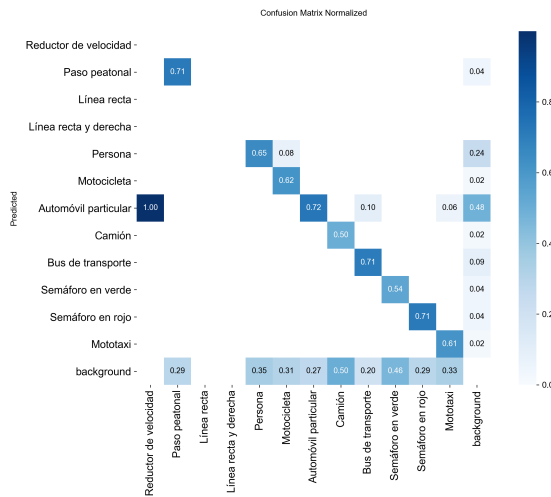


Figura 6. Matriz de confusión normalizada del modelo ajustado sobre la validación de Lima.

V. DISCUSIÓN

V-A. La localización como factor decisivo

El estudio de transferencia (Sección IV-B) ofrece evidencia directa y controlada de que la brecha de localización es el factor dominante del rendimiento en este dominio. Un

modelo competente en el tráfico andino de Ayacucho se degrada hasta un mAP50 de caja de 0.355 al enfrentarse a la dinámica vial costera de Lima, y recupera +72.7 % relativo con apenas 253 imágenes locales. El caso del semáforo en rojo (0.013 $\rightarrow$ 0.711) es el más elocuente: la señalización lumínica, cuyas características de captura (intensidad, fondo, exposición) varían fuertemente entre ciudades, resulta casi irreconocible sin datos locales. Este hallazgo respalda empíricamente [16] y justifica la recolección de datos del entorno de despliegue por encima del mero aumento del volumen de datos no representativos. Es relevante que la mejora no sea uniforme sino estructurada (Tabla V): las clases de apariencia universal (automóvil, persona) ya eran competentes y mejoran marginalmente, mientras que las clases de señalización y de baja frecuencia, fuertemente dependientes del contexto local, concentran las mayores ganancias. Promediado sobre las clases con soporte suficiente, el modelo ajustado alcanza un mAP50 de caja de 0.68; la severidad de la brecha que medimos está en línea con el colapso de rendimiento del 97 % al 29 % (mAP@50) reportado al transferir un detector de señales a imágenes reales no representativas de su conjunto de entrenamiento [16].

V-B. La mototaxi como clase segmentable

La mototaxi, ausente de BDD100K [25] y de los repositorios internacionales [17], alcanza un mAP50 de caja y de máscara de 0.692 tras el ajuste fino. Que el contorno de un vehículo de tres ruedas con geometría atípica se segmente de forma fiable a partir de un conjunto local modesto sugiere que la barrera para los ADAS peruanos no es la dificultad intrínseca de la clase, sino la disponibilidad de datos anotados localmente.

V-C. Viabilidad sobre hardware de consumo

El entrenamiento completo se realizó en 41 min (el mejor modelo en 8 min) y la inferencia pura del modelo alcanza ≈ 35 FPS (28 ms/imagen), operando la aplicación completa a 14–43 FPS, todo sobre una GPU de consumo de gama de entrada (RTX 3050 de 4 GB). Esto contrasta con los requisitos habituales de los ADAS de investigación y refuerza la propuesta



Figura 7. Predicciones de segmentación de instancias del modelo ajustado sobre lotes de validación de Lima. Las máscaras y etiquetas corresponden a las 12 clases del esquema oficial.



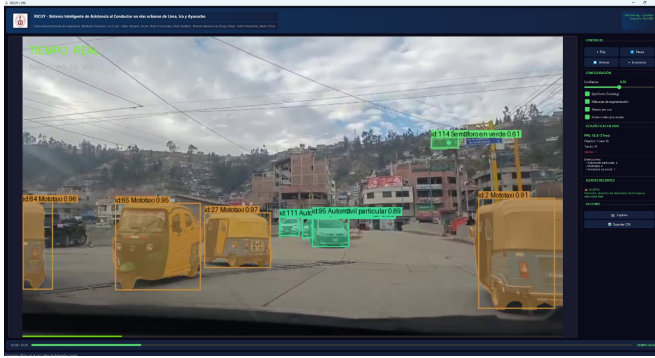
Figura 8. Aplicación RICUY en operación sobre secuencias de validación de Lima, mostrando máscaras de segmentación, identificadores de seguimiento, distancia monocular estimada y alertas activas (RTX 3050).

de RICUY como una solución de asistencia desplegable y de bajo costo, alineada con las restricciones económicas de los países en desarrollo [11].

VI. LIMITACIONES

Este estudio presenta limitaciones que delimitan el alcance de sus conclusiones y orientan el trabajo futuro. Primero, la evaluación cuantitativa se circunscribe a una partición de validación de Lima de 49 imágenes proveniente de una única partición determinista (semilla 42); la validación operativa en Andahuaylas es de carácter cualitativo, y las métricas de las

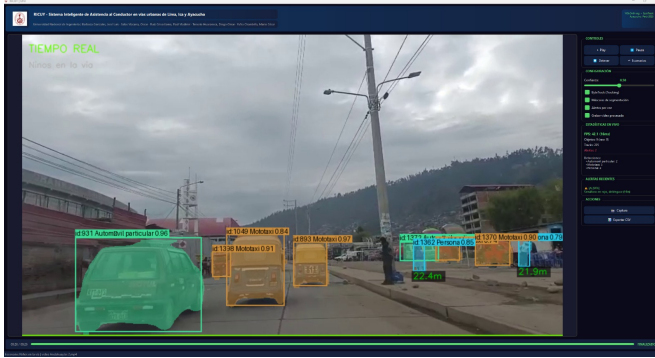
clases con pocas instancias deben interpretarse con cautela. La evaluación sobre 3 particiones independientes (Sección IV-D) confirma que el resultado principal es robusto a la partición, si bien un mayor número de particiones y una validación externa en otra ciudad lo reforzarían. Segundo, tres clases del esquema (reductor de velocidad, línea recta y línea recta y derecha) presentan un soporte casi nulo en el conjunto local y, en consecuencia, no aprenden señal útil; su detección fiable requerirá una campaña de anotación focalizada. Tercero, la estimación monocular de distancia se valida de forma cualitativa a través de la operación del sistema; una validación métrica



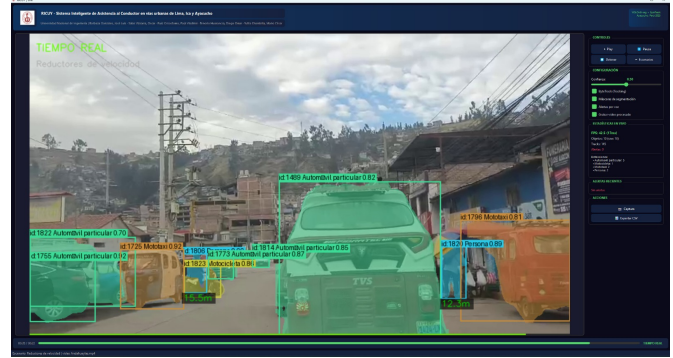
(a) Mototaxis y semáforo en verde en el escenario de niños en la vía.



(b) Alerta de semáforo en rojo con estimación de distancia.



(c) Mototaxis y un peatón con distancias estimadas.



(d) Escenario de tráfico denso.

Figura 9. Validación operativa de RICUY sobre secuencias de conducción naturalista de Andahuaylas, ciudad andina intermedia con alta penetración de mototaxi.

formal contra distancias de referencia (p. ej., con *ground truth* de telemetría) queda como trabajo futuro. Cuarto, no se realizó un análisis formal de explicabilidad; la verificación de que la atención del modelo se concentra en los objetos críticos mediante técnicas como Grad-CAM [28] permanece pendiente. Finalmente, la respuesta del conductor ante las alertas de voz no se evaluó con instrumentación biométrica.

VII. CONCLUSIONES

Se presentó RICUY, un Sistema Avanzado de Asistencia al Conductor de tiempo real que integra segmentación de instancias, seguimiento multiobjeto, estimación monocular de distancia y alertas de voz para el tráfico urbano peruano. El núcleo del trabajo es un estudio de transferencia y localización controlado que demuestra que el ajuste fino de un modelo YOLOv8L-seg andino [30] con un conjunto local de apenas 253 imágenes de Lima eleva el mAP50 de caja de 0.355 a 0.613 (+72.7 %) y el de máscara de 0.318 a 0.556 (+74.8 %) sobre una partición de validación idéntica, con el semáforo en rojo ascendiendo de 0.013 a 0.711. El sistema completo se validó de forma operativa sobre secuencias naturalistas de Lima y Andahuaylas a 14–43 FPS en una GPU de consumo de 4 GB, y segmentó de manera fiable la mototaxi, clase de exclusividad local ausente de los repositorios internacionales. Estos resultados confirman que, para el tráfico urbano peruano, la recolección de datos representativos del entorno de despliegue es más determinante que el volumen de datos genéricos, y que un ADAS de segmentación de instancias es viable sobre hardware de recursos limitados.

Trabajo futuro

Las líneas futuras incluyen la ampliación del conjunto local a las clases de marcas viales actualmente sub-representadas, la validación métrica formal de la estimación de distancia, un análisis de explicabilidad mediante Grad-CAM [28], la evaluación de arquitecturas más recientes de la familia YOLO [22] y la evaluación instrumentada de la respuesta del conductor ante las alertas en condiciones reales de conducción.

DISPONIBILIDAD DE DATOS

Las anotaciones de segmentación de instancias generadas en este estudio (253 imágenes, esquema de 12 clases, en formatos LabelMe y YOLOv8-seg, con la partición 204 entrenamiento/49 validación) están disponibles públicamente en <https://github.com/lucho19jose/Dataset-Ricuy> bajo licencia CC BY 4.0. Las imágenes de origen no se publican en abierto por contener rostros de peatones y placas vehiculares; se comparten bajo solicitud, para uso académico y de investigación, contactando al autor de correspondencia.

ABREVIATURAS

ADAS	Sistema Avanzado de Asistencia al Conductor (Advanced Driver Assistance System)
FCW	Alerta de Colisión Frontal (Forward Collision Warning)
FPS	Fotogramas por Segundo (Frames per Second)

Grad-CAM	Mapeo de Activación de Clase Ponderado por Gradiente (Gradient-weighted Class Activation Mapping)
IMU	Unidad de Medición Inercial (Inertial Measurement Unit)
MAE	Error Absoluto Medio (Mean Absolute Error)
mAP	Precisión Media Promedio (Mean Average Precision)
MTC	Ministerio de Transportes y Comunicaciones
NHTSA	Administración Nacional de Seguridad del Tráfico en las Carreteras (National Highway Traffic Safety Administration)
TTC	Tiempo hasta la Colisión (Time-to-Collision)
TTS	Texto a Voz (Text-to-Speech)
YOLO	You Only Look Once

REFERENCIAS

- [1] World Health Organization. *Global Status Report on Road Safety 2023*; WHO: Geneva, Switzerland, 2023.
- [2] World Health Organization. Global Leaders Make New Road Safety Commitments, Endorse New Declaration on Action to Reduce Road Deaths; News Release; WHO: Geneva, Switzerland, 18 February 2025 (4th Global Ministerial Conference on Road Safety, Marrakech, Morocco, 18–20 February 2025).
- [3] Hornberger, N.H.; Coronel-Molina, S.M. Quechua Language Shift, Maintenance, and Revitalization in the Andes: The Case for Language Planning. *Int. J. Sociol. Lang.* **2004**, *167*, 9–67. <https://doi.org/10.1515/ijsl.2004.025>.
- [4] Cerrón-Palomino, R. *Quechua Sureño: Diccionario Unificado*; Biblioteca Nacional del Perú: Lima, Peru, 1994.
- [5] Observatorio Nacional de Seguridad Vial (ONSV). *Boletín Estadístico de Siniestralidad Vial 2023*; Ministerio de Transportes y Comunicaciones: Lima, Peru, 2024.
- [6] Policía Nacional del Perú (PNP). *Anuario Estadístico de Accidentes de Tránsito 2023*; PNP: Lima, Peru, 2024.
- [7] Tenorio-Huaranca, D.O.; Lizarbe-Alarcon, H.; Ayala-Bizarro, R.G.; Tenorio-Palomino, M.G.; Bravo-Anaya, R.G.; Ircañaupa-Huamani, A.S.; Leon-Palacios, E.; Bellido-Aedo, V. Intelligent Traffic Light Optimization System Using Convolutional Neural Networks for Historic City Centers in Complex Scenarios. *Math. Model. Eng. Probl.* **2025**, *12*, 3247–3264.
- [8] Sukkar, M.; Jadeja, R.; Shukla, M.; Mahadeva, R. A Survey of Deep Learning Approaches for Pedestrian Detection in Autonomous Systems. *IEEE Access* **2025**, *13*, 3994–4007. <https://doi.org/10.1109/ACCESS.2024.3524501>.
- [9] Ayachi, R.; Said, Y.; Afif, M.; Alshammari, A.; Hleili, M.; Abdelali, A.B. Assessing YOLO Models for Real-Time Object Detection in Urban Environments for Advanced Driver-Assistance Systems. *Alex. Eng. J.* **2025**, *123*, 530–549.
- [10] Liu, D.; Zhang, D.; Wang, L.; Wang, J. Semantic Segmentation of Autonomous Driving Scenes Based on Multi-Scale Adaptive Attention Mechanism. *Front. Neurosci.* **2023**, *17*, 1291674. <https://doi.org/10.3389/fnins.2023.1291674>.
- [11] Dai, X.; Yin, Z.; Fang, C.; Chen, T.; Tian, Y. Multi-Task Learning for Simultaneous Monocular Depth Estimation and Object Detection. *IEEE Intell. Transp. Syst. Mag.* **2021**, *13*, 218–232.
- [12] Chaman, M.; El Maliki, A.; Dahou, H.; Hadjoudja, A. Benchmarking YOLO-Based Deep Learning Models for Real-Time Object Detection in Hybrid ADAS and Intelligent Transportation Systems. *Results Eng.* **2026**, *29*, 108942. <https://doi.org/10.1016/j.rineng.2025.108942>.
- [13] National Highway Traffic Safety Administration. *A Test Track Protocol for Assessing Forward Collision Warning Driver-Vehicle Interface Effectiveness*; Report No. DOT HS 811 501; NHTSA: Washington, DC, USA, 2011.
- [14] Park, S.; Kim, B.; Kim, C.; Kim, J. Pedestrian Detection Based on Monocular Camera Using Geometry Constraint. *Int. J. Adv. Robot. Syst.* **2014**, *11*, 153.
- [15] Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Weng, F.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2022; pp. 1–21.
- [16] Linardi, R.; Soo, N.; Nabua, A.M.; Adhi, C.G.S.; Engel, V.J.L. Traffic Sign Detection Using YOLO: Evaluating Failures on Localized Data. *Procedia Comput. Sci.* **2025**, *269*, 1494–1501.
- [17] Akujobi, M.U.; Abubakar, A.U.; Mailabari, R.J.; Thuku, I.T.; Musa, S.Y.; Visa, I.M.; Abioye, A.O. Keke-Aware Vehicle Counting for Traffic Measurement Using YOLO: Dataset and Field Evaluation. *Appl. Sci.* **2026**, *16*, 4316. <https://doi.org/10.3390/app16094316>.
- [18] Wibowo, A.; Trilaksono, B.R.; Hidayat, E.M.I.; Munir, R. Object Detection in Dense and Mixed Traffic for Autonomous Vehicles with Modified YOLO. *IEEE Access* **2023**, *11*, 134866–134877. <https://doi.org/10.1109/ACCESS.2023.3335826>.
- [19] Kaya, Ö.; Çodur, M.Y.; Mustafaraj, E. Automatic Detection of Pedestrian Crosswalk with Faster R-CNN and YOLOv7. *Buildings* **2023**, *13*, 1070.
- [20] Russon, D.; Guennec, A.; Naredo-Turrado, J.; Xu, B.; Boussuge, C.; Battaglia, V.; Hiron, B.; Lagarde, E. Evaluating Pedestrian Crossing Safety: Implementing and Evaluating a Convolutional Neural Network Model Trained on Paired Aerial and Subjective Perspective Images. *Heliyon* **2025**, *11*, e42428. <https://doi.org/10.1016/j.heliyon.2025.e42428>.
- [21] Kulawik, J.; Kuczyński, Ł. AI-Based Detection and Classification of Horizontal Road Markings in Digital Images Dedicated to Driver Assistance Systems. *Appl. Sci.* **2025**, *15*, 12189. <https://doi.org/10.3390/app152212189>.
- [22] Jocher, G.; Qiu, J. Ultralytics YOLO26. 2026. Available online: <https://github.com/ultralytics/ultralytics/tree/main> (accessed on 8 April 2026).
- [23] Fang, S.; Zhang, B.; Hu, J. Improved Mask R-CNN Multi-Target Detection and Segmentation for Autonomous Driving in Complex Scenes. *Sensors* **2023**, *23*, 3853. <https://doi.org/10.3390/s23083853>.
- [24] Microsoft Corporation. Edge TTS: Text-to-Speech via Microsoft Azure Neural Voices. 2024. Available online: <https://github.com/rany2/edge-tts> (accessed on 8 April 2026).
- [25] Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; Darrell, T. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In *Proceedings of the IEEE/CVF CVPR*; IEEE: Seattle, WA, USA, 2020; pp. 2636–2645.
- [26] Russell, B.C.; Torralba, A.; Murphy, K.P.; Freeman, W.T. LabelMe: A Database and Web-Based Tool for Image Annotation. *Int. J. Comput. Vis.* **2008**, *77*, 157–173.
- [27] Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLOv8. 2023. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 8 April 2026).
- [28] Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *Proceedings of the IEEE ICCV*; IEEE: Venice, Italy, 2017; pp. 618–626.
- [29] Urbina-Dominguez, E.J.; Lizarbe-Alarcon, H.; Galvez-Gastelu, Y.; Porras-Flores, E.E.; Moncada-Sosa, W.; Estrada-Cardenas, J.E.; Leon-Palacios, E.; Tenorio-Huaranca, D.O. ADAS-TSR: A Deep Learning-Based Traffic Sign Recognition System with Voice Alerts for Andean Historic City Centers. *Appl. Sci.* **2026**, *16*, 2664. <https://doi.org/10.3390/app16062664>.
- [30] De la Cruz-Moran, A.; Lizarbe-Alarcon, H.; Moncada, W.; Bellido-Aedo, V.; Carrasco-Badajoz, C.; Rayme-Chalco, C.; Aldana-Yarlequé, C.; Saavedra, Y.; Saavedra, E.; Pereda, A. QHAWAY: An Instance Segmentation and Monocular Distance Estimation ADAS for Vulnerable Road Users in Informal Andean Urban Corridors. *Sensors* **2026**, *26*, 2569. <https://doi.org/10.3390/s26082569>.