Adaptación de YOLO Versión 5 Para Detección de Vehículos en Espacio 3D sobre Imágenes Monoculares.

Caicedo M. Yesid, *Estudiante de la Maestría en Inteligencia Artificial*, Parra R. Carlos, *Profesor Titular*, Wilches P. Carlos, *Profesor Cátedra*.
FACULTA DE INGENIERÍA
PONTIFICIA UNIVERSIDAD JAVERIANA.

***Resumen*— Este proyecto consistió en adaptar el modelo YOLO versión 5 para detectar y clasificar vehículos en un espacio tridimensional, y con esto, evaluar el riesgo de colisión trasera en Bogotá usando imágenes RGB de videos de vigilancia. El modelo YOLO versión 5 entrenado tuvo un mAP@50 de 91.8% y se usó para estimar la longitud de los vehículos utilizando la perspectiva, en particular, los puntos de fuga. Se dividió la región de interés (ROI) en tres zonas según estos puntos de fuga, lo que permitió calcular distancias aproximadas entre par de vehículos consecutivos. A partir de la información recopilada y los resultados obtenidos, se definieron categorías de riesgo de colisión y se ajustaron las reglas y funciones de pertenencia para reflejar la realidad considerando la normatividad del tránsito sobre la separación espacial entre vehículos en circulación. Entre los resultados, se consiguió un modelo con un desempeño satisfactorio y se da a conocer la metodología para pasar de detecciones en un espacio bidimensional a uno tridimensional. Además, se observó que los conductores no respetan en gran medida las distancias de seguridad sugeridas ya que el 95.5% de las distancias calculadas se clasificaron en riesgo alto y medio, lo que incrementa la posibilidad de accidentes viales en la ciudad.**

***Índice de Términos* — monocular, visión, detección de objetos, 3d, RGB, tráfico, YOLO, puntos de fuga.**

I. Introducción

Debido al alto número de muertes por accidentes de tránsito a nivel mundial resalta la necesidad de contar con información que contribuya a mitigar los riesgos a los que se enfrentan los actores viales (conductores y peatones). Esta información resulta fundamental para la toma de decisión y poder reducir la alta tasa que prevalece, principalmente, en las ciudades[1]. Los accidentes de tránsito pueden estar asociados a diversas causas como el exceso de velocidad, la congestión vial, la incompetencia, el descuido, la conducción temeraria, etc. [2].

Existe una amplia gama de investigaciones[3][4][5] y aplicaciones [6]–[9] [10] que promueven diversas metodologías para desarrollar sistemas inteligentes de monitoreo de tráfico. Estos sistemas tienen como objetivo prevenir colisiones en intersecciones [11] y abordar uno de los accidentes de tránsito más frecuentes, como las colisiones traseras[2]. La importancia de los sistemas de monitoreo del tráfico radica en su capacidad para detectar y seguir vehículos de cualquier tipo, lo cual resulta altamente beneficioso para una variedad de aplicaciones como la seguridad de los conductores[10], [12], [13] y seguimiento de accidentes [14].

De acuerdo con la organización mundial de la salud (OMS) [15] la siniestralidad vial representa un importante problema de salud pública mundial, causando la muerte de alrededor de 1,3 millones personas en todo el mundo en colisiones de tránsito, siendo la primera causa de muerte en grupos de edad infantil y juvenil. La OMS también destaca que, desde la creación del automóvil, más de 50 millones de personas han dejado la vida en las carreteras, una cifra superior al número de muertos en la primera guerra mundial o como resultado de algunas de las peores epidemias. Además, a esto se suma desde el punto de vista económico, según el doctor Etienne Krug, que las muertes por colisiones de tránsito cuestan anualmente a los países cerca del 3% de su PIB. En este sentido, resulta urgente mejorar la seguridad de los sistemas de movilidad con el fin de salvar vidas y promover entornos viales saludables [15] . En consecuencia, la declaración política bajo el lema '*The 2030 horizon for road safety: securing a decade of action and delivery*'. Los gobiernos de todo el mundo se comprometieron para el año 2.030 reducir en un 50% las muertes y los traumatismos causados por las colisiones de tránsito, constituyendo esto un hito para la seguridad vial y la movilidad sostenible [15].

En el ámbito nacional, se registraron 145.921 siniestros viales durante el año 2021, de los cuales 108.027 fueron el resultado de colisiones entre vehículos, lo que representa un 74% del total. En cuanto a las personas fallecidas dentro de los 30 días posteriores a los siniestros, se reportaron 7.238 casos, principalmente en el grupo de edad comprendido entre los 20 y 30 años. Es importante destacar que el 59.7% de estas víctimas eran conductores de motocicletas, siendo esta la cifra más alta registrada en los últimos 10 años [1].

En el ámbito local, en la ciudad de Bogotá D.C., se reportaron

28.418 siniestros viales y se registraron 494 personas fallecidas dentro de los 30 días posteriores a los siniestros [1].

Con base en lo expuesto, la seguridad vial se presenta como un campo de acción fundamental para la inteligencia artificial, especialmente en el área de visión por computador. Esta tecnología permite la detección de vehículos y peatones, lo que brinda la capacidad de comprender su comportamiento y proporcionar información estadística para la toma de decisiones. Esto contribuye a reducir el riesgo de accidentes y mejorar el flujo del tráfico, en última instancia, mejorando la calidad de vida de los ciudadanos; los avances en la detección de vehículos en tiempo real gracias a la visión por computador han llevado a mejoras significativas en la precisión de las estimaciones, lo que aumenta las posibilidades de obtener resultados más efectivos en la protección de vidas humanas [16] [17], [18].

Dentro del campo de la inteligencia artificial, existen diversas ramas de aplicación. Sin embargo, en el ámbito de la seguridad vial, el enfoque se centra en modelos de detección y clasificación de objetos en un plano bidimensional (2D) o tridimensional (3D). De este último, la capacidad de recuperar información sobre los objetos en este entorno abre un amplio abanico de aplicaciones, como la navegación robótica [19], la inspección de infraestructuras [20], el cine en 3D [21], los vehículos autónomos [22], [23] y la seguridad vial para reducir el riesgo al que están expuestos los actores viales, que es relevante para este proyecto [6], [24]. Lo cual, es importante destacar que la recuperación de información sobre objetos en 3D presenta desafíos significativos para los modelos de visión por computador, y el progreso en esta área ha estado condicionado por el tipo de formato de la imagen [25].

Adicionalmente, los modelos de visión por computador se enfrentan a dificultades durante su entrenamiento, como la eficiencia en las estimaciones y la minimización del costo computacional. Estos son aspectos clave que requieren una atención cuidadosa para lograr resultados óptimos.

Dentro del ámbito de la visión por computador aplicada a la seguridad vial, se han identificado tres tipos principales de imágenes utilizadas para entrenar los modelos: RGB, RGB-Depth y nube de puntos. Cada uno de estos tipos de imágenes ha dado lugar al desarrollo de arquitecturas y métodos de extracción de características específicos, siendo las redes neuronales convolucionales uno de los enfoques más utilizados. Estas arquitecturas se combinan con diversas técnicas para realizar predicciones de cuadros delimitadores en 2D o 3D [25].

En el caso particular de este proyecto, se trabaja con imágenes monoculares o RGB. Estas imágenes son capturas de un instante del mundo real en un plano bidimensional (2D) [26] y presentan la particularidad de ser simples, es decir, no proporcionan información sobre la profundidad de los objetos capturados. En el campo de la seguridad vial, es importante y necesario realizar estimaciones de la profundidad o la longitud de los objetos. Para lograr esto, se utilizan diversas técnicas, como modelos con plantillas, redes neuronales, propiedades geométricas o combinaciones de estas metodologías, que permiten obtener estimaciones de cuadros delimitadores en 3D [23], [27]–[32].

Actualmente se viene realizando es la fusión de los distintos métodos de detección como por ejemplo redes neuronales convolucionales en 3D y para afinar las estimaciones usan arquitecturas como PointNet [32] y con esto se ha demostrado mejorar el rendimiento de los modelos [33].

Efectivamente, con base en lo expuesto, es de vital importancia salvar vidas y reducir el riesgo de colisiones entre vehículos en la ciudad de Bogotá. Para lograr esto, es necesario contar con información confiable y precisa que respalde la toma de decisiones en políticas de movilidad y mejora de las zonas viales. Este proyecto puede complementarse con otras investigaciones y aplicaciones que contribuyan a tomar decisiones más acertadas en el ámbito de la movilidad y el mejoramiento de las condiciones de tránsito en la ciudad.

Asimismo, se espera que este proyecto sea un valioso aporte y una guía para futuras aplicaciones que puedan ser realizadas por la comunidad. Pues es importante fomentar la colaboración y el intercambio de conocimientos para continuar mejorando la seguridad vial y promoviendo una movilidad más segura y eficiente en la ciudad.

El orden del documento se compone por la sección II que contiene los artículos relacionados con este proyecto, la sección III los objetivos, la IV son los materiales y métodos, ésta comprende los procesamientos a las imágenes, descripción de la arquitectura utilizada, metodología del entrenamiento y el cómo con las detecciones en 2D se logró convertir éstas en detecciones 3D y al final de esta sección, los resultados con sus respectivos análisis. Seguido, se tiene la sección V donde se dan algunas conclusiones y recomendaciones para trabajos futuros del presente proyecto.

II. Trabajos relacionados

En esta sección se presentan enfoques relevantes en visión por computador para la detección en 3D utilizando imágenes RGB. Se destacan métodos que se centran en la fusión de diferentes tipos de imágenes y la detección en 2D con YOLO versión 5 (denotado de ahora en adelante como YOLOv5), que luego se transforma en detecciones en 3D [34], [35]. Además, se mencionan otros métodos que utilizan restricciones geométricas para la detección de vehículos en 3D [36] y métodos basados exclusivamente en aprendizaje profundo para la detección de profundidad en imágenes [37]

Asimismo, se presentan investigaciones adicionales, como un nuevo método propuesto para la detección de puntos de fuga en escenarios de conducción real [38]. También se describe una solución basada en redes neuronales convolucionales y regresión del mapa de calor para la detección de puntos de fuga en carreteras destapadas[39]. Estas investigaciones sirven como referencia y guía para el desarrollo del presente proyecto.

A. Investigaciones basadas en aprendizaje profundo.

En primer lugar, se destaca la referencia [33] pues hace una revisión exhaustiva de 100 investigaciones en el área de interés y se enfoca en dar a conocer todas las etapas para llevar a cabo la detección de objetos en imágenes, incluyendo tipos de datos, representaciones de estos, extractores de características y

funcionalidades de los modelos de detección de objetos. Respecto a este último, las investigaciones más recientes se centran en el uso de técnicas de aprendizaje profundo por su flexibilidad y mayor rendimiento, dentro de este se cuenta con marcos de detecciones con y sin anclajes y detecciones híbridas. Entre las detecciones con anclajes se pueden tener marcos de una etapa de la familia YOLO [40] ya que se caracterizan por su simplicidad y rápidos al transformar los datos y emplean una red convolucional para estimar directamente los cuadros delimitadores y los niveles de confianza de los objetos detectados. Por otro lado, los marcos de dos etapas que son de la familia R-CNN [36], se enfocan primero en identificar regiones potenciales de posibles objetos y en la segunda etapa se tienen refinamientos de estas estimaciones a través de un análisis más detallado de la extracción de características logrando que la capa final genere la clasificación y los cuadros delimitadores. Caso diferente sucede con los enfoques de detección sin anclajes porque se basa en estimaciones puntuales o por segmentos y pueden presentar un mayor costo computacional. Por último, los enfoques de marcos híbridos combinan métodos con y sin anclajes para lograr mejorar las estimaciones [33].

Por lo tanto, exponen una variedad de modelos con base al tipo de imágenes y combinación de los marcos, métodos y técnicas. Sin embargo, como el presente proyecto se enmarca en imágenes monoculares. Se procedió a revisar en detalle uno de los 19 modelos mencionados, específicamente el modelo SHIFT R-CNN [36].

La referencia [36] propone una red neuronal usando restricciones geométricas para realizar las detecciones en 3D en imágenes RGB. En general este modelo está compuesto por 3 etapas para realizar las predicciones en tercera dimensión:

1. En esta primera etapa se cuenta con una ResNet-101 previamente entrenada con el conjunto de datos KITTI para estimar los cuadros delimitadores en 2D.
2. En la segunda etapa, un sistema de ecuaciones que es resuelto por medio de mínimos cuadrados que posteriormente, estas estimaciones y la matriz de proyección de la cámara permite lograr un ajuste preciso teniendo en cuenta la traslación entre 2D y 3D.
3. En la tercera etapa se tiene una red completamente conectada llamada ShiftNet, que aprende la dependencia del error de las etapas anteriores y corrige, refinando así los cuadros delimitadores.

Para llevar a cabo los experimentos, los investigadores utilizaron la base de datos KITTI, que cuenta con 3.712 imágenes para entrenamiento y 3.769 para validación. Además, establecieron 3 niveles de dificultad para las estimaciones: fácil, medio y difícil. Ya teniendo esto hicieron comparación de los AP con otros métodos de detección y lograron resultados superiores utilizando la red propuesta. Por mencionar algunas comparaciones, el modelo DeepBox [41] fue superado en detección y clasificación de vehículos en la categoría “difícil”, en un 1%. En el caso de la detección de peatones, la diferencia en el puntaje de precisión (AP) con respecto al método OFT-Net [5] fue del 9,53%. Para la detección de ciclistas, la diferencia en los AP fue de 3,40%. Estos resultados demuestran la competitividad de la arquitectura propuesta en este estudio.

La referencia [37] propone una red llamada DLCN (Dynamic Locally Dilated Convolutional Network) para abordar las dificultades en la detección de la profundidad en imágenes 2D. Esta red aprovecha la información de profundidad obtenida de otras redes neuronales previamente entrenadas, como DORN, PSMNET y DispNet, para guiar la generación de representaciones 3D más precisas a partir de imágenes RGB.

El enfoque propuesto por DLCN utiliza filtros dinámicos, profundos y con campos receptivos adaptables (dilatación) para cada píxel y canal de diferentes imágenes. La arquitectura DLCN se basa en una ResNet-50 y está diseñada de la siguiente manera:

1. Un filtro ejemplar aprende la geometría de una escena específica en cada imagen.
2. La convolución local es para distinguir las regiones de los objetos y el fondo para cada píxel.
3. La convolución en profundidad es para aprender diferentes filtros de canal en una capa convolucional con diferentes propósitos, además de reducir complejidad computacional.
4. Cada filtro aprende a partir de tres valores de dilatación definidos, lo que permite que los filtros desempeñen diferentes funciones en el modelo. Esto significa que diferentes píxeles de diferentes imágenes tienen filtros específicos (algunos filtros se especializan en identificar objetos con diferentes escalas).

En cuanto a los resultados, el enfoque DLCN logra una mejora del 9.1% en comparación con el estado del arte utilizando la base de datos KITTI del año 2020. Esto demuestra la efectividad y competitividad de la arquitectura propuesta en la tarea de estimación de profundidad en imágenes 2D.

En la referencia [34] se proporciona una plataforma de monitoreo en tiempo real con el objetivo de detectar y clasificar peatones y 10 tipos de vehículos. Adicional a esto, se logra mejora en el algoritmo de seguimiento SORT usando filtro Kalman para estimar la velocidad de los objetos. Lo interesante de este artículo es que se emplea una combinación de imágenes de una cámara de vigilancia (CCTV - RGB) con imágenes satelitales fusionándolas para realizar las estimaciones de la trayectoria. Particularmente, para lograr esto calibraron la cámara usando los parámetros como altura y ángulo, además de hacer algunas correcciones como la distorsión radial, eliminar el fondo y hacer coincidencia de los histogramas. Una vez realizadas estas correcciones, fue posible realizar una transformación de vista de pájaro de 2D a 3D y viceversa para engranar los dos tipos de imágenes.

En cuanto a la arquitectura utilizada para las detecciones, emplearon un modelo pre-entrenado YOLOv5 que ofrece una de las mejores velocidades y precisiones en las estimaciones. Este modelo logró un mAP de 84% superando en 4% el estado del arte de YOLO v1, v2, v4, ensambles de Faster y ssd300.

Además, los resultados evidencian un análisis de riesgo de los actores viales a partir de mapas de calor donde se ilustra la alta densidad de uso del espacio, interacciones y velocidades entre peatones y vehículos.

La referencia [35] se presenta un marco de trabajo denominado "Multi-feature Fusion VoteNet" (MFFVoteNet) que tiene como objetivo mejorar el rendimiento de la detección de objetos 3D en escenas desordenadas y con oclusión. Una característica interesante de este artículo es que se utiliza tanto la información de los puntos en la nube como la imagen en RGB como entrada para detectar objetos en el espacio 3D. Además, se propone un método de supresión no máxima de proyección (PMNS) en la detección de objetos en este espacio 3D para eliminar propuestas redundantes de cuadros delimitadores.

La arquitectura de MFFVoteNet consta de un módulo de votación, un módulo de predicción, un módulo de codificación de imágenes y el algoritmo PNMS. Este enfoque se basa en una arquitectura ResNet-18. Para validar el modelo, se utilizó el conjunto de datos ScanNetv2, que presenta un alto grado de complejidad. Los resultados obtenidos superaron a los modelos de estado del arte, como HGNet [42], MR CNN 2D-3D [43] y DSS [44]. En la etapa de prueba, se logró un mAP un 2.6% más alto que HGNet y hasta un 48.7% más alto que DSS.

Estos resultados demuestran que MFFVoteNet es altamente competitivo en la detección de objetos 3D en escenas desordenadas y con oclusión, superando a los modelos previos en términos de precisión y rendimiento.

B. Investigaciones basadas en puntos de fuga.

La referencia [38] presenta el problema de la detección de puntos de fuga en escenarios de conducción real. Los puntos de fuga son puntos donde las líneas paralelas en el mundo real parecen converger en la imagen. La detección de puntos de fuga es útil para estimar la orientación y la profundidad de la escena, así como para detectar obstáculos y carriles. El artículo revisa los métodos existentes para la detección de puntos de fuga y sus limitaciones, y propone un nuevo método basado en características del espacio de filas que es rápido y robusto. Este método consta de cuatro pasos:

Extracción de bordes: Este paso consiste en aplicar un detector de bordes a la imagen para obtener los píxeles que forman parte de los bordes de los objetos. El detector de bordes usado en el estudio es el algoritmo de Canny, que es un método clásico basado en el gradiente de la intensidad de la imagen.

Agrupación de bordes: En esta fase se agrupan los píxeles de bordes en segmentos de línea recta usando un algoritmo basado en la transformada de Hough probabilística. Este algoritmo busca las líneas que mejor se ajustan a los píxeles de bordes usando un criterio probabilístico y un umbral de distancia. El resultado es un conjunto de segmentos de línea recta que representan los bordes de los objetos.

Extracción de características del espacio de filas: consiste en extraer las características del espacio de filas para cada segmento de línea recta. El espacio de filas es el espacio ortogonal al espacio columna de una imagen, que se puede obtener aplicando la descomposición en valores singulares a la matriz de la imagen. Las características del espacio de filas son las coordenadas del vector normal al segmento de línea recta en el espacio de filas. Estas características son invariantes a la escala y la rotación de la imagen, y reflejan la dirección de las líneas paralelas en el mundo real.

Clasificación de puntos candidatos a puntos de fuga: En este último paso se clasifican los segmentos de línea recta en dos clases: los que apuntan al punto de fuga y los que no. Para ello, se usa un clasificador lineal entrenado previamente con ejemplos positivos y negativos. El clasificador lineal usa las características del espacio de filas como entrada y devuelve una puntuación que indica la probabilidad de que el segmento apunte al punto de fuga. Los segmentos con una puntuación mayor que un umbral se consideran candidatos a puntos de fuga.

Este método propuesto se comparó con método de bordes, método de Kong, método HrNet, entre otros del estado del arte, los cuales, los experimentos se realizaron con un conjunto de datos sintético que contiene imágenes generadas por un simulador de conducción con diferentes condiciones de iluminación, clima y tráfico. Y un conjunto de datos real, que contiene imágenes capturadas por una cámara montada en un vehículo en diferentes escenarios urbanos y rurales.

Los resultados muestran que el método propuesto es más rápido y preciso que los métodos comparados, tanto en el conjunto de datos sintético como en el real. Por lo tanto, el método es capaz de detectar puntos de fuga en tiempo real y con alta precisión, lo que lo hace adecuado para escenarios de conducción real. Sin embargo, el método tiene algunas limitaciones, como la dependencia del parámetro del umbral de agrupación de bordes, la sensibilidad al ruido y la oclusión, y la incapacidad para detectar múltiples puntos de fuga.

A pesar de esto, los autores mencionan que el método propuesto es un avance significativo en la detección de puntos de fuga y que tiene potencial para ser usado en aplicaciones como la navegación autónoma o la reconstrucción 3D.

Por último, la referencia [39] presenta una solución novedosa para la detección de puntos de fuga en carreteras no estructuradas (carreteras destapadas) utilizando redes neuronales convolucionales (CNN por sus siglas en inglés) y regresión del mapa de calor. La primera etapa utiliza una CNN (HrNet modificada) para extraer características de la imagen y generar un mapa de calor que indica la probabilidad de que cada píxel sea un punto de fuga. La CNN está compuesta por varias capas convolucionales, capas de activación ReLU capas de agrupación máxima y una capa de salida. La capa de salida produce un mapa de calor con la misma resolución que la imagen de entrada, y el mapa de calor representa la distribución espacial del punto de fuga en la imagen. En la segunda etapa, la regresión de mapa de calor se utiliza para localizar el punto de fuga a partir del mapa de calor generado por la CNN. La regresión de mapa de calor consiste en aplicar una función gaussiana al mapa de calor para suavizarlo y luego encontrar el máximo local del mapa suavizado, siendo este valor el correspondiente a las coordenadas del punto de fuga en la imagen.

En este estudio se compara la solución propuesta con los métodos existentes basados en RANSAC (RANdom SAmple Consensus), transformación de Hough y CNN. Demostrando que la solución propuesta supera a los métodos existentes en

términos de precisión y robustez, y que puede detectar puntos de fuga en diferentes condiciones ambientales y escenarios complejos.

III. Pregunta de Investigación y Objetivo Principal

En esta sección, se plantea la pregunta de investigación y se establecen los objetivos que guiaron el desarrollo del presente proyecto.

A. Pregunta de investigación:

¿Cómo contribuir a la disminución del riesgo de colisión trasera en zonas específicas de la ciudad de Bogotá?

Con base en la revisión del estado del arte y considerando el contexto de riesgo de accidentes, se ha identificado que el uso de técnicas de aprendizaje profundo puede ser de gran utilidad. Un ejemplo destacado es el caso de la ciudad de Leeds, en Inglaterra, donde se adaptó el modelo YOLOv5 [34] para el monitoreo del tráfico en una intersección concurrida, involucrando a diversos actores viales.

En este sentido, se propone llevar a cabo una adaptación similar de dicho modelo en la ciudad de Bogotá, con el fin de proporcionar información precisa sobre el riesgo de colisión entre vehículos. La aplicación de este enfoque permite obtener información relevante sobre una de las zonas que se lograron recolectar y poder identificar situaciones de alto riesgo y tomar medidas preventivas que contribuyan a reducir la accidentalidad en la ciudad de Bogotá.

B. Objetivo General:

El objetivo principal de este proyecto es adaptar el modelo YOLO versión 5 medium P6 (YOLOv5m-P6) para la detección y clasificación de vehículos en un espacio tridimensional (3D) en la ciudad de Bogotá. El enfoque se centra en medir la distancia entre los vehículos detectados y proporcionar información que permita evaluar el nivel de riesgo de colisión.

C. Objetivos Específicos:

1. Extraer escenas de los videos de vigilancia compartidos por la Secretaria Distrital de Movilidad.
2. Seleccionar y etiquetar escenas extraídas para ingresar a la red neuronal YOLOv5m-P6.
3. Entrenar y refinar la red neuronal YOLOv5m-P6 usando las escenas etiquetas de la ciudad de Bogotá.
4. Identificar el método adecuado para generar cuadros delimitadores en el espacio tridimensional a partir de las detecciones realizadas en el espacio bidimensional por el modelo YOLOv5m-P6.
5. Calcular la distancia entre vehículos detectados en espacio de tercera dimensión.

IV. Métodos y Materiales

En esta sección se describe el procesamiento de las escenas obtenidas de los videos, la arquitectura de YOLOv5m-P6, el entrenamiento del modelo, el procesamiento para obtener las estimaciones en tercera dimensión de los vehículos detectados y, finalmente; el cálculo de las distancias entre vehículos consecutivos para evaluar el riesgo de colisión.

Cada uno de los procesos mencionados se presentan de forma resumida en la Fig. 1. Permitiendo esto tener un panorama general de la solución propuesta en este proyecto.

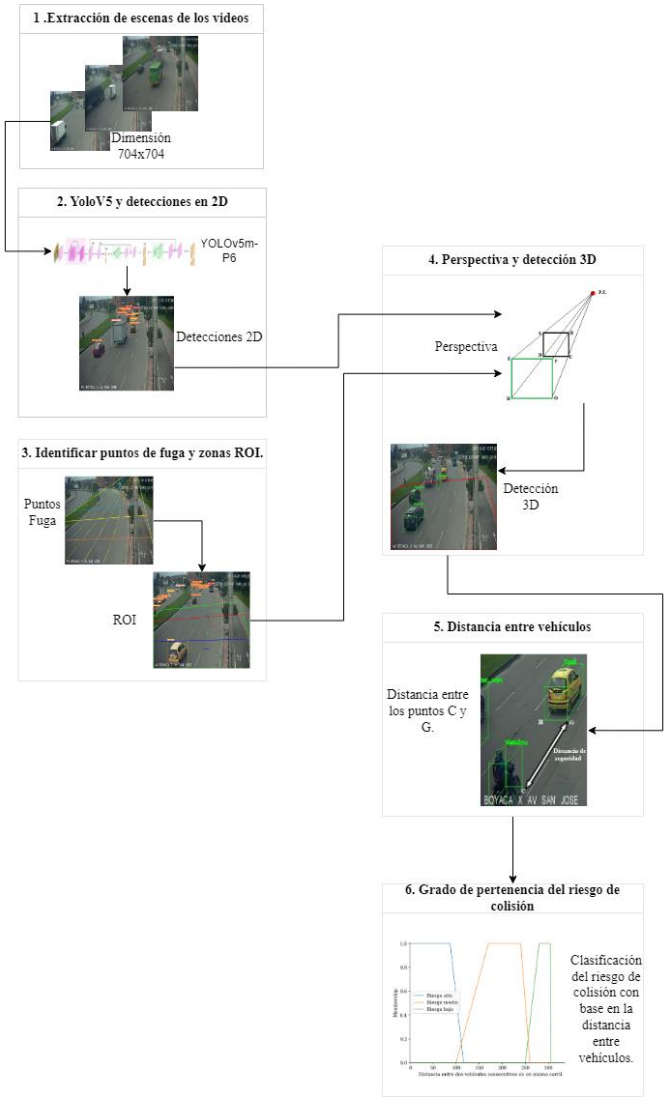


Fig. 1 Etapas resumidas de la solución del proyecto; la imagen de la red neuronal en la etapa 2 es tomada de [45]

A. Datos

Se obtuvieron un total de 11 videos de vigilancia de movilidad de la ciudad de Bogotá, proporcionados por la Secretaría Distrital correspondiente. Cada video tiene una duración de 10 minutos, y se extrajeron aproximadamente 400 escenas de cada uno, con la excepción del video de la Cra 96 con Avenida el Dorado, el cual tenía una toma de la carretera muy reducida, como se puede apreciar en la Fig. 2. En total, se obtuvieron 4.096 escenas, las cuales pasaron por un proceso de selección y etiquetado.

B. Etiqueta de los datos

Para la selección de las escenas, se aplicaron criterios específicos con el objetivo de capturar vehículos motorizados y no motorizados que se encontraran cerca de la región de interés (ROI), es decir, la calzada más próxima a la cámara. Esto permitiría obtener mediciones más precisas de la distancia entre pares de vehículos. Después de aplicar estos criterios, se obtuvo un total de 1.565 escenas seleccionadas. Seguido a esto, se realizó el etiquetado de los objetos en la plataforma en línea de Roboflow [46] donde se definieron 5 clases con base a la similaridad entre vehículos [47]. Así, las 5 clases se ilustran el Fig. 3.

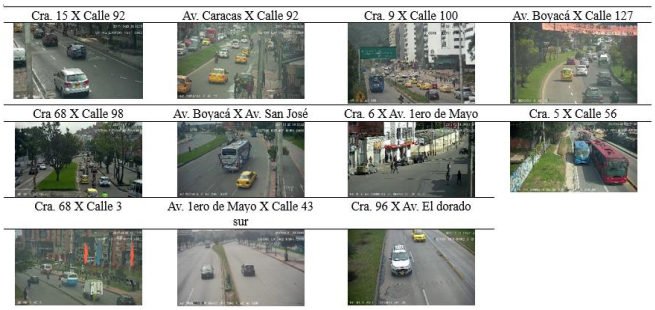


Fig. 2 Escenas de cada video, abreviaciones: Cra: carrera, Av.: Avenida.

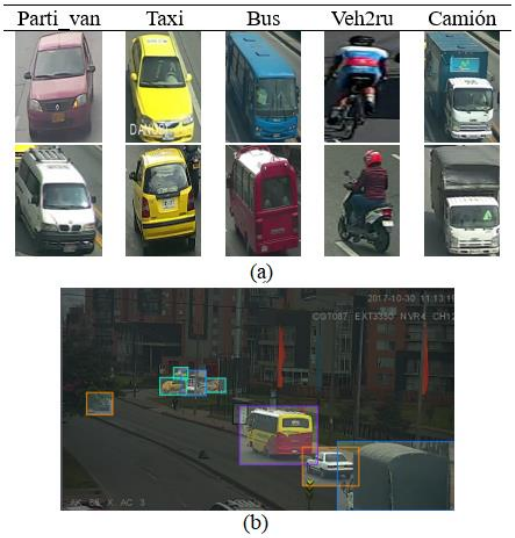


Fig. 3 (a) Las 5 clases definidas, (b) Ilustración de las etiquetas en Roboflow.

Detalle de los tipos de vehículos que están en cada clase:

1. **Par_van:** automóvil, camioneta, van.
2. **Bus:** microbús, bus.
3. **Camión:** camiones de carga pesada, camiones de basura, camiones pequeños, volquetas.
4. **Veh2ru:** motocicleta y bicicleta.
5. **Taxi.**

En la Fig. 4 se ilustra la distribución de las etiquetas por clases de las 1.565 escenas, se observa identifica que la clase *Par_van* tiene la mayor cantidad de etiquetas, mientras *camión* que es la de menor número de etiquetas (instancias).

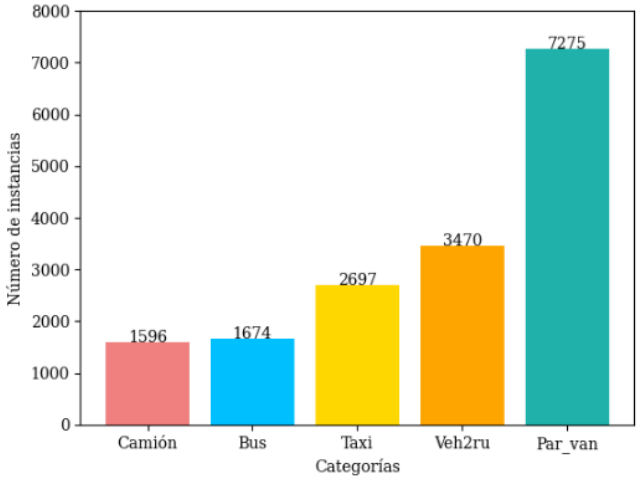


Fig. 4 Número de instancias por categoría.

C. Arquitectura de YOLOv5m-P6[1] y Métricas.

La elección de YOLOv5 se basó en su rendimiento [48], así como en su flexibilidad en el tamaño de la red, ya que ofrece diferentes variantes: nano, small, medium, large y extra-large.

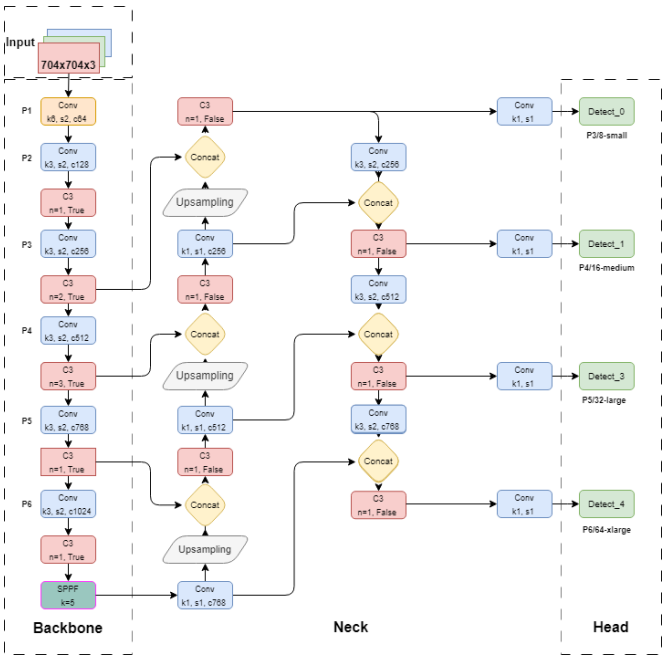
Además, YOLOv5 se destaca por ser la primera red de la familia YOLO[49] nativa de pytorch, implicando mayor facilidad para su desarrollo y experimentación. Otra ventaja es que ha pasado más de dos años desde su lanzamiento lo que permite tener un amplio soporte por *ultralitycs* y la comunidad [50] lo cual permite ayudar a superar distintos errores o necesidad específicas durante su uso.

En este proyecto, se optó por entrenar inicialmente los modelos *small* y *medium* de YOLOv5, que tiene 12.6 y 35.7 millones de parámetros, respectivamente.

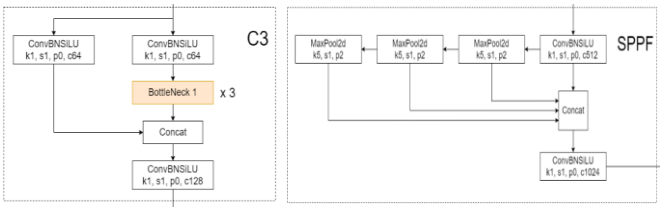
En la Fig. 5-a) se ilustra la arquitectura YOLOv5m-P6 que cuenta con 3 bloques importantes: La *Columna vertebral* que extrae las características de las escenas a diferentes escalas usando principalmente Cross Stage Partial Network (CSP-Darknet53) (Fig. 5-b) [50]. Esta columna vertebral está conectada al *cuello*, que es una pirámide de características basada en CSP-PANet [51] éste se encarga de fusionar las características extraídas para obtener tres tipos de características, cada una con una escala diferente. Estas características se pasan luego al *cabezal* del modelo para hacer las detecciones y clasificaciones de objetos pequeños, medianos y grandes. Cabe destacar que el cabezal utilizado en YOLO versión 3 también se emplea en este modelo [50], [52].

La notación de P6 hace referencia a que se aumenta una capa más de salida con saltos (stride) de 64 para objetos grandes, lo que permite aprovechar escenas con dimensiones más grandes a 640 pixeles. Esta adición presenta un mejor rendimiento en la métrica de evaluación de COCO AP [53].

[1] YOLO versión 5 cuenta con distintos lanzamientos, desde 1 hasta 7, referente a sus actualizaciones. Hasta el momento de nuestra aplicación se tomó el más actual, lanzamiento 6. Como se ilustra en su repositorio con la notación *'YOLOv5m6'*.



(a)



(b) (c)

Fig. 5 (a) Elaboración propia basado en [50], [54] de la arquitectura de YOLOv5m-P6. (b) Modulo C3-cross Stage Partial network. (c) Modulo SPPF - Spatial Pyramid Pooling Faster.

En la evaluación de los modelos de visión por computador, es importante establecer los criterios para determinar las detecciones verdaderas y falsas. A continuación, se definen las métricas de recall, precisión y f1-score, que son utilizadas para calcular el promedio de la precisión (mAP)[46]:

Verdadero positivo (VP): Es cuando el modelo detecta correctamente un objeto.
Falso Positivo (FP): Cuando se encuentra un objeto que no está realmente en la imagen.
Falso Negativo (FN): Es cuando no se detecta un objeto estando en la imagen.

El verdadero negativo se interpreta como todos los objetos correctamente no detectados: fondo de la imagen. Los verdaderos negativos no se usan porque dichas regiones no se etiquetan explícitamente.

Con base en las definiciones dadas, se pueden definir las métricas:

Precisión (P): Es una relación de las clasificaciones en verdaderos positivos (VP) y el número total de predicciones positivas, incluyendo los falsos positivos (FP). Ver (1).

$$P = \frac{VP}{VP + FP} \tag{1}$$

Recall (R): Es la relación de los VP y el total de verdaderos positivos y falsos negativos (FN). Ver (2).

$$R = \frac{VP}{VP + FN} \tag{2}$$

F1-score (F1): es una medida de la precisión de un modelo que se define como la media armónica que combina la precisión y el recall. Proporciona una medida equilibrada del rendimiento del modelo en términos de detección y precisión.

$$F1 = 2 * \frac{PxR}{P + R} \tag{3}$$

Finalmente, para medir el rendimiento del modelo de detección y clasificación de los objetos se usa el **Mean Average Precision (mAP)** que es igual al promedio del AP (average precision) de todas las clases, permitiendo esto comparar diferentes modelos en la misma tarea y diferentes versiones del mismo modelo; mAP está en un rango de 0 y 1 [55]

$$mAP@ \propto = \frac{\sum_{k=1}^{k=n} AP_k}{n} \tag{4}$$

n es el número de clases, $\propto$ es el IoU que se pretende usar para calcular el mAP y el AP_k es el average precision de cada clase.

D. Entrenamiento y selección del modelo.

La distribución del conjunto de escenas etiquetadas se ilustra en la Tabla I, particularmente, *el conjunto de prueba está construido sólo con las escenas del video de la Avenida Boyacá con avenida San José* (ver Fig. 2) para que el modelo sea probado con escenas que no haya visto durante su entrenamiento.

Tabla I

Conjunto de escenas	Número de escenas	Porcentaje
Entrenamiento	1108	71%
Validación	392	25%
Prueba	65	4%
Total	1565	100%

Inicialmente, para agilizar el proceso de entrenamiento y reducir el costo computacional, las dimensiones de las escenas se redujeron a 704x704 sin comprometer la calidad de estas.

Para seleccionar el modelo con el mejor rendimiento, se llevó a cabo el entrenamiento de los modelos *small* y *medium*, y se compararon sus resultados. Se realizaron experimentos utilizando diferentes valores de IoU (0.5, 0.6 y 0.7) y se seleccionó el valor que proporcionó el mejor rendimiento en términos del mAP.

Además, se realizó preprocesamientos de las escenas convirtiéndolas en escala de grises y aplicando algoritmos para la nitidez y mejorarlas, pero estos preprocesamientos no dieron ningún resultado favorable. Por tanto, se trabajó con las imágenes en su estado original (crudo) durante el entrenamiento de los modelos.

Los modelos se entrenaron en Google Colaboratory (Google Colab) [56], y las características de la plataforma fueron las

siguientes: una GPU (unidad de procesamiento gráfico) Tesla T4 con 15 GB de RAM, un disco duro de 78.2 GB, dos CPU Intel ® Xeon ® con 2.30 GHz cada uno, y una RAM del sistema de 12.7 GB, con una arquitectura x86_64. Cada modelo se entrenó a lo largo de 200 épocas, y todos los modelos entrenados mostraron un comportamiento muy similar al ilustrado en la Fig. 6. Esto se reflejó en un consumo máximo de memoria RAM en la GPU de 9 GB y en la memoria RAM del sistema de 7 GB. Por estas razones, se incluyen los comportamientos de dos modelos en esta gráfica.

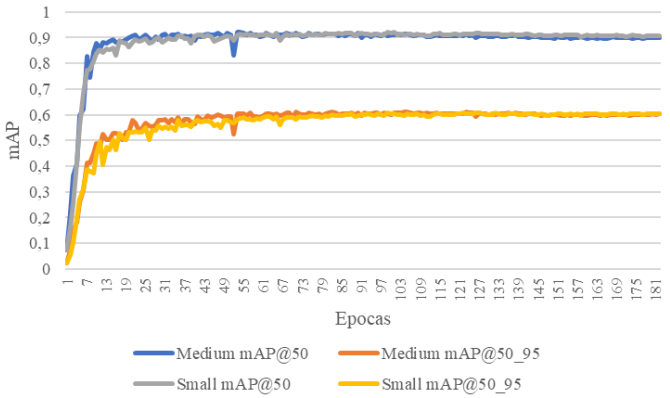


Fig. 6 Métrica mAP de los modelos *small* y *medium*.

En la Tabla II se presentan las métricas del mAP para cada modelo utilizando los conjuntos de datos de validación y prueba, mostrando que la variación del IoU no fue determinante para tener diferencias importantes entre los mAP's de los modelos. Esto se debe a que el modelo realiza una evaluación de1.000 valores de confianza y construye una curva F1 promedio de todas las clases para seleccionar el valor de confianza que da el máximo en esta curva promedio con el fin de ser usado las detecciones finales.

Con base en los resultados de la Tabla II se tomó el modelo *small* y *medium* con IoU igual a 0.5 y se decidió seleccionar el modelo que tuviese los mejores AP, pero conservando un balance entre las clases, es decir, los AP de las clases *camión* y *veh2ru* aumentarán.

Tabla II

Escenas de Validación									
Número				Modelo Small				Modelo Medium	
Escenas	IoU	P	R	mAP50	mAP50_95	P	R	mAP50	mAP50_95
392	50	0,881	0,886	0,917	0,608	0,886	0,875	0,915	0,609
392	60	0,889	0,879	0,917	0,608	0,883	0,876	0,915	0,609
392	70	0,884	0,875	0,914	0,609	0,883	0,872	0,913	0,610

Escenas de Prueba									
Número				Modelo Small				Modelo Medium	
Escenas	IoU	P	R	mAP50	mAP50_95	P	R	mAP50	mAP50_95
65	50	0,859	0,849	0,912	0,607	0,871	0,864	0,918	0,603
65	60	0,859	0,848	0,912	0,606	0,871	0,864	0,918	0,602
65	70	0,859	0,848	0,913	0,606	0,871	0,864	0,918	0,602

En la Tabla III se ilustra los AP de ambos modelos, y se observa que el modelo *medium* logró los aumentos en las clases mencionadas. Específicamente, se observa en la columna mAP@50, un aumento del 3.7% en el AP para la clase *camión* y un aumento del 4.9% para la clase *veh2ru*. Cabe destacar que estas dos clases fueron las de menos instancias en el conjunto de entrenamiento. Por lo tanto, con base en estos resultados se seleccionó el modelo ***medium*** como la mejor opción para llevar a cabo las detecciones y clasificaciones de los vehículos.

En la Fig. 7 se muestra la matriz de confusión del modelo seleccionado, *medium*, usando el conjunto de datos de prueba, donde se observa buenos resultados en las detecciones y clasificaciones de todas las clases, pues se evidencia errores por debajo del 6% entre éstas; la única clase que está generando el 51% falsos positivos es *Par_van*, y esta confusión tiende a darse con el fondo de la imagen, y posiblemente se deba por la composición de vehículos de esta clase, pues tiene múltiples colores, tamaños y formas (van, automóvil y camioneta).

Tabla III

			Modelo IoU con 0.5			
			Small		Medium	
Clase	No. escenas	Instancias	mAP50	mAP50_95	mAP50	mAP50_95
Bus	65	93	0,954	0,651	0,922	0,641
Camión	65	58	0,895	0,627	**0,932**	0,638
Par_van	65	179	0,928	0,623	0,937	0,635
Taxi	65	29	0,942	0,638	0,908	0,592
Veh2ru	65	56	0,843	0,495	**0,892**	0,509
Total	**65**	**415**	**0,912**	**0,607**	**0,918**	**0,603**

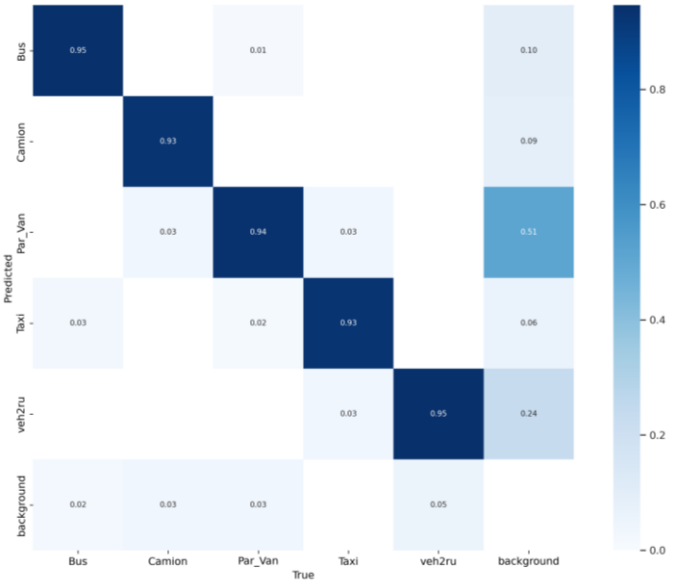


Fig. 7 Matriz de confusión del modelo *medium con IoU igual a 0.5*

E. Regiones de interés (RoI).

Para estimar la longitud de los vehículos en escenas 2D se contemplaron distintos métodos descritos en la sección II. Sin embargo, debido a la falta de información sobre los parámetros de las cámaras y altura de la cámara en la escena, entonces, se optó por trabajar desde la geometría proyectiva [26], [57], específicamente, con puntos de fuga [38] denotados de ahora en adelante como P.F. La proyección de la perspectiva en las escenas permite construir una aproximación del espacio tridimensional dentro de las detecciones 2D realizadas por el modelo entrenado, aprovechando así la velocidad de la arquitectura seleccionada.

El desarrollo de las estimaciones de longitud de los vehículos se trabajó únicamente con las 65 escenas del conjunto de prueba; y la *región de interés* [58] seleccionada fue la calzada más cercana a la cámara para hacer un cálculo aproximado de las distancias entre par de vehículos consecutivos.

Como se ilustra en la Fig. 8-a) se cuenta con una escena donde la calzada tiene una curvatura que llevó a identificar 3 P.F. utilizando como base las líneas blancas separadoreas de carril y los bordes de los andenes. Seguido a esto se tuvo que dividir la zona ROI en zona 5, 6 y 7 correspondientes a los 3 P.F.

Como se muestra en la Fig. 8-b) las rectas proyectadas desde lo puntos fuga 6 y 7 se intersecan, lo que permite conocer los límites de la perspectiva que brinda cada P.F y establecer, por ejemplo, los límites para la *zona 6*. Identificar estas intersecciones permite definir las zonas ROI como se muestra en la Fig. 8-c).

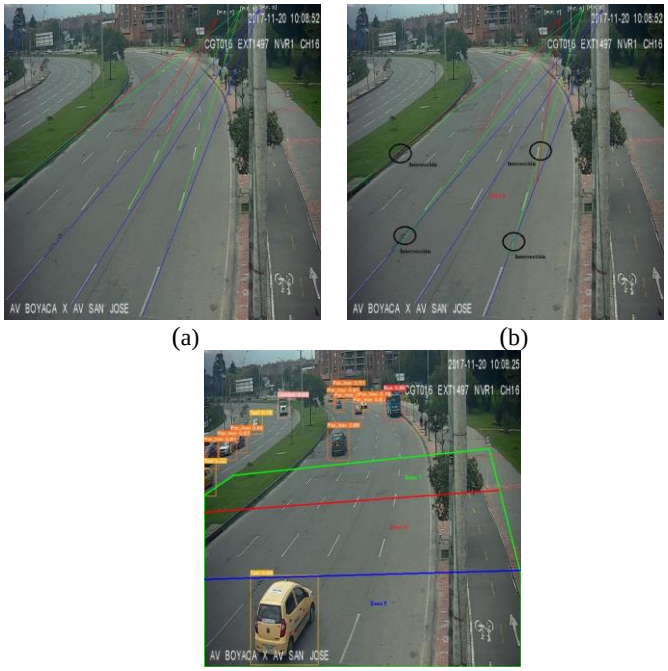


Fig. 8 Se ilustra la construcción de las regiones de interés -ROI. (a) rectas proyectadas desde las líneas separadoras de carril a los puntos de fuga. (b) Señalización de las intersecciones de las rectas proyectadas de los puntos de fuga que dan a conocer los límites de la zona 6. (c) Ilustración de las 3 zonas ROI definidas con base a los 3 puntos de fuga.

F. Estimación de la profundidad de los vehículos.

Como se cuenta con los puntos de fuga de la escena se procedió a realizar la estimación de la longitud del vehículo usando la perspectiva y las zonas definidas. Lográndose con esto pasar de las detecciones 2D a detecciones en 3D.

Las zonas 5, 6 y 7 tienen la finalidad de mantener la perspectiva y por ende dar un mejor ajuste de la detección en 3D sobre el vehículo.

Antes de dar a conocer la forma de construir los cubos de los objetos, se da la claridad, el *cuadro frontal* hace referencia a la parte delantera del vehículo y el *cuadro frontal proyectado* a la parte trasera del vehículo [26] , ver Fig. 9.

A continuación, se enumera el proceso metodológico que se empleó para lograr proyectar en 3D las detecciones de los vehículos. Ver Fig. 9.

1. Se identifica las coordenadas correspondientes al punto medio del cuadro delimitador obtenido en el modelo entrenado (cuadro color naranja). Estas coordenadas se denotan como (x_{med}, y_{med}).

2. Con el punto de fuga y el vértice *B* se traza una recta, usando la formula denotada en (5), que es denotada como *Y_B*. De manera similar, para el vértice *H* también se traza una recta denotada como *Y_H*.

$$y_i = X_j m_i + b_i \; ; \; i = A, B, C, H; j = 0{,}1, ..704 \qquad 5)$$

3. Se evalúa la coordenada x_{med} en la recta *Y_H* para obtener las coordenadas del vértice D y con esto se puede construir el cuadro frontal, vértices A, B, C y D.

4. Posteriormente con el punto de fuga y el vértice A y C se trazan las rectas *Y_A* y *Y_C*, respectivamente. Usadas para proyectar el cuadro frontal.

5. La forma de identificar las coordenadas de los vértices E y G se logró por medio de la intersección de la recta *Y_A* con el segmento H-T y la recta Y_C con el segmento H-Q.

6. Teniendo identificado los vértices E y G se evalúa la coordenada x_G en la recta *Y_B* para obtener las coordenadas del vértice F.

7. Al identificarse todos los vértices del cuadro frontal y del cuadro frontal proyectado dentro del cuadro delimitador (cuadro naranja) se procedió a unir los vértices para formar el cubo o detección en 3D del vehículo en escena como se muestra en la Fig. 9 y 10.

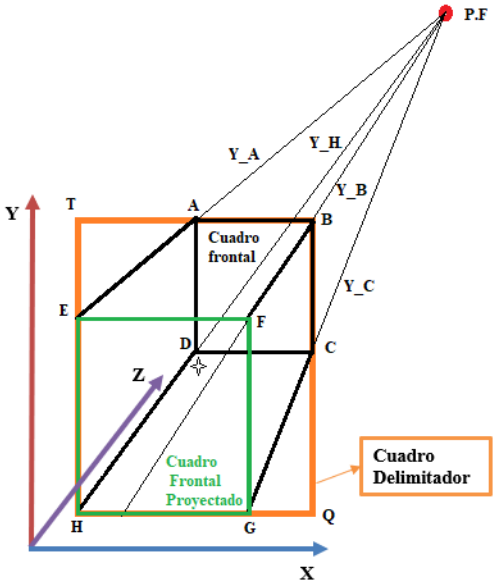


Fig. 9 Proyección del cuadro frontal con restricciones por el cuadro delimitador en 2D (cuadro naranja).

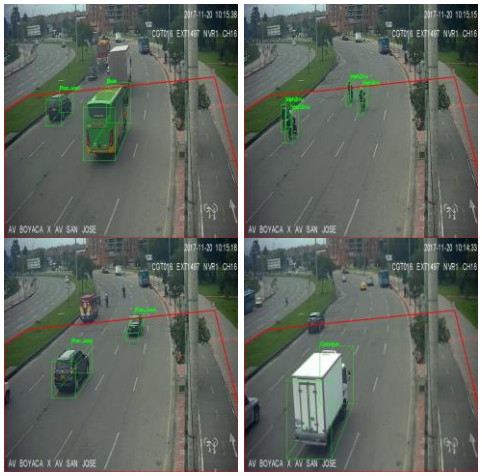


Fig. 10. Detecciones en 3D de los vehículos del conjunto de prueba.

G. Cálculo de distancia entre vehículos consecutivos.

Al tenerse las detecciones en 3D se puede tener mayor aproximación sobre la ubicación del vehículo en escena, lo que se traduce en calcular distancias entre vehículos consecutivos. *Para esto, se debe tener en cuenta que las escenas usadas para el proceso siguen siendo, únicamente, las del conjunto de prueba lo que significa que se debió seleccionar de las 65 escenas aquellas que contaran con vehículos consecutivos y que estuvieran dentro de la región de interés* (ver Fig. 8-c), pues fuera de estas zonas no se puede calcular las distancias. Por lo tanto, con base en estos criterios *únicamente 17 escenas fueron seleccionadas*.

Para calcular la distancia [59] entre vehículos se utilizó como referencia los vértices C (cuadro frontal del vehículo) y los vértices H o G de la parte trasera del vehículo que precede (Ver Fig. 11). También, *se definió usar como medida de distancia los pixeles* por falta información sobre la ubicación y parámetros de la cámara.

Fig. 11 Ilustración de la distancia que se calcula entre los dos vehículos usando los vértices C, H y G.

De las 17 escenas se calcularon en total 22 distancias de las cuales el promedio fue de 130 píxeles con una desviación estándar de 84 píxeles. En la Fig. 12-a) se observan los promedios de distancia entre par de vehículos, así, la clase Veh2ru (motos y bicicletas) son los que tienen la distancia promedio más baja, siendo de 28 pixeles, seguido por taxi que es de 35 pixeles. Contrario a esto, la distancia más amplia que se calculó fue de Par_Van (vehículos particulares y vans) que antecede una moto o bicicleta, siendo este promedio de 193 pixeles.

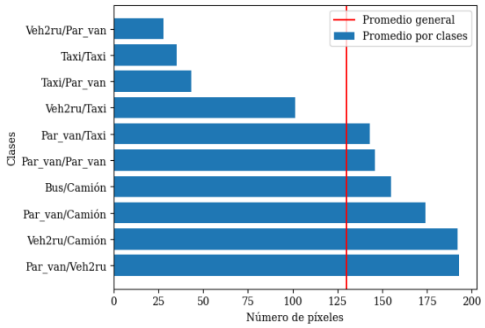
También se puede evidenciar que aquellos vehículos que tienen distancias superiores al promedio general, 130, son principalmente los que anteceden un camión, por ejemplo, Bus, Par_Van y Veh2ru.

Con base en las distancias que se obtuvieron y con el fin de brindar información que aporte a determinar si una distancia puede tener riesgo de colisión se determinó establecer 3 categorías con sus respectivas funciones de pertenencia ver Fig. 12-b).

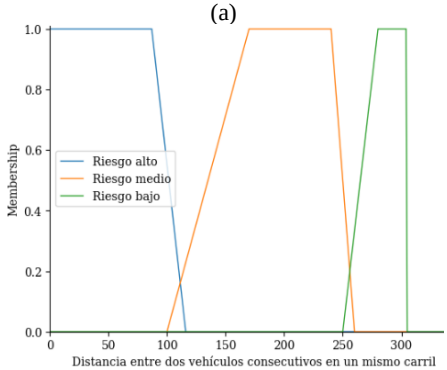
Con el fin de establecer las categorías de riesgo de colisión, se tomaron en cuenta diversos factores. En primer lugar, se promediaron las longitudes de varios vehículos similares a un Chevrolet Aveo de 4 puertas en las escenas, obteniendo un valor de 58 píxeles, que corresponde a una longitud real de 4.3 metros [60]. Además, con base en las normas del tránsito [59] los vehículos que circulan a velocidades entre 30 y 60 kilómetros por hora, la separación recomendada entre vehículos que viajan en el mismo carril es de 20 metros.

Por lo tanto, Con base en esta información y los resultados de las distancias promedio entre par de vehículos mostrados en la Fig. 12-a), se establecieron las siguientes reglas generales para determinar las categorías de riesgo:

Riesgo alto de colisión: vehículos con menos de 8.2 metros de separación entre sí.
Riesgo medio de colisión: vehículos con separación entre 8.2 y 19.97 metros.
Riesgo bajo de colisión: vehículos con separación mayor a 19.97 metros entre sí.



(a)



(b)

Fig. 12. a) Distancia Promedio en pixeles entre par de vehículos b) Funciones de pertenencia del riesgo de colisión.

Las funciones de pertenencia se plantearon trapezoidales y se hizo énfasis en que los grados de pertenencia sean altos para las funciones cuando se pasa a una categoría de mayor riesgo de colisión. Permitiendo esto poder alertar inmediatamente que se pasa a una categoría que tiene más posibilidad de riesgo, ver Fig.12-b).

La función de pertenencia expresada en función a trozos de pixeles se muestra en (5) y para realizar las inferencias se determinó usar el método de Mandami[61], [62].

$$\pi(x) = \begin{cases} Riesgo\ Alto\ ; 0 < x \leq 112 \\ Riesgo\ Medio\ ; 112\ < x \leq 257 \\ Riesgo\ Bajo\ ;\ 257 < x \leq 304 \end{cases} \quad (5)$$

En la ecuación (6) se muestra la formulación para calcular el grado de pertenencia en las funciones trapezoidales de la Fig. 12-b).

$$\pi(x) = \begin{cases} \frac{x-a}{b-a}\ ; a < x \leq b \\ 1\ ;\ \ b < x \leq c \\ \frac{d-x}{d-c}\ ; c < x \leq d \end{cases} \quad (6)$$

En la Fig. 13. se muestra la distribución de la clasificación del riesgo de colisión que presentaron las 22 distancias usando las respectivas funciones de pertenencia. Por consiguiente, el 45% de estas distancias están en riesgo alto de colisión con el vehículo que lo precede, seguido por el 50% tienen un riesgo medio de colisión y con un porcentaje muy bajo sólo el 4.5% está en riesgo bajo.

Los resultados obtenidos reflejan una realidad evidente en la cual los conductores no respetan en gran medida las distancias de seguridad recomendadas. Esta falta de cumplimiento de las normas de tránsito y las distancias adecuadas entre vehículos puede aumentar significativamente el riesgo de colisión en las vías. En consecuencia, las reglas y funciones de pertenencia utilizadas en este proyecto fueron ajustadas para reflejar esta situación y adaptarse a la realidad observada en las escenas de tráfico analizadas. De esta manera, se busca tener en cuenta las condiciones reales en las que se desenvuelven los actores viales en la ciudad y brindar una evaluación más precisa del nivel de riesgo de colisión.

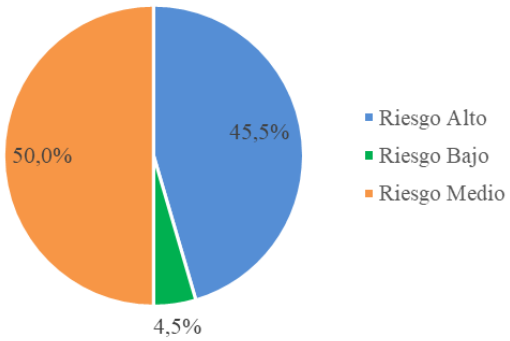


Fig. 13. Distribución del riesgo de colisión de las 22 distancias entre par de vehículos consecutivos.

Por último, una vez se han cumplido los objetivos establecidos, se presentan los tiempos de ejecución del modelo seleccionado, denominado *medium*. La Tabla IV ofrece una representación visual de estos tiempos. Es importante destacar que, al utilizar este modelo ya entrenado y probado, el tiempo requerido para realizar inferencias de cuadros delimitadores (CD) en 2D, convertir los CD de 2D a 3D y calcular distancias por instancia es de tan solo 0,152 milisegundos (ms).

Tabla IV

Etapas del proceso	Número de escenas	Número de instancias	ms	ms por instancia
Entrenamamiento/Validación	1.565	16.712	7,84E+06	469,24
Inferencias de CD en 2D	65	415	1,28E+04	30,84
Convertir CD de 2D a 3D	65	415	1,30E+04	31,27
Calculo de distancia	65	415	5,70E+02	1,37

V. Conclusiones y trabajos futuros.

1. El modelo seleccionado y finalmente entrenado presentó un mAP@50 de 91.8, lo cual es un resultado satisfactorio para el presente proyecto, pues, además se resalta el hecho de que este modelo logró un rendimiento suficientemente alto con sólo 1.108 escenas en su entrenamiento.
2. Con las detecciones 2D se logró a partir de puntos de fuga de la escena reconstruir con proyecciones geométricas las detecciones en 3D de los vehículos motorizados y no motorizados.
3. A medida que se iba avanzando en cada etapa del proyecto se fueron omitiendo escenas del conjunto de prueba. Sin embargo, las 17 escenas seleccionadas en la etapa final permitieron hacer los análisis correspondientes y mostrar que los conductores de vehículos automóviles, en especial motocicletas y bicicletas, tienen mayor posibilidad de riesgo de colisión. Pues su estadística dio que el 95.5% de las distancias de separación entre par de vehículos consecutivos estuvieron entre riesgo alto y medio.
4. Es importante destacar que una de las finalidades de este proyecto es generar conciencia sobre la importancia de mantener distancias de seguridad adecuadas y fomentar el cumplimiento de las normas de tránsito para prevenir accidentes y mejorar la seguridad vial en la ciudad de Bogotá.
5. La geometría proyectiva y el uso de puntos de fuga pueden ser una estrategia efectiva para estimar la longitud y distancia entre vehículos en escenas 2D, lo cual contribuye a evaluar el riesgo de colisión de manera aproximada.
6. Se identificó un limitante sobre la perspectiva y es que los objetos alejados de la cámara no se pueden tomar para las mediciones de distancia espacial, pues su precisión se verá afectada también por la oclusión.
7. A pesar de los buenos resultados del modelo entrenado, poder aumentar el conjunto de datos sería

adecuado para mejorar aún más el rendimiento de este modelo YOLOv5m-P6.

8. Respecto a los posibles trabajos futuros se debería adaptar un método de detección de puntos de fuga eficiente, podría ser el descrito en la referencia [38] con el fin de tener detecciones más ajustadas con base en la perspectiva.
9. En consecuencia, con el punto anterior, poder usar más vídeos en el conjunto de prueba y en el cálculo de las distancias con el fin de evaluar su funcionamiento y mejorar lo propuesto en este proyecto.
10. Es importante poder medir el ajuste de las detecciones en 3D al utilizar los puntos de fuga como partida, lo cual se propone a futuro poder implementar métricas para medir las detecciones en tercera dimensión y, con esto brindar mayor confianza en los resultados con estas métricas.
11. Desde una perspectiva práctica, este modelo puede aplicarse en diversas zonas de la ciudad de Bogotá con el propósito de recopilar datos que contribuyan a mejorar tanto la gestión del tráfico como la seguridad vial. Por lo tanto, es posible utilizarlo para la monitorización en tiempo real del flujo vehicular y para la obtención de estadísticas de tráfico que respalden la planificación urbana. Estas estadísticas, basadas en la densidad vehicular y distancias entre los vehículos, pueden servir como fundamento para la toma de decisiones relacionadas con la expansión de carreteras, señalización de velocidad, construcción de nuevos puentes y otras iniciativas urbanísticas.
12. Consecuente con el punto anterior, es importante tener en cuenta que idealmente para lograr obtener distancias entre vehículos se debe aplicar el modelo construido sobre carreteras con flujo vehicular medio o bajo, pues en flujos altos se tendrán vehículos extremadamente cerca desde la perspectiva y puede fallar las estimaciones que se realizan.

REFERENCES

[1] Agencia Nacional de Seguridad Vial, “Anuario Nacional de Siniestralidad Vial - Colombia 2.021,” May 2022, [Online]. Available: https://ansv.gov.co/sites/default/files/2022-07/Anuario_Nacional__2021_Vfinal.pdf

[2] L. L. Lacatan, R. S. Santos, J. W. Pinkihan, R. Y. Vicente, and R. S. Tamargo, “Brake-Vision: A Machine Vision-Based Inference Approach of Vehicle Braking Detection for Collision Warning Oriented System,” *Proceedings of 2nd IEEE International Conference on Computational Intelligence and Knowledge Economy, ICCIKE 2021*, pp. 485–488, 2021, doi: 10.1109/ICCIKE51210.2021.9410750.

[3] M. S. Almutairi, K. Almutairi, and H. Chiroma, “Hybrid of deep recurrent network and long short term memory for rear-end collision detection in fog based internet of vehicles,” *Expert Syst Appl*, vol. 213, no. PC, p. 119033, 2023, doi: 10.1016/j.eswa.2022.119033.

[4] J. S. Kim, D. H. Lee, D. W. Kim, H. Park, K. J. Paik, and S. Kim, “A numerical and experimental study on the obstacle collision avoidance system using a 2D LiDAR sensor for an autonomous surface vehicle,” *Ocean Engineering*, vol. 257, no. March, 2022, doi: 10.1016/j.oceaneng.2022.111508.

[5] T. Roddick, A. Kendall, and R. Cipolla, “Orthographic Feature Transform for Monocular 3D Object Detection,” *30th British Machine Vision Conference 2019, BMVC 2019*, Nov. 2018, [Online]. Available: http://arxiv.org/abs/1811.08188

[6] S. Tak, S. Kim, and H. Yeo, “Development of a Deceleration-Based Surrogate Safety Measure for Rear-End Collision Risk,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2435–2445, 2015, doi: 10.1109/TITS.2015.2409374.

[7] U. Hwhfwlrq, D. Q. G. Qdo, I. R. U. Fodvvli, L. Q. J. Wkh, and D. Lpdjhv, “Post-Crash detection and Traffic Analysis,” pp. 1092–1097, 2021.

[8] Krishna, M. Poddar, M. K. Giridhar, A. S. Prabhu, and V. Umadevi, “Automated traffic monitoring system using computer vision,” *Proceedings of 2016 International Conference on ICT in Business, Industry, and Government, ICTBIG 2016*, 2017, doi: 10.1109/ICTBIG.2016.7892717.

[9] N. Ben Romdhane, H. Mliki, R. El Beji, and M. Hammami, “Combined 2d/3d traffic signs recognition and distance estimation,” *IEEE Intelligent Vehicles Symposium, Proceedings*, vol. 2016-Augus, no. Iv, pp. 355–360, 2016, doi: 10.1109/IVS.2016.7535410.

[10] F. Wang, Z. Wang, X. Li, X. Yao, W. Hu, and T. Fu, “Improved Time-to-collision Considering Vehicle Speed Adaptation based on Trajectory Data,” *6th International Conference on Transportation Information and Safety: New Infrastructure Construction for Better Transportation, ICTIS 2021*, pp. 1432–1437, 2021, doi: 10.1109/ICTIS54573.2021.9798583.

[11] D. Roy, T. Ishizaka, C. K. Mohan, and A. Fukuda, “Detection of Collision-Prone Vehicle Behavior at Intersections Using Siamese Interaction LSTM,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 3137–3147, 2022, doi: 10.1109/TITS.2020.3031984.

[12] S. Agrawal and S. W. Varade, “Collision detection and avoidance system for vehicle,” *Proceedings of the 2nd International Conference on Communication and Electronics Systems, ICCES 2017*, vol. 2018-Janua, no. Icces, pp. 476–477, 2018, doi: 10.1109/CESYS.2017.8321325.

[13] S. M. Mansoor Roomi, C. Ramkumar, and K. Mutharasi, “Deep Learnt Anti Collision System for Highway Safety,” *2019 5th International Conference on Advanced Computing and Communication Systems, ICACCS 2019*, pp. 672–674, 2019, doi: 10.1109/ICACCS.2019.8728406.

[14] E. Shreyas and M. H. Sheth, “3D Object Detection and Tracking Methods using Deep Learning for Computer Vision Applications,” pp. 735–738, 2021.

[15] OMS, “Political Declaration of the High-Level Meeting on Improving Global Road Safaty.,” Jul. 2022. [Online]. Available: https://www.who.int/es/news/item/30-06-2022-new-political-declaration-to-halve-road-traffic-deaths-and-injuries-by-2030-is-a-milestone-achievement

[16] D. Trucks, “A Review of Applications of Artificial Intelligence in Heavy Duty Trucks,” pp. 1–20, 2022.

[17] D. K. Dewangan and S. P. Sahu, “Towards the design of vision-based intelligent vehicle system: methodologies and challenges,” *Evol Intell*, no. 0123456789, 2022, doi: 10.1007/s12065-022-00713-2.

[18] R. Teng, “Prevention Detection for Cyclists based on Faster R-CNN,” pp. 142–148, 2022, doi: 10.1109/cncit56797.2022.00030.

[19] C. Fernandez-Labrador, J. M. Facil, A. Perez-Yus, C. Demonceaux, J. Civera, and J. J. Guerrero, “Corners for Layout: End-to-End Layout Recovery from 360 Images,” Mar. 2019, [Online]. Available: http://arxiv.org/abs/1903.08094

[20] S. Pathak, A. Moro, A. Yamashita, and H. Asama, “Optical Flow-Based Epipolar Estimation of Spherical Image Pairs for 3D Reconstruction,” *SICE journal of control, measurement, and system integration*, vol. 10, no. 5, pp. 476–485, Sep. 2017, doi: 10.9746/JCMSI.10.476.

[21] IEEE Circuits and Systems Society and Institute of Electrical and Electronics Engineers, “Stacked Omnistereo for Virtual Reality with Six Degrees of Freedom,” 2009.

[22] F. Munir, S. Azam, and M. Jeon, “SSTN: Self-Supervised Domain Adaptation Thermal Object Detection for Autonomous Driving,” in *IEEE International Conference on Intelligent Robots and Systems*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 206–213. doi: 10.1109/IROS51168.2021.9636353.

[23] X. Ma, Z. Wang, H. Li, P. Zhang, W. Ouyang, and X. Fan, “Accurate monocular 3D object detection via color-embedded 3D reconstruction for autonomous driving,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-Octob, no. Iccv, pp. 6850–6859, 2019, doi: 10.1109/ICCV.2019.00695.

[24] V. S. Sindhu, “Identificación de vehículos a partir de video de tráfico Vigilancia usando YOLOv4,” no. Iciccs 2021, pp. 1768–1775, 2022.

[25] M. Drobnitzky, J. Friederich, B. Egger, and P. Zschech, “Survey and Systematization of 3D Object Detection Models and Methods,” pp. 1–32, 2022, [Online]. Available: http://arxiv.org/abs/2201.09354

[26] A. M. Andrew, *Multiple View Geometry in Computer Vision*, vol. 30, no. 9–10. 2001. doi: 10.1108/k.2001.30.9_10.1333.2.

[27] Q. Lian, B. Ye, R. Xu, W. Yao, and T. Zhang, “Exploring Geometric Consistency for Monocular 3D Object Detection,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2022-June, pp. 1675–1684, 2022, doi: 10.1109/CVPR52688.2022.00173.

[28] T. L. T. da Silveira, P. G. L. Pinto, J. Murrugarra-Llerena, and C. R. Jung, “3D Scene Geometry Estimation from 360° Imagery: A Survey,” *ACM Comput Surv*, Mar. 2022, doi: 10.1145/3519021.

[29] X. Shi, Q. Ye, X. Chen, C. Chen, Z. Chen, and T. K. Kim, “Geometry-based Distance Decomposition for Monocular 3D Object Detection,” *Proceedings of the IEEE International Conference on Computer Vision*, pp. 15152–15161, 2021, doi: 10.1109/ICCV48922.2021.01489.

[30] S. Shi, X. Wang, and H. Li, “PointRCNN: 3D object proposal generation and detection from point cloud,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 770–779, 2019, doi: 10.1109/CVPR.2019.00086.

[31] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, “Frustum PointNets for 3D Object Detection from RGB-D Data,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 918–927, 2018, doi: 10.1109/CVPR.2018.00102.

[32] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep learning on point sets for 3D classification and segmentation,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 77–85, 2017, doi: 10.1109/CVPR.2017.16.

[33] M. Drobnitzky, J. Friederich, B. Egger, and P. Zschech, “Survey and Systematization of 3D Object Detection Models and Methods,” Jan. 2022, [Online]. Available: http://arxiv.org/abs/2201.09354

[34] M. Rezaei, M. Azarmi, F. Mohammad, and P. Mir, “Traffic-Net: 3D Traffic Monitoring Using a Single Camera,” 2021.

[35] Z. Wang, Q. Xie, M. Wei, K. Long, and J. Wang, “Multi-feature Fusion VoteNet for 3D Object Detection,” *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 18, no. 1, pp. 1–17, Jan. 2022, doi: 10.1145/3462219.

[36] A. Naiden, V. Paunescu, G. Kim, B. Jeon, and M. Leordeanu, “Shift R-CNN: Deep Monocular 3D Object Detection with Closed-Form Geometric Constraints,” *Proceedings - International Conference on Image Processing, ICIP*, vol. 2019-Septe, pp. 61–65, 2019, doi: 10.1109/ICIP.2019.8803397.

[37] M. Ding *et al.*, “Learning depth-guided convolutions for monocular 3d object detection,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, 2020, pp. 11669–11678. doi: 10.1109/CVPR42600.2020.01169.

[38] Q. Yang *et al.*, “A fast vanishing point detection method based on row space features suitable for real driving scenarios,” *Sci Rep*, vol. 13, no. 1, pp. 1–11, 2023, doi: 10.1038/s41598-023-30152-7.

[39] Y. B. Liu, M. Zeng, and Q. H. Meng, "Unstructured Road Vanishing Point Detection Using Convolutional Neural Networks and Heatmap Regression," *IEEE Trans Instrum Meas*, vol. 70, pp. 1–8, 2021, doi: 10.1109/TIM.2020.3019863.

[40] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[41] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3D Bounding Box Estimation Using Deep Learning and Geometry," Dec. 2016, [Online]. Available: http://arxiv.org/abs/1612.00496

[42] J. Chen, B. Lei, Q. Song, H. Ying, D. Z. Chen, and J. Wu, "A Hierarchical Graph Network for 3D Object Detection on Point Clouds," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 389–398, 2020, doi: 10.1109/CVPR42600.2020.00047.

[43] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, pp. 2980–2988, 2017, doi: 10.1109/ICCV.2017.322.

[44] S. Song and J. Xiao, "Deep sliding shapes for amodal 3D object detection in RGB-D images," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 808–816, 2016, doi: 10.1109/CVPR.2016.94.

[45] A. Kathuria, "What's new in YOLO v3?," Towards Data Science. Accessed: Jun. 24, 2023. [Online]. Available: https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b

[46] RoboFlow, "Roboflow: Give your software the power to see objects in images and video," January. Accessed: Nov. 24, 2022. [Online]. Available: https://roboflow.com/

[47] Z. Luo *et al.*, "MIO-TCD: A New Benchmark Dataset for Vehicle Classification and Localization," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 5129–5141, Oct. 2018, doi: 10.1109/TIP.2018.2848705.

[48] S. L. Chandrakant Patel Pinal Salot, "Survey on Different Object Detection and Segmentation Methods," 2021. [Online]. Available: www.ijisrt.com

[49] J. Terven and D. Cordova-Esparza, "A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond," pp. 1–27, 2023, [Online]. Available: http://arxiv.org/abs/2304.00501

[50] J. Glenn, "YOLOv5 (6.0/6.1) brief summary: 1. Model Structure." Accessed: Jun. 03, 2023. [Online]. Available: https://github.com/ultralytics/yolov5/issues/6998#:~:text=4.3 Build Targets-,1. Model Structure,-YOLOv5 (v6.0/6.1

[51] I. S. Gillani *et al.*, "Yolov5, Yolo-x, Yolo-r, Yolov7 Performance Comparison: A Survey," no. Figure 1, pp. 17–28, 2022, doi: 10.5121/csit.2022.121602.

[52] Z. Wang, L. Jin, S. Wang, and H. Xu, "Postharvest Biology and Technology Apple stem / calyx real-time recognition using YOLO-v5 algorithm for fruit automatic loading system," vol. 185, no. December 2021, 2022.

[53] J. Glenn, E. Ayush, and I. Wilson, "v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations," Release v5.0. Accessed: Jun. 27, 2023. [Online]. Available: https://github.com/ultralytics/yolov5/releases/tag/v5.0#:~:text=v5.0%20%2D%20YOLOv5%2DP6%201280%20models%2C%20AWS%2C%20Supervise.ly%20and%20YouTube%20integrations

[54] J. Dabai, "深入浅出Yolo系列之Yolov3&Yolov4&Yolov5&Yolox核心基础知识完整讲解 - 知乎." Accessed: Jun. 08, 2023. [Online]. Available: https://zhuanlan.zhihu.com/p/143747206

[55] P. K. Yadav *et al.*, "Assessing the performance of YOLOv5 algorithm for detecting volunteer cotton plants in corn fields at three different growth stages," *Artificial Intelligence in Agriculture*, vol. 6, pp. 292–303, 2022, doi: 10.1016/j.aiia.2022.11.005.

[56] "Te damos la bienvenida a Colaboratory - Colaboratory." Accessed: Oct. 02, 2023. [Online]. Available: https://colab.research.google.com/?hl=es

[57] D. Mery, *Visión por Computador*. 2004. Accessed: Apr. 25, 2023. [Online]. Available: http://dmery.sitios.ing.uc.cl/Prints/Books/2004-ApuntesVision.pdf

[58] J. A. Pacheco Sánchez, "Sistema De Reconocimiento De Objetos Removidos De Una Escena, Utilizando Visión Por Computador," Pontificia Universidad Javeriana, Bogotá, 2011. [Online]. Available: https://repository.javeriana.edu.co/bitstream/handle/10554/7073/tesis535.pdf?sequence=1

[59] Secretaría de Tránsito y Transporte de Bogotá., *CÓDIGO NACIONAL DE TRÁNSITO TERRESTRE*. Bogotá D.C.: Alcaldía Mayor de Bógota, 2002.

[60] Administrador, "Automotores Online." Accessed: Jun. 11, 2023. [Online]. Available: https://automotoresonline.blogspot.com/2011/03/

[61] F. Palacios-pereira and M. E. Ayala-silva, "Design of a pitch angle control system in wind turbines through an adaptive pi control by fuzzy logic," vol. 10, pp. 91–114, 2023.

[62] O. Camacho, E. Iglesias, M. Herrera, and H. Aboukheir, "Fuzzy logic-based control : From fundamentals to applications," vol. 4, no. 2, pp. 6–37, 2021.