

Técnicas de visión por computadora para la estimación de la intensidad, la velocidad y la densidad en el flujo vehicular.

Computer vision techniques for estimation of intensity, speed and density in vehicular flow.

David Salazar-Solórzano

Ciencias de la Computación

Universidad Técnica Particular de Loja

Loja, Ecuador

dasalazar14@utpl.edu.ec

Jorge Flores-Ordóñez

Ciencias de la Computación

Universidad Técnica Particular de Loja

Loja, Ecuador

jeflores17@utpl.edu.ec

Luis Barba-Guaman

Ciencias de la Computación y Electrónica

Universidad Técnica Particular de Loja

Loja, Ecuador

lrbarba@utpl.edu.ec

Resumen—El presente documento describe la implementación y evaluación de varios algoritmos de visión por computadora, los cuales son YOLOv7, YOLOv5, YOLOv4, SSD MobileNet, EfficientDet D0 y SSD MobileNet LITE, con el objetivo de calcular las principales magnitudes del flujo vehicular como la intensidad, densidad y velocidad. Además, se exploró el uso del algoritmo de seguimiento de objetos StrongSort y se emplearon los frameworks de Deep Learnig TensorFlow, Darknet y PyTorch.

Por otro lado, se realizaron cuatro experimentos en diversas condiciones de luminosidad, determinando que las arquitecturas de YOLOv5 y YOLOv7 sobresalen como las más efectivas tanto en situaciones de alta como de baja luminosidad, superando al resto de los modelos. Finalmente, se logró desarrollar un prototipo de software altamente eficiente para obtener las principales magnitudes del flujo vehicular, lo que representa un avance significativo en el campo de la visión por computadora aplicada al tráfico.

Palabras clave—flujo vehicular, visión por computadora, YOLO.

Abstract—This paper describes the implementation and evaluation of several computer vision algorithms, which are YOLOv7, YOLOv5, YOLOv4, SSD MobileNet, EfficientDet D0 and SSD MobileNet LITE, with the objective of calculating the main magnitudes of vehicular flow such as intensity, density and speed. In addition, the use of the StrongSort object tracking algorithm was explored and the Deep Learnig frameworks TensorFlow, Darknet and PyTorch were employed.

On the other hand, four experiments were carried out in different lighting conditions, determining that the YOLOv5 and YOLOv7 architectures stand out as the most effective in both high and low lighting situations, outperforming the rest of the models. Finally, a highly efficient software prototype was developed to obtain the main magnitudes of vehicular flow, which represents a significant advance in the field of computer vision applied to traffic.

Keywords—traffic flow, computer vision, YOLO.

I. INTRODUCCIÓN

En las últimas cinco décadas, se ha venido dando un rápido incremento en las tasas de urbanización en todo el mundo.

De acuerdo con las Perspectivas de Urbanización Mundial de las Naciones Unidas, se estima que para el año 2050, más de dos tercios de la población mundial vivirán en áreas urbanas [1]. Esta acelerada transición hacia entornos urbanos se debe principalmente a la necesidad de las personas de mejorar sus condiciones de vida. No obstante, este crecimiento desenfrenado conlleva diversos desafíos, siendo uno de los más apremiantes el aumento del tráfico vehicular debido al crecimiento exponencial del parque automotor. Situación que provoca embotellamientos, accidentes y estrés en los conductores, a ello se suma las pérdidas significativas de tiempo a causa del flujo vehicular, especialmente en América Latina.

En relación con el crecimiento desenfrenado de las áreas urbanas mencionado anteriormente, estudios realizados en 2021 demuestran que las ciudades de la región enfrentan un elevado número de embotellamientos. Por ejemplo, Bogotá es considerada una de las urbes con el peor tráfico vehicular, donde los ciudadanos pierden aproximadamente 94 horas al año, seguido de Monterrey con 70 horas y Ciudad de México con 67 horas [2]. Este alarmante panorama pone en evidencia la necesidad de emplear soluciones inteligentes para una gestión eficiente del flujo vehicular.

En respuesta a la alarmante situación, el desarrollo de sistemas de manejo inteligente del tráfico se ha convertido en una área de investigación crucial en busca de resolver inconvenientes asociados al transporte en los núcleos urbanos. Por tal motivo, se hace relevante el estudio y análisis de las principales magnitudes, las cuales son: la intensidad, densidad y velocidad vehicular, mediante el empleo de técnicas de visión por computadora. Estos métodos a su vez permiten recopilar la información necesaria y precisa para enfrentar el desafío del flujo vehicular.

A continuación, se presenta la estructura de este artículo. En la Sección I se expone la introducción, donde se contextualiza el tema. La Sección II describe las arquitecturas utilizadas

y su aplicación en diversos campos. La Sección III detalla los métodos, materiales, procedimientos y herramientas empleadas. La Sección IV muestra los resultados de los cuatro experimentos y el prototipo desarrollado. Finalmente, en la Sección V se presentan las conclusiones de la investigación y se ofrecen recomendaciones para futuros trabajos a desarrollarse en esta área de estudio.

II. ESTADO DEL ARTE

En esta sección, se realizará una revisión del estado del arte en visión por computadora aplicada al tráfico vehicular. A continuación, se describirán las principales arquitecturas y técnicas, así como los enfoques más destacados para el análisis del flujo vehicular en entornos urbanos.

En el campo de la visión por computadora aplicada al flujo vehicular se han propuesto diferentes soluciones como sistemas de monitoreo que utilizan redes neuronales convolucionales difusas (CFNN), estas son un tipo de red neuronal artificial que combina técnicas de aprendizaje profundo y lógica difusa para mejorar la precisión en la detección y clasificación de objetos [3], por ello tienen la capacidad de registrar el volumen de circulación y la información de distintos tipos de vehículos, enfocándose en la detección y cómputo para el cálculo del flujo [4].

Así mismo, se han planteado un nuevo método de conteo basado en detección de imágenes espacio temporales (TSI), que usa una combinación de información espacial y temporal para mejorar su precisión. Para ello se diseñó una red base de seis capas con un máximo de 60 canales, obteniendo resultados que presentan mejoras respecto a los métodos tradicionales [5].

Estas técnicas, CFNN y TSI, han demostrado ser efectivas en el análisis del flujo vehicular en entornos urbanos, proporcionando una mayor precisión en la detección y clasificación de objetos y en el conteo de vehículos. Sin embargo, no son las únicas arquitecturas que se han utilizado con éxito en este campo. A continuación, se describirán otras arquitecturas populares para la detección de objetos en tiempo real, como EfficientDet D0, SSD MobileNet y su variante más ligera, SSD MobileNet LITE.

Con respecto a EfficientDet D0, este es un modelo de detección de objetos escalable y eficiente desarrollado por el equipo de Google Brain. Utiliza una red neuronal convolucional EfficientNet como columna vertebral y una nueva red de características piramidales bidireccionales (BiFPN) para lograr una alta precisión mientras emplea menos parámetros y cálculos que otros detectores anteriores [6].

Por otro lado, el detector de objetos SSD MobileNet destaca por ser una arquitectura de detección de objetos basada en la combinación de una red neuronal convolucional MobileNet y un detector de objetos de una sola toma (SSD). Esta arquitectura es popular debido a su velocidad y eficiencia en la detección de objetos en tiempo real [7]. Por ello, se empleó la misma base para la creación de SSD MobileNet LITE, que se distingue por ser una variante más ligera y eficiente, que usa capas convolucionales separables en profundidad

para reducir el número de parámetros y mejorar la eficiencia computacional.

Ante lo mencionado anteriormente, en los últimos años ha aumentado el interés en la investigación sobre Inteligencia Artificial, con especial enfoque en resolver problemas relacionados con el tráfico. En ese sentido, los avances en la creación de los detectores de objetos basados en redes neuronales convolucionales (CNN) han evolucionado considerablemente. Uno de los grandes logros en esta área es la arquitectura YOLO (You Only Look Once), desarrollada por Joseph Redmon en 2015 [8]. YOLO utiliza una CNN para predecir las coordenadas de los cuadros delimitadores y las clases de los objetos en una imagen.

A continuación, se muestra tres versiones de YOLO que serán parte de los experimentos:

a) *YOLOv4*: Es un modelo de detección de objetos eficiente y potente que funciona con Darknet, capaz de entrenarse en una GPU convencional de 8 o 16 GB-VRAM. Además, logra una gran velocidad en tiempo real en el conjunto de datos MS COCO de un 43,5% AP funcionando a 65 FPS en un Tesla V100 [9].

Estas características han llevado a YOLOv4 a su aplicación en diversas áreas de estudio, una de ellas es la detección del flujo vehicular. Ejemplo de ello es la modificación de la arquitectura para aumentar los tiempos de convolución después de la capa de entidades. Donde, por un lado, se profundiza en la red y se aumenta la capacidad y complejidad de la misma y, por otro lado, se obtiene la información de características de forma global y de mayor nivel semántico, logrando una precisión estadística mejorada del 93%, siendo un 2% más alta que la versión base [10].

b) *YOLOv5*: Es el primer modelo que se escribe en el marco de PyTorch, siendo más ligero y fácil de usar [11]. La principal diferencia con la versión anterior es la velocidad de entrenamiento mejorada [12]. Sin embargo, diferentes pruebas muestran que YOLOv4 es más rápido y más preciso [13]. En fin, esta nueva versión resulta un modelo bastante bueno, con mejoras en algunos aspectos, pero aún no supera con diferencia a su antecesor.

Sin embargo, YOLOv5 ha sido utilizado en varios estudios, incluyendo la detección de objetos a partir de la fusión de datos de radar de ondas milimétricas, lo cual es esencial para que los vehículos autónomos perciban su entorno circundante [14]. Además, de proponer un método mejorado de detección de vehículos en diversos escenarios de tráfico, haciendo uso del algoritmo Flip-Mosaic para mejorar la detección de objetos pequeños en la red y reducir la tasa de falsas detecciones de vehículos causadas por la oclusión [15].

c) *YOLO v7*: Es un modelo que supera a todos los detectores de objetos conocidos tanto en velocidad como en precisión en un rango de 5 FPS a 160 FPS y tiene una mayor precisión entre todos los detectores de objetos en tiempo real disponibles, siendo esta de 56,8% [16].

Al ser un modelo relativamente nuevo, aún no existe un significativo número de estudios que muestren su efectividad en la detección vehicular, sin embargo, al compararlo con

otros detectores de objetos conocidos, YOLOv7 puede reducir efectivamente alrededor del 40% de parámetros y el 50% de cálculo de detecciones de objetos en tiempo real y lograr una velocidad de inferencia más rápida y una mayor precisión de detección [17].

En resumen, en este apartado se ha presentado una revisión de las arquitecturas y técnicas en visión por computadora aplicada al tráfico vehicular, incluyendo CFNN, TSI y varias versiones de YOLO. Estos avances han mejorado la precisión y eficiencia en la detección y clasificación de objetos y conteo de vehículos. Por ello, en la presente investigación se evaluará específicamente los modelos YOLOv7, YOLOv5, YOLOv4, SSD MobileNet, EfficientDet D0 y SSD MobileNet LITE.

III. MÉTODOS Y MATERIALES

En esta sección se presenta el diseño general del prototipo, los materiales, herramientas y métodos utilizados durante el desarrollo de la investigación, destacando el uso de StrongSort [18] como algoritmo de seguimiento de objetos y el cálculo de las magnitudes de flujo vehicular.

A. Diseño General

La arquitectura general del prototipo se divide en tres etapas: datos de entrada, detección de vehículos y salida. El proceso inicia con la configuración de los parámetros y la carga de los archivos multimedia. Posteriormente, se realiza la detección y cálculo de las magnitudes del flujo vehicular en simultáneo. Finalmente, como resultado se obtiene una imagen o video analizado y procesado con los resultados de la intensidad, densidad y velocidad según se requiera. Además, se genera un archivo CSV para recopilar datos para futuros análisis. La Figura 1 muestra una representación gráfica de la arquitectura.

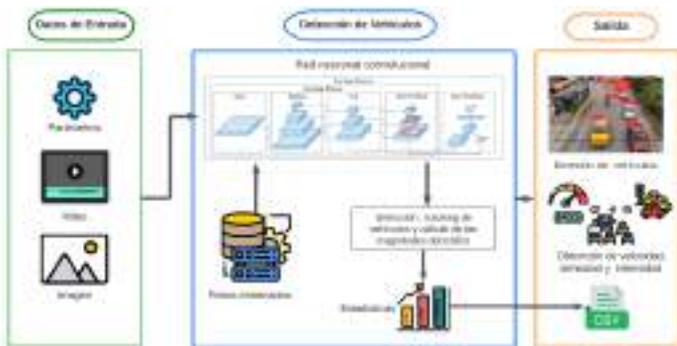


Figura 1. Arquitectura general del sistema.

B. Materiales

Los materiales utilizados en este trabajo son imágenes y videos. En primer lugar, para el entrenamiento de los modelos de detección se requiere un conjunto de imágenes, para ello se usó Open Images Dataset v7 [19] y fotos de autoría propia, conformando un dataset total de 10 000 imágenes. Además, para el etiquetado y procesado de las imágenes se empleó la herramienta Roboflow [20]. A lo mencionado anteriormente,

es importante recalcar que solo se etiquetó una sola clase denominada vehículos.

En segundo punto, para la evaluación de los modelos se grabó un total de 20 videos, 10 en un ambiente diurno y 10 en un ambiente nocturno, cada video tiene una resolución de 1280 x 720 píxeles a 60 FPS y una duración de un minuto. Estos videos permitieron evaluar la eficacia y eficiencia de los algoritmos en distintas condiciones de luminosidad.

Por otro lado, para el proceso de entrenamiento se dividió el conjunto de imágenes en 70% para training set, 20% validation set y 10% testing set, para ello se utilizó la herramienta Roboflow.

Finalmente, se empleó la plataforma Google Colab Pro [21] para el entrenamiento de los modelos y se utilizaron computadoras portátiles con procesador Intel Core I7 y 16 GB de RAM para ejecutar y evaluar los modelos localmente. Además, se configuró los ordenadores con los drivers necesarios, siguiendo la configuración que se observa en la Figura 2.

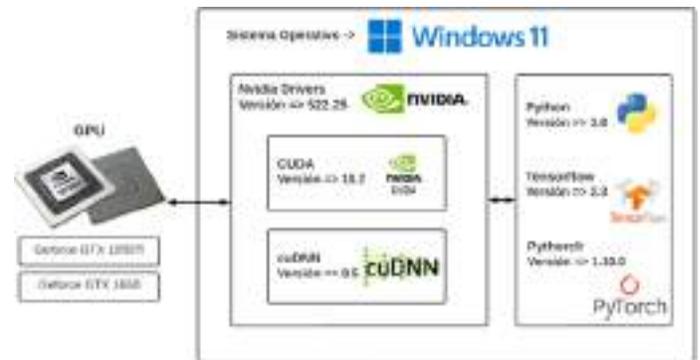


Figura 2. Configuración de entorno local.

C. Métodos

En este apartado se menciona los métodos utilizados para el cálculo de las magnitudes del flujo vehicular, el algoritmo de seguimiento StrongSort y la región de interés. Antes de ello, con base en el estado del arte, se seleccionó los modelos a evaluar en los experimentos, mismos que se muestran en la Tabla I junto al Framework y lenguaje de programación.

Tabla I
ALGORITMOS DE DETECCIÓN DE OBJETOS SELECCIONADOS

Algoritmos de detección	Deep Learning Framework	Lenguaje de Programación
EfficientDet D0	TensorFlow	Python
SSD MobileNet v2	TensorFlow	Python
SSD MobileNet v2 FPN LITE	TensorFlow	Python
YOLOv4	Darknet	Python
YOLOv5	PyTorch	Python
YOLOv7	PyTorch	Python

D. Algoritmo StrongSort en el tracking de vehículos

Los algoritmos de seguimiento de objetos son clave en el campo de la visión por computador. Por ello, basándonos en

las métricas Multi-Object Tracking Accuracy (MOTA) [22], IDF1 [23] y High Order Tracking Accuracy (HOTA) [24] que se observan en Figura 3, se eligió el algoritmo StrongSort [18] para el tracking de vehículos, debido a que es una mejora al clásico DeepSort [25], sus valores en las métricas mencionadas demuestran su eficiencia en el conjunto de datos MOT20.

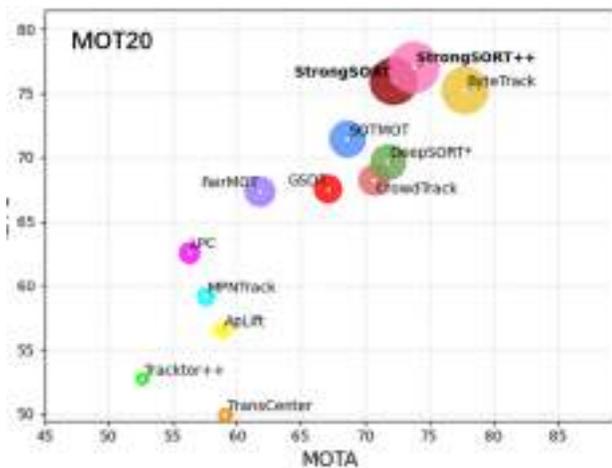


Figura 3. StrongSort en MOT20 test [18].

E. Región de interés (ROI)

Para realizar el cálculo de las magnitudes del flujo vehicular es necesario fijar una región de interés (ROI), caracterizada por ser una zona determinada de la grabación definida por coordenadas en píxeles, en la cual los modelos entrenados pueden ejecutar la detección sin gastar recursos en objetos que se encuentren fuera de ella. Para su implementación, en primer lugar, se debe establecer una máscara haciendo uso de la librería OpenCV para delimitar la zona de interés en la grabación, tal como muestra la Figura 4.

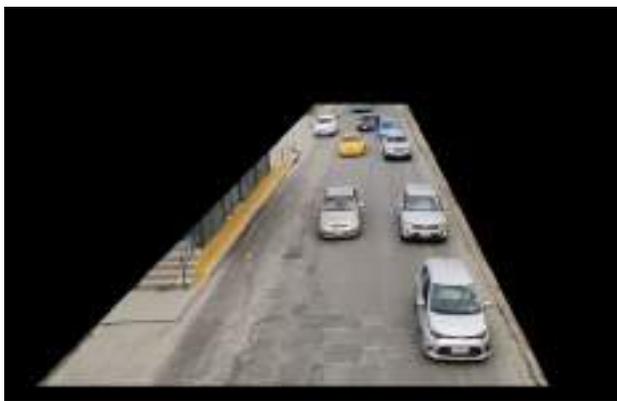


Figura 4. Máscara para obtención del ROI.

Una vez establecida la zona en píxeles con la aplicación de una máscara, se puede realizar la integración con la imagen original y realizar el dibujo de las coordenadas, obteniendo como resultado el ROI en el video, tal como se observa en la Figura 5.



Figura 5. Región de interés (ROI).

F. Cálculo de las magnitudes del flujo vehicular

Una vez establecida la región de interés, se lleva a cabo el cálculo de las magnitudes del flujo vehicular. A continuación, se describe el proceso para obtener cada una de ellas.

a) *Velocidad*: La velocidad de un cuerpo en movimiento se calcula dividiendo la distancia que recorre entre dos puntos, por el tiempo que tardó en recorrer esa distancia [26]. Para ello, se aplica la Ecuación 1, en la cual la distancia se obtiene mediante $X_f - X_i$, y el tiempo se representa por el valor en segundos que tarda cada vehículo en pasar de X_i hasta X_f , tal como se observa en la Figura 6.

$$Velocidad = \frac{distancia}{tiempo} \quad (1)$$

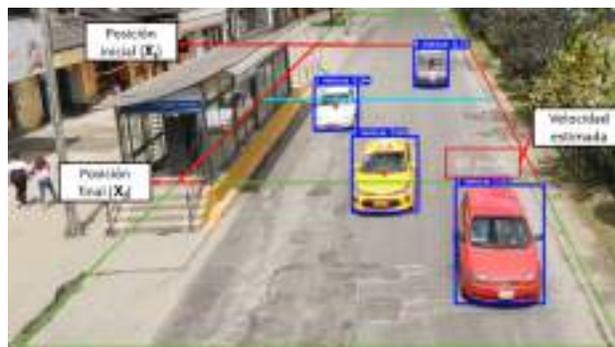


Figura 6. Cálculo de la velocidad.

Una vez que se obtiene la velocidad de cada vehículo, se procede a calcular la velocidad promedio mediante la aplicación de la Ecuación 2. Dicha métrica se calcula sumando todas las velocidades individuales de los vehículos que atraviesan el punto inicial (X_i) hasta el punto final (X_f), y luego dividiendo esa suma entre el número total de vehículos.

$$Velocidad\ promedio = \frac{velocidad\ acumulada}{vehículos\ totales} \quad (2)$$

b) *Intensidad*: La intensidad es una variable que se obtiene al contar el número de vehículos que cruzan una vía o terreno en un período de tiempo determinado. Para ello, se

utilizó la Ecuación 3, donde $n(x)$ es el número de vehículos que atraviesan una sección fija y t el intervalo de tiempo [27].

$$Intensidad = \frac{n(x)}{t} \quad (3)$$

Para el cálculo de esta magnitud, se considera la región de interés como la sección fija y se establece un intervalo de tiempo de un minuto. Además, se dibujó con OpenCV una línea de color celeste que cambiará a amarilla cuando incremente el conteo de intensidad, tal como se observa en la Figura 7.

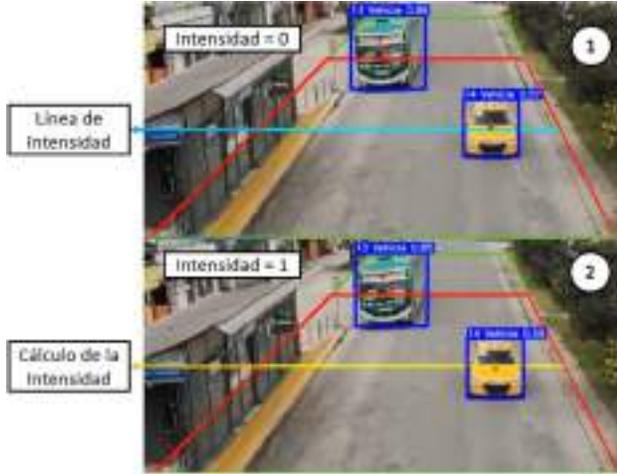


Figura 7. Cálculo de la intensidad.

c) *Densidad*: Con el fin de realizar el cálculo de la densidad se utilizó la Ecuación 4, que consiste en contar el número de vehículos presentes en un espacio y momento determinado. A diferencia de la intensidad, que se calcula durante un intervalo de tiempo, la densidad se obtiene de un frame capturado cada 20 segundos de los videos de un minuto de duración, calculado la variable en cuatro instantes de cada grabación. Un ejemplo de ello se muestra en la Figura 8.

$$Densidad = \text{Número de vehículos en el ROI} \quad (4)$$

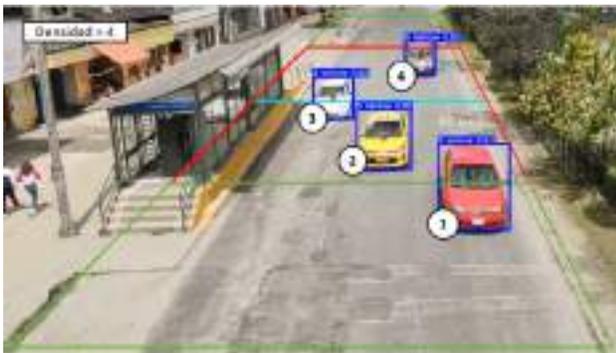


Figura 8. Cálculo de la densidad.

G. Evaluación de los modelos

En la investigación para seleccionar la mejor arquitectura, se llevó a cabo la evaluación y validación de modelos. Para ello, se utilizó la matriz de confusión y las métricas asociadas, las cuales son: precisión, exhaustividad y el ratio de éxito. A continuación, se presentan las ecuaciones de cada una de ellas.

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

IV. RESULTADOS

Se realizaron cuatro experimentos en diferentes condiciones de luminosidad, con el objetivo de comprobar la efectividad de los modelos de detección y encontrar las variables del flujo vehicular.

A. Experimento 1. Medición de precisión, exhaustividad y ratio de éxito de los modelos en horario diurno.

El objetivo de este experimento es medir la precisión, exhaustividad y ratio de éxito de los seis modelos en condiciones de alta luminosidad (horario diurno). Para ello, se utilizó la matriz de confusión para realizar el cálculo de las métricas, obteniendo los resultados que muestra la Tabla II.

Tabla II
RESULTADOS DE PRECISIÓN, EXHAUSTIVIDAD Y RATIO DE ÉXITO DE CADA MODELO EN HORARIO DIURNO

Modelos	Precisión	Exhaustividad	Ratio de éxito
SSD MobileNet FPN LITE	94,78 %	94,02 %	89,39 %
SSD MobileNet	95,45 %	92,03 %	88,17 %
Efficient D0	96,36 %	94,82 %	91,54 %
YOLOv4	72,97 %	99,21 %	72,54 %
YOLOv5	93,41 %	96,79 %	90,60 %
YOLOv7	95,38 %	98,80 %	94,30 %

En general, los resultados del experimento indican que los mejores modelos son YOLOv7, Efficient D0 y SSD MobileNet, considerando las tres métricas. No obstante, la elección de ellos dependió de las necesidades específicas de la investigación, tomando en cuenta únicamente los que se basan en la arquitectura YOLO para los siguientes experimentos. Esto se debe a que, aunque los modelos SSD MobileNet FPN LITE, SSD MobileNet y Efficient D0 son populares y ampliamente utilizados en la detección de objetos, pueden presentar algunas limitaciones en términos de precisión y capacidad para detectar vehículos pequeños en movimiento. Tal como se observa en la Figura 9.

En contraste, YOLO se destaca como una excelente opción cuando se busca combinar una buena precisión con una alta velocidad de inferencia en aplicaciones en tiempo real, puesto que utiliza un enfoque de detección de objetos basado en una

sola pasada por la red, lo que le permite ser más rápido que otros modelos de detección.

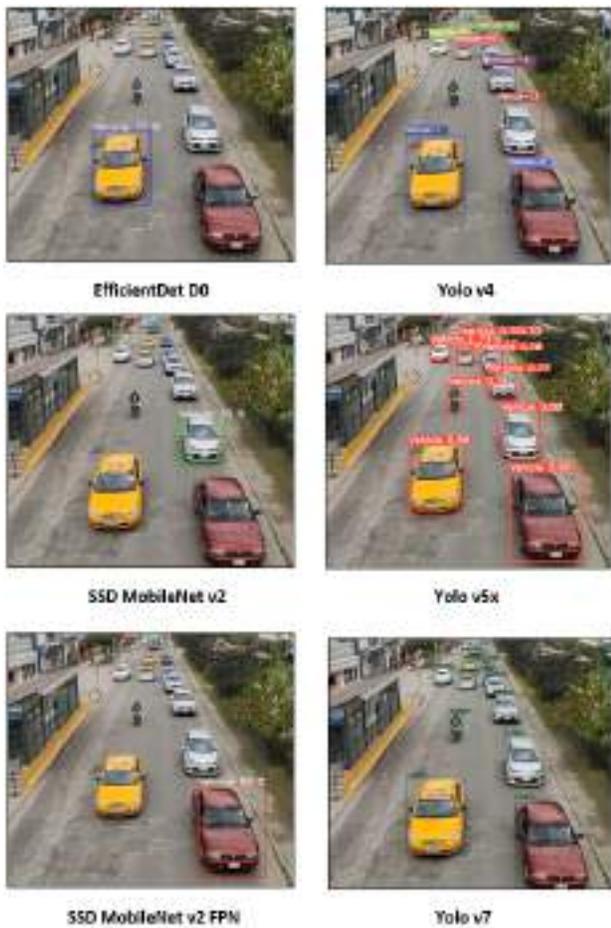


Figura 9. Detección de los modelos en horario diurno.

B. Experimento 2. Medición de precisión, exhaustividad y ratio de éxito de los modelos en horario nocturno.

En este experimento, se busca evaluar la precisión, exhaustividad y ratio de éxito de los tres modelos basados en la arquitectura YOLO en condiciones de baja luminosidad (horario nocturno), los resultados se muestran la Tabla III.

Tabla III
RESULTADOS DE PRECISIÓN, EXHAUSTIVIDAD Y RATIO DE ÉXITO DE CADA MODELO EN HORARIO NOCTURNO

Modelos	Precisión	Exhaustividad	Ratio de éxito
YOLOv4	82,27 %	88,16 %	74,08 %
YOLOv5	92,79 %	87,89 %	82,27 %
YOLOv7	72,38 %	100,00 %	72,38 %

En general, los modelos YOLOv5 y YOLOv7 muestran una disminución en el resultado de las métricas, sin embargo, el modelo YOLOv4 presenta una mejora de casi un 10%. A pesar de esto, el mejor modelo tomando como referencia la precisión y ratio de éxito es YOLOv5. Esto se evidencia en la Figura 10.



Figura 10. Detección de los modelos en horario nocturno.

C. Experimento 3. Evaluación de las métricas del flujo vehicular en horario diurno.

Basándonos en los resultados de los experimentos anteriores, se seleccionó exclusivamente YOLOv7 debido a que mostró el mejor rendimiento en condiciones de alta luminosidad. Por lo tanto, el objetivo de este experimento es evaluar el cálculo de las tres magnitudes del flujo vehicular en horario diurno.

Para la evaluación de la densidad se seleccionó cuatro momentos de cada video, es decir, se extrae un frame del video cada 20 segundos para el conteo de vehículos en el ROI. Por otro lado, para la evaluación de la intensidad, se registra el total de vehículos que pasan la línea de intensidad en el ROI en cada video. Finalmente, para el presente experimento se tomó en cuenta la velocidad promedio en cada video. Los resultados se evidencian en la Tabla IV.

Tabla IV
RESULTADOS DE LAS MAGNITUDES DEL FLUJO VEHICULAR EN HORARIO DIURNO

Videos	1	2	3	4	5	6	7	8	9	10
Precisión de la Intensidad	82 %	100 %	96 %	79 %	95 %	89 %	100 %	100 %	96 %	100 %
Precisión de la densidad	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %
Velocidad Promedio (Km/h)	39.87	39.87	42.50	43.50	42.68	42.32	43.52	42.21	41.43	43.10

El resultado encontrado fue que el cálculo de la densidad durante el día es preciso, mientras que la intensidad presenta ciertas fallas, esto se debe al algoritmo *StrongSort*, que a pesar de ser efectivo en algunos casos de seguimiento, identifica

nuevamente un vehículo previamente etiquetado al momento de realizar el *tracking*, asignando hasta dos etiquetas a un mismo vehículo, tal como se observa en la Figura 11.

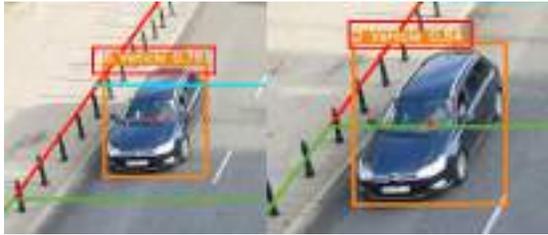


Figura 11. Fallo de StrongSort al realizar el seguimiento.

En general, el rendimiento del modelo YOLOv7 es más que aceptable. Tal como se observa en la Figura 12 el cálculo de la densidad es perfecto, y la intensidad tiene un 94% de acierto. No obstante, es importante señalar que no fue posible verificar el porcentaje de acierto de la velocidad debido a la falta de recursos necesarios para obtener la velocidad real y contrastar los valores, como pueden ser sensores físicos similares a los usados por los fotorradares.

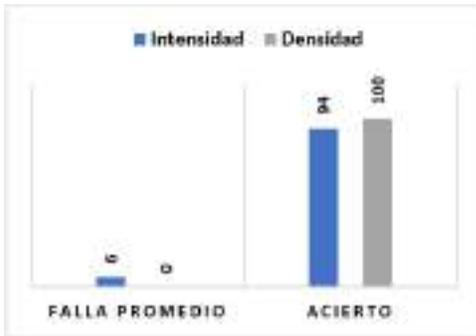


Figura 12. Resultados intensidad y densidad del modelo YOLOv7 en horario diurno.

D. Experimento 4. Evaluación de las métricas del flujo vehicular en horario nocturno.

El presente experimento es similar al anterior, con la diferencia que en este caso se usó el modelo YOLOv5, que fue escogido en el experimento 2. Los resultados del experimento se muestran en la Tabla V.

Tabla V

RESULTADOS DE LAS MAGNITUDES DEL FLUJO VEHICULAR EN HORARIO NOCTURNO

Videos	1	2	3	4	5	6	7	8	9	10
Precisión de la Intensidad	62 %	97 %	87 %	69 %	91 %	93 %	97 %	97 %	83 %	100 %
Precisión de la densidad	69 %	52 %	51 %	53 %	61 %	55 %	46 %	45 %	49 %	69 %

Además, cabe recalcar que no fue posible calcular la velocidad debido a que el modelo StrongSort pierde con facilidad al vehículo debido a las condiciones de luminosidad, dificultando la obtención de esta métrica.

En resumen, tal como se observa en la Figura 13, se determinó que la densidad en horario nocturno no alcanza una alta precisión debido a los problemas mencionados, presentando un porcentaje de acierto del 45%. De igual forma, en la intensidad el porcentaje de error alcanza el 13% en comparación al experimento anterior. Por lo cual, se concluye que el modelo de detección no es tan preciso y eficiente en condiciones de baja luminosidad.



Figura 13. Resultados intensidad y densidad del modelo YOLOv5 en horario nocturno.

E. Prototipo TraffSmart

Una vez finalizados los experimentos, se concluyó con él desarrolló de un prototipo software utilizando los mejores modelos de detección como YOLOv5 y YOLOv7. Este prototipo se caracteriza por realizar una eficaz evaluación del flujo vehicular, puesto que calcula las tres principales magnitudes como son intensidad, densidad y velocidad, tanto en videos pre grabados como en tiempo real, tal como se muestra en la Figura 14.



Figura 14. Prototipo TraffSmart.

V. CONCLUSIONES

En este trabajo se realizó una revisión de la literatura para examinar, analizar y comprender las arquitecturas de módulos de detección de objetos en el campo de la visión por computadora aplicada al tráfico vehicular. Durante la cual se encontró información acerca de varios modelos que son ampliamente utilizados para el análisis computacional de imágenes, entre ellos se destacan SSD MobileNet, EfficientNet y YOLO.

De los modelos analizados, se determinó que la arquitectura YOLO es ampliamente usada en la detección de objetos debido a su velocidad y precisión. Esto se respalda con la experimentación realizada específicamente con YOLOv5 y YOLOv7, que destacan por su capacidad para detectar vehículos en distintas condiciones de luminosidad del resto de arquitecturas.

Una vez determinados los mejores modelos para la detección de objetos, se procedió al cálculo de las principales magnitudes del flujo vehicular: densidad, intensidad y velocidad. Para ello se emplearon técnicas de análisis de imágenes y seguimiento de objetos, logrando resultados sobresalientes en horario diurno, con un 100% de precisión en el cálculo de la densidad y un 94% para la intensidad. Por otro lado, en el horario nocturno, se observa un decrecimiento considerable en el cálculo de la densidad e intensidad, obteniendo un 45% y 87% respectivamente.

Sin embargo, aunque los modelos utilizados obtuvieron resultados sobresalientes, se registró que los factores de iluminación y el algoritmo de seguimiento implementado afectan a la efectividad de detección y seguimiento de objetos. En condiciones de baja luminosidad, se observa un decrecimiento importante en el cálculo de la densidad y la intensidad. Por tal motivo, para mejorar la detección en diversas condiciones de luminosidad, es recomendable emplear cámaras infrarrojas o de visión nocturna.

Otro punto a tomar en cuenta es que los experimentos se realizaron con las tarjetas gráficas NVIDIA GeForce 1050 Ti y 1650, logrando hasta 30 FPS en la detección con YOLOv5 y YOLOv7, con un tiempo promedio de procesamiento de 2 a 3 minutos por video. Por lo tanto, se recomienda usar tarjetas gráficas como la RTX 3060 en adelante, junto con un procesador Intel Core i7 y al menos 16 GB de RAM o NVIDIA Jetson AGX Orin 32 GB, si se requiere realizar un análisis efectivo en tiempo real entre 60 y 120 FPS.

En conclusión, luego de analizar los modelos de detección y el algoritmo de seguimiento, se desarrolló un prototipo denominado TraffiSmart, desarrollado con técnicas avanzadas de visión por computadora para evaluar las magnitudes del flujo vehicular en tiempo real durante el horario diurno. Por lo cual, TraffiSmart se presenta como una opción altamente recomendable para mejorar la gestión del tráfico vehicular en ciudades con alta congestión.

REFERENCES

- [1] H. Ritchie, "Urbanization," 6 2018. [Online]. Available: <https://ourworldindata.org/urbanization>
- [2] Statista, "América Latina: peores embotellamientos por ciudad 2021," 3 2023. [Online]. Available: <https://es.statista.com/estadisticas/1088889/ciudades-mas-embotellamientos-america-latina/>
- [3] L. Barba-Guaman, J. Eugenio Naranjo, and A. Ortiz, "Deep learning framework for vehicle and pedestrian detection in rural roads on an embedded gpu," *Electronics*, vol. 9, no. 4, p. 589, 2020.
- [4] C.-J. Lin and J.-Y. Jhang, "Intelligent traffic-monitoring system based on yolo and convolutional fuzzy neural networks," *IEEE Access*, vol. 10, pp. 14 120–14 133, 2022.
- [5] H. Yang, Y. Zhang, Y. Zhang, H. Meng, S. Li, and X. Dai, "A fast vehicle counting and traffic volume estimation method based on convolutional neural network," *IEEE Access*, vol. 9, pp. 150 522–150 531, 2021.
- [6] M. Tan and A. Yu, "Efficientdet: Towards scalable and efficient object detection – google research blog," 2020. [Online]. Available: <https://ai.googleblog.com/2020/04/efficientdet-towards-scalable-and.html>
- [7] M. Singhal, "Object Detection using SSD Mobilenet and Tensorflow Object Detection API : Can detect any single class from coco dataset." 6 2023. [Online]. Available: <https://medium.com/@techmayank2000/object-detection-using-ssd-mobilenetv2-using-tensorflow-api-can-detect-any-single-class-from-31a31bbd0691>
- [8] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015. [Online]. Available: <http://arxiv.org/abs/1506.02640>
- [9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020.
- [10] Y. Yang, L. Cai, H. Wei, T. Qian, and Z. Gao, "Research on traffic flow detection based on yolo v4," *ICCSE 2021 - IEEE 16th International Conference on Computer Science and Education*, pp. 475–480, 8 2021.
- [11] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, K. Michael, TaoXie, J. Fang, imyhxy, Lorna, Z. Yifu, C. Wong, A. V. D. Montes, Z. Wang, C. Fati, J. Nadar, Laughing, UnglvKitDe, V. Sonck, tkianai, yxNONG, P. Skalski, A. Hogan, D. Nair, M. Strobel, and M. Jain, "ultralytics/yolov5: v7.0 - yolov5 sota realtime instance segmentation," 11 2022. [Online]. Available: <https://zenodo.org/record/7347926>
- [12] J. Nelson, "Responding to the Controversy about YOLOv5," *Roboflow Blog*, 3 2021. [Online]. Available: <https://blog.roboflow.com/yolov4-versus-yolov5/>
- [13] I. Idrisov, PhD, "YOLOv4 vs YOLOv5 - Deelvin Machine Learning - Medium," 12 2021. [Online]. Available: <https://medium.com/deelvin-machine-learning/yolov4-vs-yolov5-db1e0ac7962b>
- [14] Y. Song, Z. Xie, X. Wang, and Y. Zou, "Ms-yolo: Object detection based on yolov5 optimized fusion millimeter-wave radar and machine vision," *IEEE Sensors Journal*, vol. 22, no. 15, pp. 15 435–15 447, 2022.
- [15] Y. Zhang, Z. Guo, J. Wu, Y. Tian, H. Tang, and X. Guo, "Real-Time Vehicle Detection Based on Improved YOLO v5," *Sustainability*, vol. 14, no. 19, p. 12274, 9 2022. [Online]. Available: <https://doi.org/10.3390/su141912274>
- [16] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022.
- [17] G. Boesch, "YOLOv7: The Most Powerful Object Detection Algorithm (2023 Guide)," *viso.ai*, 6 2023. [Online]. Available: <https://viso.ai/deep-learning/yolov7-guide/>
- [18] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng, "Strongsort: Make deepsort great again," 2023.
- [19] "Open Images V7." [Online]. Available: <https://storage.googleapis.com/openimages/web/index.html>
- [20] "Roboflow: Give your software the power to see objects in images and video." [Online]. Available: <https://roboflow.com/>
- [21] Google, "Google Colaboratory." [Online]. Available: <https://colab.research.google.com/>
- [22] P. Dendorfer, H. Rezatofghi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "Mot20: A benchmark for multi object tracking in crowded scenes," 2020.
- [23] E. Ristani, F. Solera, R. S. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," 2016.
- [24] J. Luiten, A. Osep, P. Dendorfer, P. H. S. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "HOTA: A Higher Order Metric for Evaluating Multi-object Tracking," *International Journal of Computer Vision*, vol. 129, no. 2, pp. 548–578, 2 2021. [Online]. Available: <https://doi.org/10.1007/s11263-020-01375-2>
- [25] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," 2017.
- [26] M. Espitia and J. Páez, "Sistema de captación automático de datos para evaluar las condiciones de tráfico vehicular en vías," *Universidad Distrital Francisco José de Caldas*, 2019. [Online]. Available: <http://hdl.handle.net/11349/22508>
- [27] G. López, A. M. Pérez, and F. J. Camacho, "Variables fundamentales del tráfico: Intensidad, velocidad y densidad apellidos, nombre," *RiuNet*, 2019. [Online]. Available: <https://bit.ly/3W8Ywn3>