



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

VISIÓN ELECTRÓNICA

Algo más que un estado sólido

<https://doi.org/10.14483/issn.2248-4728>



Efficient live video traffic detection with YOLO convolutional neural architectures

Detección eficiente de tráfico de video en vivo con arquitecturas neuronales convolucionales YOLO

Ricardo Alfonso Gómez Suárez¹, Julieth Alejandra Sandoval Estupiñán², Paola Andrea Mateus Abaunza³

Abstract

Advancements in computer vision have revolutionized machines' ability to recognize and classify objects in real-time. This study explored the convolutional neural network architectures YOLOv5 and YOLOv8 for precise and efficient real-time object identification. A custom dataset was created, and Python and OpenCV were utilized for the annotation and training processes of the models. The results demonstrated the high accuracy and detection speed of both models, even in live video environments.

Palabras clave: BNN, CNN, Computer Vision, OpenCV, Python, YOLO

Resumen

¹ BSc. In Electronic Engineering, Specialist in Engineering Management Systems, from the Pontificia Universidad Javeriana. Mobile Communications Specialist, Master in Information and Communications Sciences, student Doctorate in Engineering from the Universidad Distrital Francisco José de Caldas, Bogotá, Colombia. Current position: Professor Researcher, ECCI University, Bogotá, Colombia. Email: rgomez@ecc.edu.co ORCID: <https://orcid.org/0000-0002-1142-9406>

² BSc. In industrial electronics technology, student of Electronic Engineering at the ECCI University, Bogotá, Colombia. Email: julietha.sandoval@ecc.edu.co ORCID: <https://orcid.org/0009-0005-0022-1668>

³ BSc. In Electronic Engineering, MSc. in Electronic Engineering, Pontificia Universidad Javeriana, Bogotá, Colombia. Current Position: Professor Researcher, Universidad Nacional Abierta y a Distancia, UNAD, Bogotá, Colombia. Email: paola.mateus@unad.edu.co ORCID: <https://orcid.org/0000-0002-0638-1815>

Los avances en el campo de la visión artificial han revolucionado la capacidad de las máquinas para llevar a cabo la identificación y clasificación de objetos en tiempo real. En este estudio, se examinaron las arquitecturas de redes neuronales convolucionales YOLOv5 y YOLOv8 con el propósito de lograr una identificación precisa y eficiente de objetos en escenarios de tiempo real. Para este propósito, se creó un conjunto de datos personalizado, y se emplearon Python y OpenCV para llevar a cabo la anotación y el entrenamiento de los modelos. Los resultados evidenciaron la destacada precisión y velocidad de detección de ambos modelos, inclusive en situaciones de transmisión de vídeo en directo.

Palabras clave: BNN, CNN, Visión por Computador, OpenCV, Python, YOLO.

1. Introducción

En este estudio, se llevaron a cabo análisis exhaustivos de dos arquitecturas de vanguardia, YOLOv5 y YOLOv8, con el objetivo de alcanzar una identificación de objetos en tiempo real. Este enfoque se basó en el uso de conjuntos de datos personalizados, respaldado por las potentes herramientas de Python y OpenCV [1]. Ambos modelos exhibieron un desempeño excepcional al detectar objetos, incluso en entornos desafiantes, lo que los posiciona como herramientas altamente eficaces para aplicaciones que exigen respuestas instantáneas [2]. Estos logros no solo aportan al conocimiento práctico, sino que también desencadenan una serie de nuevas posibilidades en diversas industrias, impulsando avances tecnológicos como su aplicación en camiones autónomos y la automatización industrial, lo cual, a su vez, eleva los estándares de eficiencia y seguridad [3].

2. Marco conceptual

2.1. Aprendizaje automático y redes neuronales

El campo del aprendizaje automático se enfoca en desarrollar algoritmos y modelos que toman decisiones basadas en datos, posibilitando que las máquinas aprendan y mejoren su

rendimiento a través de la experiencia. En el ámbito de la detección de objetos en tiempo real, el aprendizaje automático desempeña un papel esencial al entrenar modelos para reconocer y categorizar objetos utilizando conjuntos de datos que han sido previamente etiquetados [4].

Las redes neuronales, inspiradas por el funcionamiento del cerebro humano, son ampliamente utilizadas en el aprendizaje automático. Entre ellas, las redes neuronales convolucionales (CNN) son especialmente efectivas en tareas relacionadas con la visión por computadora, como la detección y clasificación de objetos en imágenes y videos [5].

Dos ejemplos notables de arquitecturas de CNN para la detección de objetos en tiempo real son YOLOv5 y YOLOv8. Estos modelos adquieren conocimientos visuales para identificar y ubicar objetos de manera rápida y precisa [6].

El marco conceptual de este estudio involucra la aplicación del aprendizaje automático y las CNN para abordar los desafíos asociados con la detección de objetos en tiempo real [7]. El aprendizaje automático permite a las máquinas adquirir información a partir de datos, mientras que las CNN, como YOLOv5 y YOLOv8, funcionan como herramientas específicas para la detección y clasificación de objetos. La combinación de estas tecnologías posibilita una identificación rápida y precisa de objetos en aplicaciones en tiempo real, con implicaciones significativas en áreas como la conducción autónoma, la seguridad y la vigilancia [8].

2.2. Gobernanza y tecnologías emergentes

Un punto de suma importancia que demanda especial atención es el posible impacto en la equidad y la justicia. Es esencial asegurar que los sistemas de identificación de actores en carretera en tiempo real no perpetúen inadvertidamente prejuicios o discriminación hacia ciertos grupos de usuarios viales [9]. Los sesgos no intencionados presentes en estos sistemas podrían tener consecuencias severas, generando un trato desigual y resultados injustos. Además, aunque los avances tecnológicos posibilitan la recopilación masiva de datos, es de

vital importancia dar prioridad a la protección de los datos personales y cumplir con las normativas de privacidad al recolectar y procesar información visual. Alcanzar un equilibrio adecuado entre la utilización de los datos recabados para mejorar la seguridad vial y garantizar los derechos de privacidad de las personas constituye una tarea compleja que requiere una formulación y ejecución de políticas extremadamente cautelosas [9].

2.3. Medio ambiente y movilidad

La aplicación de YOLOv5 y YOLOv8 en la detección de objetos en tiempo real tiene el potencial de mejorar tanto la seguridad vial como la gestión del tráfico. No obstante, es fundamental evaluar y abordar las implicaciones ambientales y de movilidad que conlleva esta tecnología [10]. La utilización de cámaras y sensores puede influir en la biodiversidad, por lo que resulta esencial minimizar los efectos adversos en los entornos locales. En este sentido, es prioritario centrarse en la eficiencia energética y el empleo de fuentes de energía renovable [10].

En lo que respecta a la movilidad, la detección de objetos en tiempo real proporciona información valiosa para mejorar la planificación del transporte y fomentar opciones de movilidad sostenible [11]. La optimización de los semáforos y la mejora de los sistemas de transporte público pueden contribuir a reducir la congestión vehicular y las emisiones de gases de efecto invernadero generadas por los camiones particulares [11]. Además, esta tecnología promueve modalidades de transporte sostenibles, como el carpooling y la movilidad activa, al ofrecer datos precisos sobre los servicios disponibles, lo que resulta en una disminución de las emisiones y un uso más eficiente de los recursos [10].

Para garantizar un uso responsable de esta tecnología, las políticas deben abordar cuestiones éticas, de protección de datos y ambientales a lo largo de todo el ciclo de vida de estos sistemas [10]. La colaboración entre los sectores público, privado y académico se convierte en

un componente esencial para encontrar soluciones efectivas que impulsen la sostenibilidad y la movilidad responsable [11]. En resumen, la implementación responsable y la adopción de políticas sólidas desempeñan un papel crucial para maximizar los beneficios de esta tecnología al mismo tiempo que se minimizan los impactos negativos en el medio ambiente y la movilidad urbana.

2.4. Educación y capacitación en tecnologías de detección de objetos.

La educación y la capacitación desempeñan un papel esencial en el logro exitoso de la implementación de tecnologías de detección de objetos, como YOLOv5 y YOLOv8. Es crucial establecer una base sólida en los conceptos fundamentales del aprendizaje automático y las redes neuronales convolucionales [12-13]. Adquirir conocimientos específicos sobre las características, ventajas, limitaciones y configuraciones adecuadas de estos modelos se torna esencial para una variedad de aplicaciones [13]. En el proceso formativo, es necesario incorporar ejemplos prácticos y casos de uso que permitan la aplicación efectiva en situaciones del mundo real [14].

Asimismo, es igualmente importante fomentar la conciencia acerca de los desafíos éticos vinculados a la privacidad, la discriminación y los sesgos presentes en los modelos de aprendizaje automático [14]. Los participantes deben estar preparados para abordar de manera responsable estas cuestiones éticas y legales en el contexto de aplicaciones prácticas. En última instancia, la educación y la formación capacitan a estudiantes y profesionales para emplear las tecnologías de detección de objetos de manera eficiente y ética [14].

2.5. Bogotá como ciudad innovadora en la identificación de actores viales

Bogotá, la capital de Colombia, se ha convertido en pionera en la adopción de tecnologías de identificación de actores viales, incluidas YOLOv5 y YOLOv8. Al implementar sistemas avanzados de reconocimiento de objetos, la ciudad ha mejorado significativamente la

seguridad vial, la eficiencia del transporte y la gestión del tráfico [15]. Cámaras y sensores inteligentes colocados estratégicamente en toda Bogotá permiten una vigilancia eficiente del tráfico y la detección en tiempo real de camiones, peatones y bicicletas, proporcionando datos cruciales para la toma de decisiones informadas en materia de movilidad y seguridad vial [16]. Además, Bogotá ha estado a la vanguardia en la implementación de políticas de movilidad sostenible basadas en la identificación de actores viales. Aprovechando la información de los sistemas de reconocimiento de objetos, la ciudad ha implementado medidas para mejorar el flujo de tráfico, optimizar los tiempos de los semáforos y promover el uso de opciones de transporte ecológicas como el transporte público y la movilidad activa. Estas iniciativas han reducido efectivamente la congestión, reducido las emisiones contaminantes y fomentado una red de transporte más eficiente y sostenible [17-18].

3. Descripción del modelo

3.1. Flujo de tráfico

El enfoque de detección de objetos, basado en las arquitecturas YOLOv5 y YOLOv8, se dedica a analizar el flujo vehicular en tiempo real, haciendo uso de técnicas de visión por computadora y aprendizaje automático. Su tarea principal consiste en identificar, clasificar y rastrear diversos tipos de camiones en movimiento en las vías [19]. El estudio del flujo de tráfico reviste una importancia crucial en la comprensión de la congestión vial, la optimización de la gestión del transporte y la mejora de la planificación urbana. La eficiencia con la que este modelo aborda estas tareas se fundamenta en su arquitectura de red neuronal convolucional profunda [20].

Inicialmente, una capa de detección se encarga de reconocer camiones en imágenes o secuencias de imágenes mediante la utilización de mapas de características para identificar aspectos distintivos de los camiones, como sus formas, bordes y texturas. Posteriormente, los algoritmos de detección de objetos localizan y marcan cada vehículo en la escena [20]. Una

vez detectados, una capa de clasificación asigna etiquetas a cada vehículo, diferenciándolos entre automóviles, camiones, motocicletas, bicicletas y otros, lo que proporciona información detallada acerca del flujo de tráfico y facilita análisis más específicos [21].

Además, el modelo incorpora una capa de seguimiento que monitoriza la trayectoria de los camiones a lo largo del tiempo, lo que permite analizar el comportamiento del flujo vehicular, incluyendo aspectos como la velocidad, la dirección, los patrones de movimiento y accidentalidad, figura 1. Este seguimiento preciso asegura que el recuento del flujo de tráfico se mantenga actualizado y fiable en una zona determinada [22-23]. Para mejorar la precisión y robustez del modelo, se lleva a cabo un proceso de entrenamiento utilizando conjuntos de datos etiquetados que contienen imágenes de camiones y anotaciones que indican su ubicación y clase. Mediante algoritmos de aprendizaje automático, se optimizan los pesos de la red neuronal convolucional con el fin de lograr detecciones y clasificaciones más precisas en una variedad de condiciones y entornos de iluminación [2].

Figura 1. Accidentalidad en Colombia en 2020 y 2021. [11]

| Actor vial | | | | |
|----------------------|-------------|-------------|-------------|----------------|
| Condición | 2020 | 2021 | Variación | % V |
| Usuario de moto | 2908 | 4312 | 1404 | 48,28 % |
| Peatón | 1128 | 1566 | 438 | 38,83 % |
| Usuario de vehículo | 624 | 852 | 228 | 36,54 % |
| Usuario de bicicleta | 433 | 471 | 38 | 8,78 % |
| Usuario otros | 22 | 13 | -9 | -40,91 % |
| Sin Información | 343 | 56 | -287 | -83,67 % |
| Total | 5458 | 7270 | 1812 | 33,20 % |

3.2. Implementación de YOLO en Python: Códigos y uso del modelo YOLO para la identificación de actores viales.

La implementación del modelo YOLO (You Only Look Once) en Python se ha destacado como una herramienta sumamente eficaz para la identificación precisa y eficiente de los actores en las vías en tiempo real. YOLO se basa en una arquitectura de red neuronal convolucional que posibilita la detección y clasificación simultánea de múltiples objetos en una imagen [1].

Para utilizar el modelo YOLO en Python, es necesario realizar la instalación de las bibliotecas y dependencias requeridas, como OpenCV y PyTorch. Una vez que estas estén configuradas, se pueden seguir los siguientes pasos para su implementación:

1. Descargue y configura el modelo: en primer lugar, se debe proceder a la descarga y configuración del modelo YOLO previamente entrenado. Versiones como YOLOv3, YOLOv4, YOLOv5 o YOLOv8 están disponibles en repositorios de acceso público. Una vez descargado el modelo, se procede a su carga utilizando la biblioteca PyTorch [1].
2. Preparación de datos: antes de llevar a cabo la detección de los actores en las vías, se requiere efectuar una preparación de los datos de entrada. Esto implica obtener imágenes o acceder a una transmisión de video en tiempo real. Las imágenes o los fotogramas de video deben someterse a un proceso de redimensionamiento y normalización, asegurando que se adecuen a los requisitos del modelo YOLO [1-27], figura 2.

Figura 2. Fragmento del conjunto de datos de imágenes de entrenamiento para YOLO. [27]



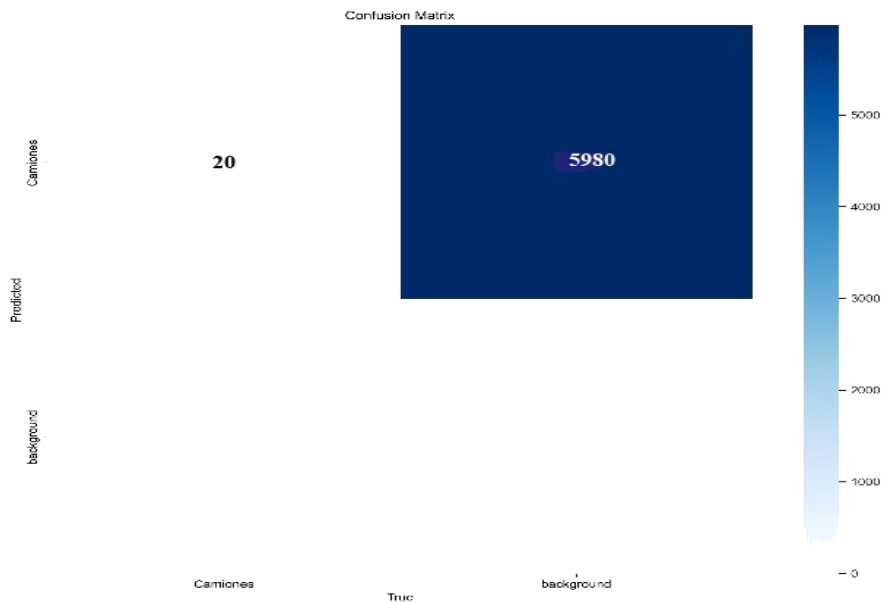
1. Clasificación de actores en la carretera: Una vez que los datos de entrada están listos, es posible llevar a cabo la clasificación de actores en la carretera utilizando el modelo YOLO. Esto implica el proceso de pasar las imágenes o fotogramas de video a través del modelo y obtener las detecciones correspondientes. Cada detección proporcionará información como las coordenadas del cuadro delimitador, la etiqueta de la clase y la confianza asociada a dicha detección [1].
2. Refinamiento de las detecciones después del proceso: Una vez obtenidas las detecciones, se puede realizar una etapa de postprocesamiento para depurar y perfeccionar los resultados. Este proceso puede implicar la eliminación de detecciones con baja confianza, la supresión no máxima para eliminar detecciones redundantes y la asignación de etiquetas a las detecciones siguiendo un esquema de clasificación [1].
3. Visualización de los resultados o conjunto de ejecuciones: Finalmente, es posible visualizar los resultados de la detección de actores en la carretera. Esto podría incluir la superposición de cuadros delimitadores y etiquetas de clases en las imágenes o videos,

permitiendo una representación visual de las detecciones realizadas por el modelo YOLO [1].

4. Matriz de confusión: Una matriz de confusión es una tabla que presenta las clases reales de objetos y las clases predichas por el modelo, figura 3. En el contexto de la detección de objetos, las filas de la matriz representan las clases reales de los objetos, mientras que las columnas representan las clases predichas por el modelo [28]. La matriz de confusión se divide en cuatro celdas:

- Verdaderos positivos (TP): Indica la cantidad de objetos que el modelo ha detectado correctamente y clasificado correctamente como positivos.
- Verdaderos negativos (TN): Indica la cantidad de objetos que el modelo ha clasificado correctamente como negativos.
- Falsos positivos (FP): Indica la cantidad de objetos que el modelo ha clasificado incorrectamente como positivos.
- Falsos negativos (FN): Indica la cantidad de objetos que el modelo ha clasificado incorrectamente como negativos.

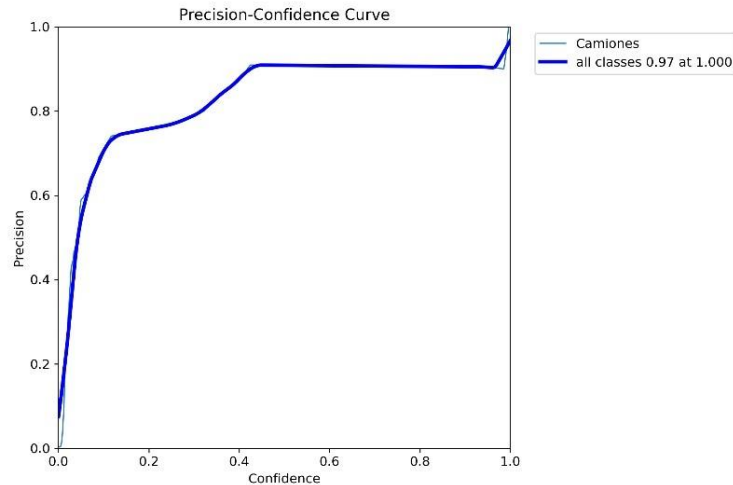
Figura 3. Matriz de confusión de una capacitación para imágenes de camiones. [28]



Esta curva, figura 4, se genera trazando la precisión en función de varios umbrales de confianza. Muestra cómo la precisión varía con distintos valores de confianza, permitiendo examinar el equilibrio entre precisión y cantidad de detecciones. Un umbral de confianza más elevado conduce a un menor número de detecciones, pero estas son más precisas.

La curva de precisión-confianza resulta fundamental para ajustar los parámetros del modelo y alcanzar el equilibrio deseado entre precisión y cantidad de detecciones en el contexto de la detección de objetos basada en YOLO [29].

Figura 4. Curva de confianza-precisión sobre el entrenamiento de imágenes de camiones.

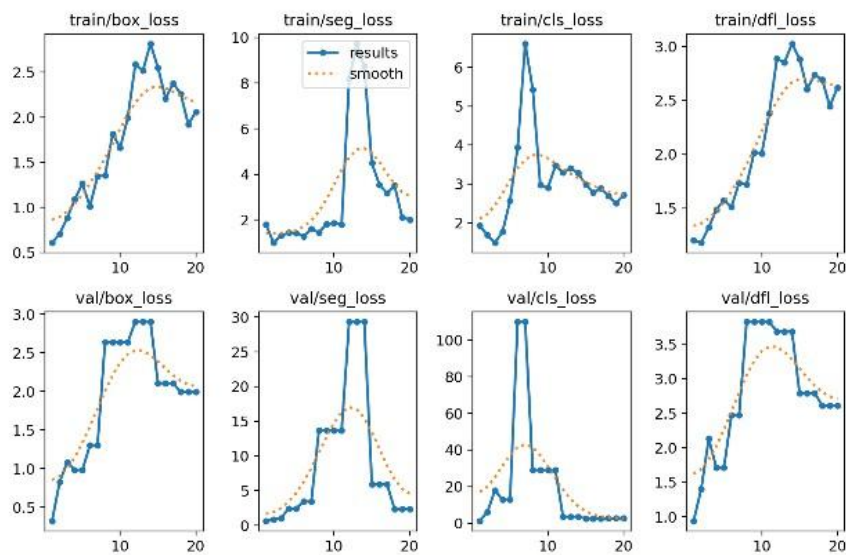


- Resultados: El archivo "results.png" representa de manera visual el proceso de entrenamiento del modelo YOLOv8. Contiene diversos tipos de información que incluyen:
 1. Gráficos de pérdida: Estos gráficos muestran cómo evoluciona la función de pérdida a lo largo de diferentes épocas o iteraciones durante el entrenamiento. Su finalidad es evaluar el progreso del aprendizaje del modelo y su mejora con el tiempo.
 2. Curvas de precisión: Estas curvas ilustran cómo aumenta la precisión del modelo durante el proceso de capacitación. Las curvas de precisión pueden organizarse por clase o incluir métricas como la precisión promedio o el *IoU* medio para las detecciones.
 3. Visualización de resultados: En esta sección, se presentan ejemplos visuales de imágenes de entrenamiento con las detecciones realizadas por el modelo en diferentes etapas del proceso de entrenamiento. Esto proporciona una evaluación visual del mejoramiento en la capacidad de detección de objetos del modelo.

El propósito fundamental de "results.png" es proporcionar a los investigadores y desarrolladores una representación visual que resuma y describa el progreso y el rendimiento del entrenamiento del modelo, figura 5. Esto facilita el seguimiento, análisis y ajustes necesarios para mejorar el modelo y su configuración [30].

- Entrenamiento de imágenes: Las imágenes de "entrenamiento" generadas en el contexto de YOLO (You Only Look Once) representan los resultados de la detección de objetos durante la fase de entrenamiento del modelo. Después de cada iteración, el modelo realiza predicciones y produce estas imágenes que muestran los objetos detectados junto con cuadros delimitadores y etiquetas correspondientes. Estas imágenes desempeñan un papel visual fundamental para evaluar el desempeño del modelo mientras aprende a detectar objetos, ajustar los cuadros delimitadores y asignar las etiquetas adecuadas. Es crucial destacar que estas imágenes de "entrenamiento" son específicas de la etapa de capacitación y se emplean para supervisar y analizar el rendimiento intermedio del modelo. Una vez que el modelo ha sido entrenado, se somete a evaluación utilizando otras imágenes en situaciones del mundo real [31], figura 6.

Figura 5. Salida "Results.png" para entrenamiento de imágenes sobre Camiones. [30]



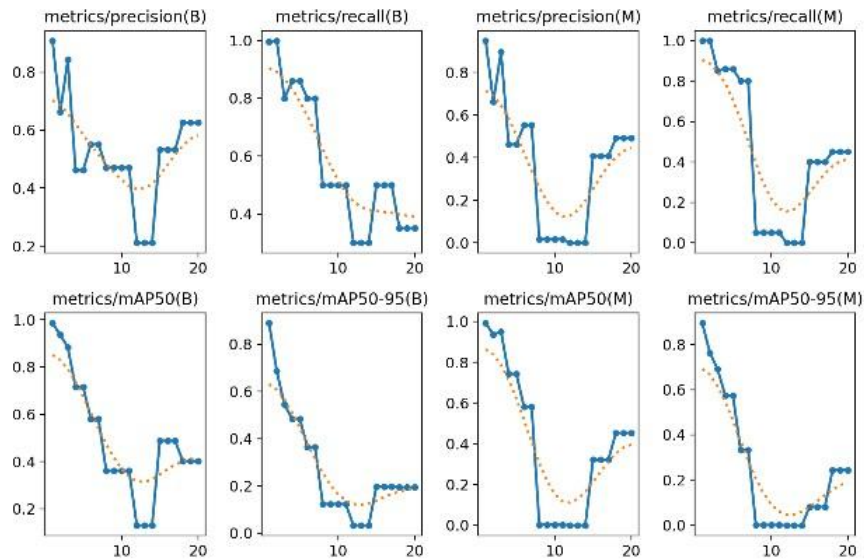


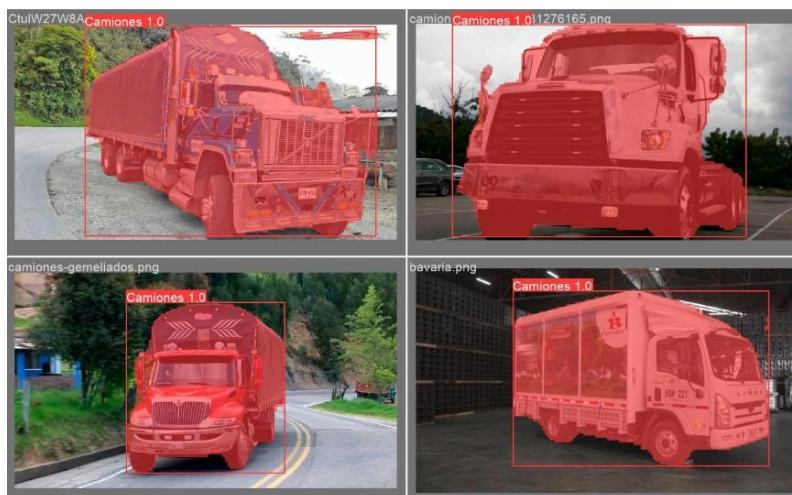
Figura 6. "Train" imágenes de salida para entrenamiento de imágenes sobre Camiones. [31]



- **Imágenes de Validación:** En el proceso de entrenamiento de YOLO, las imágenes de validación se utilizan con el propósito de evaluar el desempeño del modelo en datos nuevos y previamente no observados. Estas imágenes son diferentes al conjunto de imágenes de entrenamiento y desempeñan un papel fundamental en la estimación de la capacidad del modelo para generalizar sus conocimientos a situaciones desconocidas. Para llevar a cabo esta evaluación, se recurre a la técnica de validación cruzada durante el proceso de entrenamiento de YOLO. Esto implica la evaluación de la precisión, recuperación, puntuación de confianza y otras métricas del modelo utilizando este conjunto

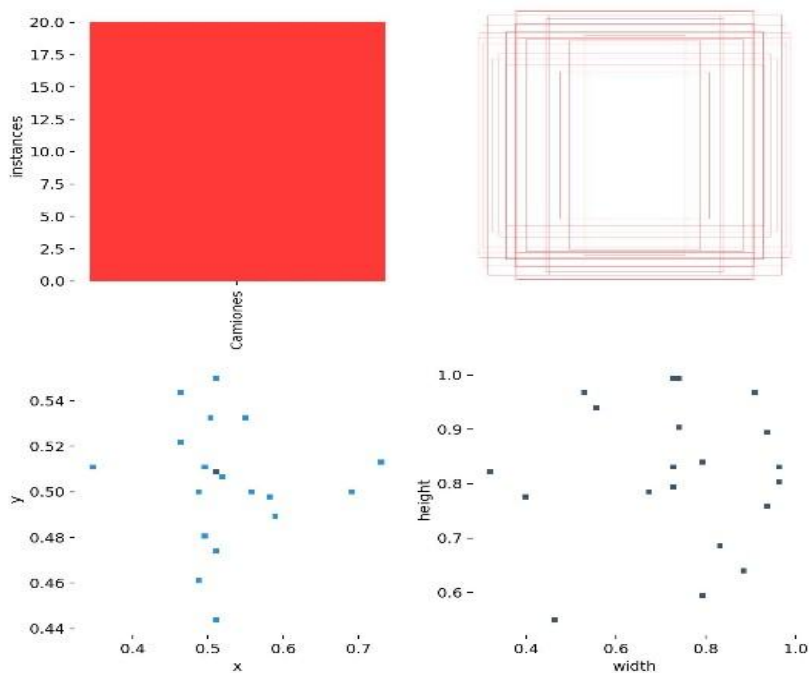
de datos de validación independiente, que incluye las imágenes de validación, figura 7. Esta metodología asegura una evaluación completa de las habilidades del modelo en relación con datos que no han sido previamente observados [32].

Figura 7. Imágenes de salida "Val" para entrenamiento de imágenes en Camiones. [32]



- Etiquetas: Durante el proceso de entrenamiento de YOLO (You Only Look Once), el archivo "labels.png" desempeña la función de proporcionar una representación visual de las etiquetas o categorías asignadas a los objetos detectados en las imágenes utilizadas para el entrenamiento. Esta representación muestra las imágenes originales con cuadros delimitadores que rodean a los objetos detectados, cada uno de los cuales está etiquetado con su categoría correspondiente. La asignación de estas etiquetas, figura 8, se realiza de manera manual durante el proceso de entrenamiento y consta de la clase del objeto y la ubicación del cuadro delimitador. La imagen generada tiene un propósito esencial al permitir la supervisión del progreso del aprendizaje del modelo, asegurar una asignación precisa de etiquetas a los objetos y observar cómo se adaptan los cuadros delimitadores durante el entrenamiento. Esta retroalimentación visual mejora la capacidad de verificar y perfeccionar las etiquetas utilizadas para entrenar el modelo YOLO, lo que, en última instancia, contribuye a mejorar su rendimiento y precisión en la detección de objetos [33].

Figura 8. Salida de los gráficos de correlación de etiquetas para el entrenamiento de imágenes en Camiones. [33]

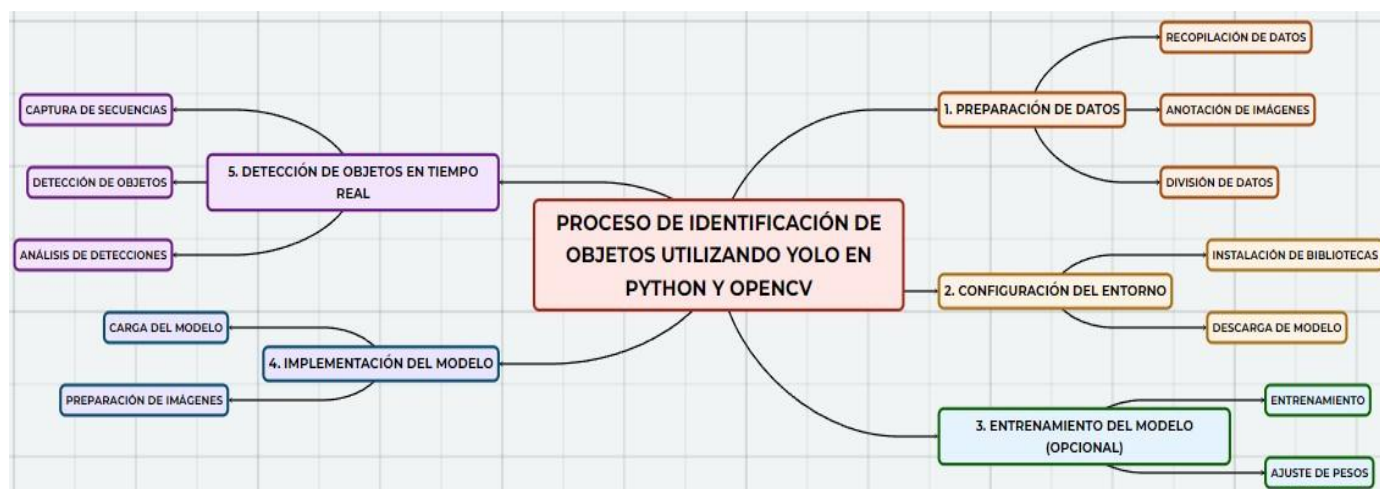


La implementación de YOLO en Python demanda un hardware potente para llevar a cabo inferencias en tiempo real. El desempeño del modelo se encuentra sujeto a variaciones dependiendo del tamaño y de los datos de entrada utilizados. El proceso de entrenamiento, figura 9, abarca la recolección de datos, el etiquetado de los mismos y la capacitación del modelo, lo cual puede requerir varias horas de dedicación [34].

Una vez concluido el entrenamiento, es imperativo evaluar el modelo empleando un conjunto de datos de prueba con el fin de medir su desempeño y, si fuese necesario, realizar ajustes. Una vez completada esta evaluación, el modelo puede ser implementado en sistemas en tiempo real, tales como cámaras de seguridad, Camiones autónomos o drones.

El empleo de YOLO en Python se traduce en una capacidad de detección precisa y eficiente en tiempo real de los actores presentes en la carretera, lo que contribuye a mejorar tanto la seguridad como la gestión del tráfico [2].

Figura 9. El esquema de entrenamiento general utilizado para el reconocimiento de imágenes con YOLO. [34]



Fundamentos clave en YOLO para la detección de actores viales en tiempo real

En el ámbito de la inteligencia artificial y la visión por computadora, YOLO (You Only Look Once) ha emergido como un enfoque altamente eficiente y preciso para la detección de objetos en imágenes. En particular, YOLO se ha utilizado de manera extensa en la detección de actores viales, como camiones, peatones y señales de tráfico, en aplicaciones relacionadas con la conducción autónoma y la seguridad vial. Con el propósito de lograr una detección precisa y en tiempo real de los actores de la carretera, YOLO se apoya en una serie de fundamentos clave, los cuales se examinarán a continuación:

- **Función de pérdida:** La función de pérdida es una herramienta que cuantifica la discrepancia entre la salida pronosticada del modelo y los datos reales. Esta función se

emplea para entrenar el modelo, ajustando sus parámetros de tal manera que se minimice la función de pérdida (Ecuación 1).

- La función de pérdida de YOLO se compone de varios componentes diferentes. Cada componente está diseñado para penalizar errores específicos en las predicciones del modelo. Los componentes de la función de pérdida se dividen de la siguiente manera:
 - Pérdida de cuadro: Este componente sanciona las predicciones incorrectas en relación con el cuadro delimitador del objeto.
 - Pérdida de clasificación sin objeto: Este componente penaliza las predicciones erróneas de la clase cuando no hay un objeto presente.
 - Pérdida de IoU: Este componente penaliza las predicciones incorrectas en relación con la medida de intersección sobre la unión (IoU) entre los cuadros delimitadores previstos y las etiquetas de entrenamiento.
 - Pérdida de compensación: Este componente penaliza las predicciones incorrectas en lo que respecta al ajuste del desplazamiento entre los cuadros delimitadores pronosticados y las etiquetas de entrenamiento.

La función de pérdida global de YOLO se define como la suma de todos estos componentes.

En esta ecuación, se describen los diferentes componentes de la función de pérdida, identificando cada uno de ellos de la siguiente manera:

$$L = Lcls + Lbox + Lnoobj + Liou + Lcomp \quad (1)$$

dónde:

- *Lcls*: Corresponde a la pérdida de clasificación.

- *Lbox*: Representa la pérdida relacionada con la caja delimitadora.
 - *Lnoobj*: Indica la pérdida de clasificación sin objeto.
 - *Liou*: Hace referencia a la pérdida de IoU (Intersección sobre Unión).
 - *Lcomp*: Se refiere a la pérdida compensada.
- Descenso de gradiente: El descenso de gradiente es una técnica empleada para minimizar la función de pérdida, ajustando los parámetros del modelo en la dirección en la que la función de pérdida disminuye con mayor rapidez. Esta optimización se lleva a cabo conforme a la Ecuación (2).

$$\frac{\partial L}{\partial \theta} = \frac{\partial L_{cls}}{\partial \theta} + \frac{\partial L_{box}}{\partial \theta} + \frac{\partial L_{noobj}}{\partial \theta} + \frac{\partial L_{iou}}{\partial \theta} + \frac{\partial L_{comp}}{\partial \theta} \quad (2)$$

dónde:

- θ se refiere a los parámetros del modelo.
- L corresponde a la función de pérdida.
- Lcls representa la pérdida de clasificación.
- Lbox indica la pérdida relacionada con las cajas delimitadoras.
- Lnoobj es la pérdida de clasificación cuando no hay objeto presente.
- Liou es la pérdida relacionada con la intersección.
- Lcomp hace referencia a la pérdida compensada.

- Red Neuronal Convolutacional: Una red neuronal convolutacional (CNN) es un tipo de estructura de red neuronal especialmente diseñada para el procesamiento de imágenes. En el contexto de YOLO, se emplean las CNN para la tarea de identificar y categorizar objetos presentes en imágenes. Esto se describe mediante la Ecuación (3).

$$y = f(Wx + b) \quad (3)$$

dónde:

- y es la salida de la CNN,
 - f es la función de activación,
 - W son los pesos de la CNN,
 - b son los sesgos de la CNN, y
 - x es la entrada a la CNN.
-
- Bóveda y regresión: Se refiere a una estructura empleada con el propósito de guardar datos, y en el contexto de YOLO, se utilizan bóvedas para almacenar tanto los pesos del modelo como los datos de entrenamiento. La regresión se clasifica como un tipo de enfoque dentro del aprendizaje automático que se utiliza con el fin de realizar predicciones relacionadas con valores continuos. En YOLO, se emplean técnicas de regresión para prever tanto la ubicación como el tipo de objetos presentes en las imágenes. Esto se describe mediante la Ecuación (4)

$$B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{bmatrix} \quad (4)$$

dónde:

- B es la bóveda,
- b_{ij} es el objeto en la celda i, fila j,
- m es el número de celdas en la bóveda, y
- n es el número de objetos en cada celda.

Cada objeto de la bóveda se representa como un vector de cinco dimensiones. Ecuación (5)

$$[x, y, w, h, c] \quad (5)$$

dónde:

- x e y son las coordenadas del centro del cuadro delimitador,
- w y h son el ancho y el alto del cuadro delimitador,
- c es la clase del objeto.

La bóveda tiene la función de producir la salida de YOLO, que se presenta como una lista de objetos. Cada uno de estos objetos incluye información como un cuadro delimitador, una clase y una probabilidad. La probabilidad refleja la posibilidad de que el objeto contenido en el cuadro delimitador corresponda a la clase específica mencionada.

4. Conclusiones

El modelo YOLO (You Only Look Once) ofrece varias ventajas significativas en la tarea de identificación de actores del tráfico, destacando por su alta precisión, eficiencia y capacidad de operar en tiempo real. Su capacidad para detectar y clasificar múltiples objetos simultáneamente lo convierte en una opción altamente adecuada para entornos de tráfico, lo que beneficia la gestión del tráfico y la seguridad vial. Además, su facilidad de implementación a través de opciones de código abierto como Python ha contribuido a su creciente popularidad. No obstante, es importante señalar que presenta algunas limitaciones, como la dificultad en detectar objetos pequeños o muy cercanos, así como desafíos en condiciones de iluminación desfavorable u obstrucciones.

Referencias

- [1] Shao, Y., Zhang, D., Chu, H., Zhang, X., & Rao, Y. (2021). A Review of YOLO Object Detection Based on Deep Learning.
- [2] Bhasin, S. (2019). Real-time Object Detection with YOLO, OpenCV and Python.
- [3] Suresh et al. (2020). Object Detection with YOLO for Intelligent Traffic Monitoring System.
- [4] Liu, Y., Shi, Q., Guo, W., & Liao, W. (2020). A Real-time, Mobile-object Detection Approach for Unmanned Aerial Vehicle Based Forest Fire Surveillance System.
- [5] Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of YOLO Algorithm Developments.
- [6] Mauro Tucci, A. B. (s/f). "YOLO-S: A Lightweight and Accurate YOLO-like Network for Small Target Selection in Aerial Imagery".

- [7] Sharma, A., Pathak, J., Prakash, M., & Singh, J. N. (2020). Object Detection using OpenCV and Python. *International Journal of Innovative Research in Computer and Communication Engineering*, 8(6), 2736-2741.
- [8] "Procesamiento de Imágenes y Visión Artificial con MATLAB Video," Mathworks.com, 2021. <https://la.mathworks.com/videos/image-processing-and-computer-vision-with-matlab-1597884648964.html> (accessed Jul. 25, 2023).
- [9] Ricardo Alirio Gonzalez, R. Ferro, and Darío Liberona, "Government and governance in intelligent cities, smart transportation study case in Bogotá Colombia," vol. 11, no. 1, pp. 25–34, Mar. 2020, doi: <https://doi.org/10.1016/j.asej.2019.05.002>.
- [10] Beatriz Elena Pineda, Claudia Helena Muñoz, & Gil, H. (2018). Aspectos relevantes de la movilidad y su relación con el medio ambiente en el Valle de Aburrá: una revisión. *Ingeniería Y Desarrollo*, 36(2), 489–508. <https://www.redalyc.org/journal/852/85259689013/html/>
- [11] IA por el Planeta: Destacando las innovaciones de IA para la movilidad sostenible y las ciudades inteligentes. (2023). Unesco.org. <https://www.unesco.org/es/articles/ia-por-el-planeta-destacando-las-innovaciones-de-ia-para-la-movilidad-sostenible-y-las-ciudades>
- [12] Gómez Zapata, C. A. (2019). Reconocimiento de objetos del hogar, usando redes neuronales convolucionales para personas con discapacidad visual. *Revista Científica de Ingeniería y Tecnología*, 2(2), 1-10. <https://dialnet.unirioja.es/descarga/articulo/7436051.pdf>.
- [13] Olabe, X. B. (s/f). REDES NEURONALES ARTIFICIALES Y SUS APLICACIONES. Ehu.eus. Recuperado el 8 de julio de 2023, de URL: https://ocw.ehu.es/pluginfile.php/40137/mod_resource/content/1/redes_neuro/contenidos/pdf/libro-del-curso.pdf
- [14] Murgui, J., & García-Sánchez, A. J. (2018). Clasificación y reconocimiento de imágenes con redes neuronales para aplicaciones industriales. URL: <https://riunet.upv.es/bitstream/handle/10251/115464/Murgui.pdf?sequence=1>

- [15] Ortiz, G., & Sánchez, A. I. (2020). Emprendimiento y tecnologías de la información y la comunicación en Bogotá. *Cuadernos de Administración*, 36(67), 199-211.
- [16] Torres, J., & Acosta, H. (2019). La innovación en el ecosistema emprendedor de Bogotá. *Cuadernos de Administración*, 35(64), 251-262.
- [17] Uribe, F., & Guzmán, J. (2021). La colaboración público-privada en el fomento de la innovación en Bogotá: el caso de la identificación de objetos en el contexto vial. *Revista Internacional de Gestión y Economía Aplicada*, 11(1), 89-101.
- [18] Centro de Investigación de la Universidad Distrital Francisco José de Caldas. (2023). Udistrital.edu.co. <https://revistas.udistrital.edu.co/index.php/visele/article/view/18942/18701>
- [19] Chiroma, R. C. U. (2021). Vehicle detection, counting, and classification in traffic videos: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 22(10), 3773-3785.
- [20] Rao, S. S. (2018). Vehicle detection and identification using computer vision and deep learning techniques. *IEEE Transactions on Intelligent Transportation Systems*, 19(10), 2827-2836.
- [21] Akhand, M. A. H. (2019). Vehicle Recognition from License Plate Number using Deep Learning. arXiv preprint arXiv:1903.09203.
- [22] Sandra Milena García Ávila, Cristian Alexander Vega Camacho, José Vicente Cadena López, Ricardo Alirio González Bustamante, Paola Andrea Mateus Abaunza. (2021). Diseño y aplicación de una herramienta para identificar y clasificar motocicletas mediante una red neuronal convolucional. researchgate.net. URL: <https://doi.org/ISBN:978-958-53278-6-3>
- [23] valentynsichkar, "Traffic Signs Detection by YOLO v3, OpenCV, Keras," Kaggle.com, Apr. 15, 2022. <https://www.kaggle.com/code/valentynsichkar/traffic-signs-detection-by-yolo-v3-opencv-keras> (accessed Jul. 25, 2023).

- [24] Motor Colombia. (2022, February 23). 7.270 muertos en accidentes de tránsito en 2021. Motor Colombia; Motor Colombia. URL: <https://www.motor.com.co/industria/7.270-muertos-en-accidentes-de-transito-en-2021-20220124-0001.html>
- [25] R. Jiménez Moreno, O. Avilés, y D. M. Ovalle, “Red neuronal convolucional para discriminar herramientas en robótica asistencial”, *Vis. Electron.*, vol. 12, no. 2, pp. 208–214, oct. 2018. <https://doi.org/10.14483/22484728.13996>
- [26] L. L. Hurtado-Cortés y J. A. Forero-Casallas, “Identification and fault detection in actuator using NN-NARX”, *Vis. Electron.*, vol. 2, no. 2, pp. 304–312, dic. 2019. <https://doi.org/10.14483/22484728.18432>
- [27] Propia. (2023). Fragmento del conjunto de imágenes de entrenamiento para YOLO [Figura].
- [28] Propia. (2023). Matriz de confusión de una capacitación sobre imágenes de Camiones. [Figura].
- [29] Propia. (2023). Curva de precisión-confianza para el entrenamiento de imágenes de Camiones. [Figura].
- [30] Propia. (2023). Salida "Results.png" sobre el entrenamiento de imágenes de Camiones. [Figura].
- [31] Propia. (2023). Salida "Train.png" sobre el entrenamiento de imágenes de Camiones. [Figura].
- [32] Propia. (2023). Salida "Val.png" sobre el entrenamiento para Camiones. [Figura].
- [33] Propia. (2023). Salida de los gráficos de correlación de etiquetas para el entrenamiento de imágenes de Camiones. [Figura].
- [34] Propia. (2023). Esquema de entrenamiento general utilizado para el reconocimiento de imágenes con YOLO. [Figura]

