

Sistema Inteligente de Gestión de Semáforos en Tiempo Real

Yim Isaias Apestequi Florentino¹, Hugo Esteban Caselli Gismondi¹

yapestequi@uns.edu.pe; hcasellig@uns.edu.pe

¹ Universidad Nacional del Santa, Av. Pacífico 508, 02711, Nuevo Chimbote, Perú.

DOI: 10.17013/risti.54.69–87

Resumen: La gestión del tráfico vehicular es un desafío importante en las ciudades del mundo, pues involucra a la gestión de semáforos y el flujo de tráfico. El problema específico es la configuración de semáforos con asignaciones de tiempos fijos, no acorde al tiempo requerido por el flujo de tráfico real. El objetivo de este trabajo es proponer un sistema inteligente para gestionar en tiempo real el semáforo en una intersección vial; el cual se ajusta a las condiciones reales del tráfico que son la velocidad y densidad vehicular. Para ello, se desarrolló un algoritmo heurístico que toma en cuenta las características del tráfico y luego toma decisiones para asignar el tiempo de fase correspondiente. Pruebas experimentales con 15000 escenarios simulados permitieron estimar el tiempo de espera entre ciclos semaforicos. Los resultados mostraron una reducción del 18.3% de tiempo del ciclo con una mejora de 47 ciclos/hora respecto a los 39 ciclos/hora de un semáforo tradicional con tiempos de fase fijos.

Palabras-clave: Semáforo inteligente; flujo vehicular; algoritmo heurístico; simulación.

Intelligent Traffic Light Management System in Real Time

Abstract: The management of vehicular traffic poses a significant challenge in cities worldwide, as it involves the control of traffic lights and traffic flow. The specific issue lies in the configuration of traffic lights with fixed-time allocations, not in line with the time required by real traffic flow. The aim of this study is to propose an intelligent system for real-time management of traffic lights at an intersection, which adjusts to the actual traffic conditions, namely speed and vehicle density. To achieve this, a heuristic algorithm was developed, considering traffic characteristics and making decisions to allocate the corresponding phase time. Experimental tests with 15,000 simulated scenarios allowed for the estimation of waiting time between signal cycles. The results demonstrated an 18.3% reduction in cycle time with an improvement of 47 cycles/hour compared to the 39 cycles/hour of a traditional traffic light with fixed phase times.

Keywords: Smart traffic light; vehicular flow; heuristic algorithm; simulation.

1. Introducción

La gestión del tráfico vehicular se ha convertido en un gran desafío en ciudades del mundo con expansión demográfica, pues a mayor crecimiento poblacional, mayor demanda de transporte. Saus (2023) menciona que los patrones de expansión de las ciudades tienen un impacto en la efectividad del transporte público. En ese sentido, los gobiernos locales suelen descuidar la planificación del transporte, en medio de una creciente cantidad de vehículos (Ministerio de Seguridad Pública de China, 2023) y la inadecuada gestión del tráfico. Las emisiones de dióxido de carbono (CO₂) del parque automotor son una situación preocupante, sabiendo que el 90% de las emisiones de CO₂ están contribuyendo al cambio climático (Nikas et al., 2022). En cuanto al control de las emisiones de CO₂, en países de Latinoamérica como Colombia, Ecuador y Perú, las sugerencias de organismos internacionales de reemplazar vehículos que utilizan combustibles fósiles por vehículos eléctricos no se pueden implementar fácilmente (OCDE, 2011).

Respecto a la gestión del tráfico, los planes de gestión siguen perdiendo eficacia. Camargo (2018) menciona que el modelo “pico y placa” implementado en la ciudad de Bogotá, Colombia, en el año 2002, no ha mejorado el problema de la congestión vehicular. En el año 2002, había 500,000 automóviles registrados; para el año 2011, este número aumentó a 1,400,000 automóviles particulares. Caso similar sucedió en la ciudad de Lima, Perú; en el año 2019 se implementó el modelo “pico y placa”, únicamente en ciertas vías principales de alta congestión. Como resultado, las vías alternas y alimentadoras resultaron sumamente congestionadas. A la fecha, en Perú, el modelo es inviable y se ha dejado de usar; en Colombia, el modelo continúa en uso, ya que “pico y placa” les es necesario debido a obras de infraestructura programadas para el 2024 y que afectarán el tráfico en la capital colombiana.

La principal motivación del trabajo propuesto es la necesidad de contribuir con la optimización del tráfico vehicular mediante una gestión inteligente de semáforos. Manrique (2018) identifica que una de las causas principales del tráfico vehicular está relacionado con la gestión de semáforos pues presentan problemas específicos como: luces dañadas, visibilidad reducida para los vehículos y tiempos fijos asignados a los colores de semáforo. Un tiempo de fase inadecuado permite la ocurrencia de accidentes y tiempos de espera perdidos. En consecuencia, el tiempo del ciclo del semáforo representa una variable importante a considerar con el fin de mejorar la productividad y seguridad en las intersecciones viales (Araya-Porras et al., 2022).

En el Perú se ha observado que los semáforos con configuraciones de tiempos de fase fijos o predefinidos son los que más se han instalado. Los tiempos de luz verde son inadecuados. Tramos de la vía que están vacíos tienen luz verde; mientras tanto, hay carriles con vehículos esperando en rojo en la vía transversal. Al respecto, diversos estudios han abordado el tema de la asignación de tiempos de ciclo del semáforo, usando algoritmos que respondan a la realidad del entorno. Para ello, han recurrido a la Inteligencia Artificial (IA), para desarrollar teorías, algoritmos, métodos, tecnologías y sistemas de aplicación (Huawei Technologies Co., Ltd., 2023). Las técnicas más usadas son las de optimización, las de gestión de sensores y técnicas de aprendizaje automático (Tomar et al., 2022).

Este trabajo tiene como objetivo proponer un sistema inteligente de gestión de semáforos en tiempo real. Las principales contribuciones de este trabajo son la validación de los métodos de conteo vehicular basados en YOLOv4 en cuanto a niveles de precisión, la precisión de medición de velocidad vehicular basada en el Método de Detección de Intrusiones (IDM) y el aporte significativo del algoritmo heurístico, que en su conjunto forman el sistema inteligente, reduciendo el tiempo de espera de los vehículos en contraste con la configuración predefinida de los semáforos tradicionales. Los resultados encontrados servirán a otros investigadores para: 1) diseñar sistemas escalables y sincronizados para varias intersecciones viales bajo el enfoque de tiempo real y 2) crear componentes de software embebidos con el algoritmo heurístico para ser utilizados con transformadores o redes LSTM.

El presente artículo está estructurado en las siguientes partes: inicia con una revisión bibliográfica del tema, seguido por la descripción de los materiales y métodos aplicados. Luego, se describe la sección de resultados y discusión, donde se analizan los resultados obtenidos de la aplicación de algoritmos en términos de precisión. Finalmente, la sección de conclusiones y trabajos futuros.

2. Revisión bibliográfica

De los trabajos más recientes relacionados con el tráfico vehicular, un grupo está enfocado en la detección de carreteras, otro en la detección y seguimiento de vehículos, y otro grupo en los sistemas de transporte inteligente (ITS). Wang et al. (2019) sugieren que es posible monitorear secciones de carreteras para predecir el tráfico con técnicas como el modelado de datos y un modelo de predicción. Usaron una secuencia de matrices para entrenar un modelo profundo que combina una red neuronal convolucional (CNN) de aplicación específica para clasificación y segmentación de imágenes, según Jiménez-Murillo et al. (2023), y una red neuronal recurrente (RNN). Fundamentalmente, entrenaron un modelo profundo para predecir el flujo de tráfico y controlar las luces de tráfico con el fin de reducir la congestión.

Se han empleado técnicas de sustracción de fondo junto con métodos de análisis de contraste en distintas regiones de interés (ROI). Las métricas fueron: el número de vehículos en espera cuando un semáforo cambia de rojo a verde y la cantidad de vehículos que atraviesan la intersección, justo antes de que el semáforo pase de verde a rojo. El método se denomina Procesamiento de Textura de un conjunto de regiones de interés (Inglés-Romero et al., 2018).

Ashifuddin et al. (2019) desarrollaron un ITS con un enfoque de Internet de las cosas (IoT) para administrar la congestión del tráfico. Se utilizaron sensores estacionarios en la carretera para estimar y categorizar el nivel de congestión del tráfico en distintos tramos viales dentro de una ciudad, así como para analizar los datos de tráfico capturados como velocidad y densidad. Emplearon una red neuronal artificial (ANN) como base del sistema para clasificar los siguientes niveles de congestión: flujo libre, congestión media y congestión alta.

Actualmente, un sistema inteligente administrador de tráfico está conformado de componentes como: a) Adquisición de imágenes que, a través de sensores de

cámara, captura datos de la realidad, b) Extracción de características en función del reconocimiento, detección y seguimiento de vehículos y c) Aplicaciones de software de tráfico que suelen utilizar sistemas de posicionamiento global, sensores de tráfico y datos de tráfico en tiempo real. Tendencias actuales de ITS están basadas en metaheurísticas, lógica difusa, redes neuronales convolucionales, aprendizaje por refuerzo y una combinación de modelos y sistemas llamados híbridos (Nigam et al., 2023).

Para la adquisición de imágenes en tiempo real, el algoritmo que ha respondido de manera muy eficiente en procesos de detección es You Only Look Once (YOLO). A la fecha, es el estado de arte para la detección de objetos (Gonçalves et al., 2023). Los algoritmos previos a YOLO para la detección de objetos utilizan R-CNN o Fast R-CNN, procedimientos que son robustos y precisos, pero son procesos complejos, lentos y poco optimizables debido a que cada componente debe entrenarse por separado (Redmon et al., 2016). YOLO solo usa regresión con la red neuronal, debido a ello su rapidez. Zuraimi & Zaman (2021) evaluaron YOLOv4 en comparación con su versión previa en un sistema de detección de vehículos; según los resultados, se alcanzó una precisión del 82,08% en tiempo real de aproximadamente 14 fps en una tarjeta gráfica GTX 1660ti.

Respecto a la medición de la velocidad de objetos basada en video, se han encontrado estudios que utilizan métodos de sustracción del objeto en movimiento. Al objeto se considera como una mancha dentro de una elipse delimitadora, la cual se va ubicando en cuadros consecutivos a través de intervalos de tiempo (Cheung & Kamath, 2005). La técnica se ha mejorado con la técnica de estimación del fondo y seguimiento del movimiento mediante la segmentación del primer plano (Jeyabharathi & Dejeey, 2016).

Javadi et al. (2019) proponen el Método de Detección de Intrusiones (IDM), que es un método empleado para identificar cuando un objeto atraviesa una línea virtual y entra en una zona de interés entre puntos discretos. El intervalo entre los puntos discretos detectados corresponde al tiempo de muestreo de la cámara. Por consiguiente, la intrusión es detectada en una variación de distancia de detección $\Delta(m)$. Esta distancia de detección está directamente relacionada con el tiempo de muestreo de la cámara $t(s)$ y la velocidad hipotética del vehículo expresada como $v(m/s)$.

3. Materiales y métodos

La Figura 1 muestra la arquitectura del Sistema de Gestión de Semáforos (SGS) para intersecciones viales con giro a la izquierda. Las cámaras capturan el tráfico en la intersección. El componente de detección y seguimiento clasifica y cuenta los vehículos con el algoritmo YOLO y determina su velocidad usando un método de detección de intrusiones. Estos datos alimentan el algoritmo heurístico de asignación de tiempos de fase. Además, los datos se almacenan en un servidor Kafka para aplicaciones cliente en tiempo real.

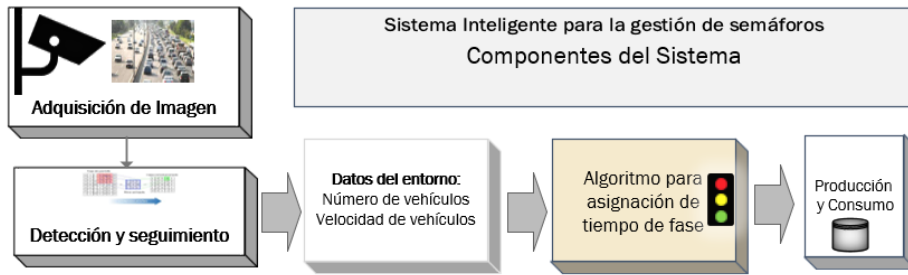


Figura 1 – Arquitectura de componentes del Sistema de Gestión de Semáforos

3.1. Adquisición de imágenes

Respecto al proceso de adquisición de imágenes, es importante mencionar que en cumplimiento a la ley de protección de los datos en el Perú, el uso de las imágenes capturadas ha sido y será usado estrictamente en el contexto de la presente investigación. El detector de objetos no detecta la placa de vehículo; clasifica a una persona, pero no usa reconocimiento facial. Por tanto, la privacidad de los transeúntes, los vehículos y las edificaciones está garantizada. Esta fase se realizó en una intersección de alta congestión vehicular durante la hora pico (7 a.m. hasta las 8 a.m.), en período escolar y con giro a la izquierda permitido (Av. Pacífico - Av. Country en Nuevo Chimbote, Perú). Se ensambló un equipo tal como se ilustra en la Figura 2. Se utilizó un trípode de 6 m de largo, las cámaras de dispositivos móviles con resolución de 2 megapíxeles con aspecto 4:3 y un ángulo de inclinación θ del equipo 4 entre 30 y 60 grados, calibrados de acuerdo a la distancia del tramo y el pase peatonal. Se instaló el software Open Camera y TeamViewer en los equipos 4 y 5, para controlar y monitorear con el equipo 5 al equipo 4.

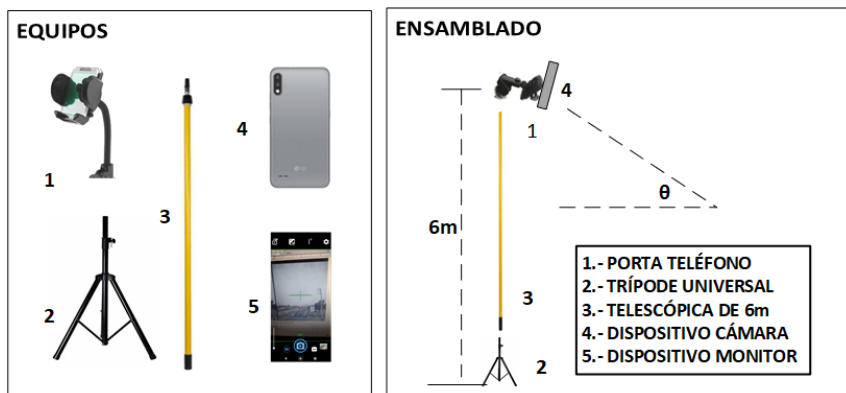


Figura 2 – Ensamblado de equipos para la adquisición de imágenes

De otra parte, antes de medir las velocidades de vehículos, se midió la longitud de las líneas discontinuas del tramo vehicular, seguido por la medición del espaciado de segmentos que es la distancia entre estas líneas, expresada en metros, ver Figura 3.

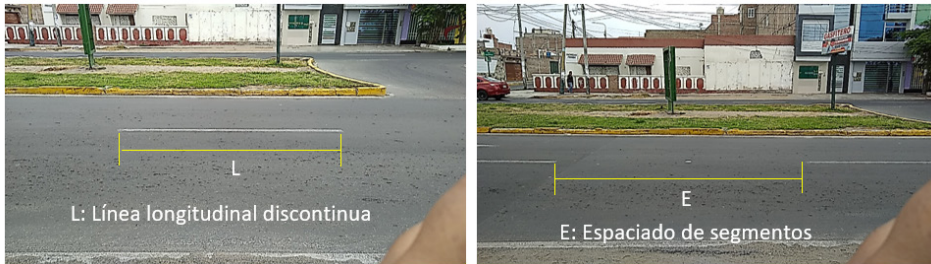


Figura 3 – Línea longitudinal y espaciado de segmentos de tramo

Se grabaron videos e imágenes para la comparación respectiva de los datos de la recolección manual con los datos que arrojen los métodos YOLO e IDM. Para los cálculos se consideró las medidas de línea longitudinal discontinua y espaciado de segmentos que son respectivamente de 3 m y 5 m para zonas urbanas de velocidad ≤ 60 km/h y de 4.5 m y 7.5 m para zonas de mayor velocidad (Ministerio de Transportes y Comunicaciones del Perú, 2018, p. 388).

3.2. Detección y seguimiento

Se utilizó un modelo preentrenado Yolov4 aplicado a una base de datos de 120 imágenes de intersección vehicular capturadas en la intersección descrita en la sección 3.1. No fue necesario proceder al etiquetado de imágenes con LabelImg porque, más que clasificación, el enfoque estuvo en detección. Se modificó el archivo yolov4.cfg, para solamente permitir detectar vehículos como camiones, bicicletas, automóviles, buses, motocicletas, entre otros, a fin de discriminar el conteo exclusivo de vehículos. Luego, para la parte experimental de conteo de las 120 imágenes de los tramos de la intersección vial, de manera aleatoria, con un intervalo de confianza de 95%, se consideró una muestra de 96 imágenes. En la Figura 4, se pueden observar tres escenarios de detección mediante algoritmo. En cuanto al proceso de seguimiento, se consideró de manera aleatoria simple 16 vehículos, durante un período de tiempo. Para llevar a cabo la medición manual de la velocidad, según el método IDM, se calculó el tiempo que un vehículo toma en recorrer una cierta distancia Δ (m), en este caso la distancia $d = 7.5$ m +2 m de tolerancia. Con un cronómetro digital se registró el momento en que el vehículo inició la intrusión de la distancia d hasta finalizar el recorrido d . El tiempo t en segundos obtenido de manera manual se emplea para determinar la velocidad en km/h mediante la siguiente fórmula: velocidad $v = (9.5 \text{ m}/t) \times (3600 \text{ s}/\text{h}) \times (\text{km}/1000 \text{ m})$.

3.3. Respecto al flujo de tráfico

El flujo de tráfico está determinado por la ecuación $Q = Vm \times D$, donde Q es el flujo de tráfico, Vm es la velocidad media y D es la densidad vehicular. Esto significa que Q varía

en función de los valores de V_m y D . En la Figura 5 se puede observar que si la densidad vehicular disminuye, la velocidad media crece y si la densidad vehicular aumenta, la velocidad media disminuye. Esto tiene su implicancia en la determinación empírica del tipo de congestión basada en la densidad.



Figura 4 – Detección y conteo de vehículos en tiempo real

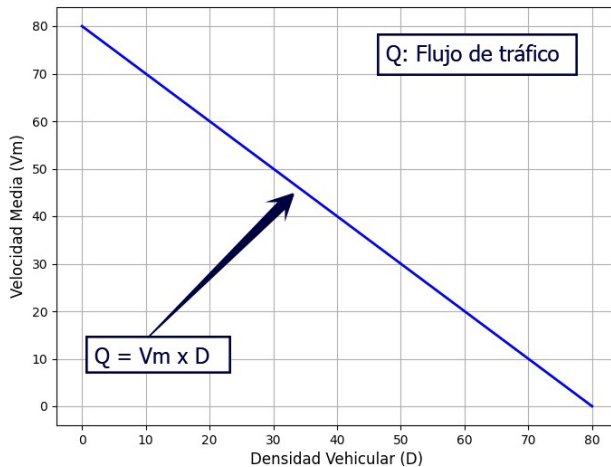


Figura 5 – Relación velocidad y densidad vehicular

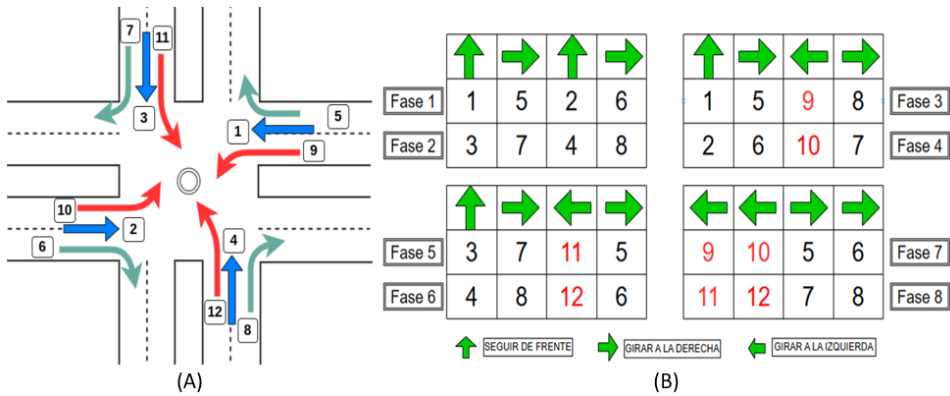


Figura 6 – Flujos conflictivos y no conflictivos en una intersección vial

En la Figura 6(A) se observa que, cuando entra en fase el par de flujos (1, 2), estos no presentarán conflictos entre sí. De manera similar sucede con el par de flujos (3, 4). El par de flujos (5, 6) y (7, 8) giran a la derecha tampoco presentan conflictos entre sí. Los flujos conflictivos son los flujos que giran a la izquierda como los pares de flujo (9, 2), (10, 1), (11, 4) y (12, 3). En la Figura 6(B) se observan 8 fases menos conflictivas conformadas por 4 flujos y se evidencia que cada vez que aparece un giro a la izquierda, la fase siempre presenta un flujo conflictivo que no afecta a los demás. Por ejemplo, en la fase 6 los flujos 4, 8 y 6 no entran en conflicto y el flujo conflictivo 12 tampoco los afecta. Por tanto, en un escenario en el que el tráfico entra en bloqueo mutuo, una de estas 8 posibilidades de fase podrá ayudar a descongestionar el tráfico.

3.4. Algoritmo para flujos no conflictivos (FNC)

La Tabla 1 muestra 3 tipos de modos: MODO 1 para condiciones de congestión de rutina, MODO 2 para congestión extrema y velocidad promedio cerca de cero (Prohibición de Giro a la Izquierda) y MODO 3, solo si resulta imposible que los flujos de tráfico lineales crucen la intersección de frente.

PAR	Par1	Par2	Par3	Par4	Par5	Par6	Par7	Par8
Flujos	(9,10)	(9,1)	(2,10)	(2,1)	(12,11)	(12,4)	(3,11)	(3,4)
MODO	3	1	1	2	3	1	1	2
PAR	NO	Par3	Par2	NO	NO	Par7	Par6	NO

Tabla 1 – Pares de flujos no conflictivos en una intersección vial

Algoritmo 1: Función FNC

1. Crear una estructura FNC [Flujo A, Flujo B, MODO, Par] donde contenga la matriz de la Fig.7:
2. Si MODO = 1 entonces, Avance de frente = True, Giro de Izquierda = True

Algoritmo 1: Función FNC

3. Si MODO = 2 entonces, Avance de frente = True, Giro de Izquierda = False
4. Si MODO = 3 entonces, Avance de frente = False, Giro de Izquierda = True

3.5. Algoritmo de inicio y configuración

En este algoritmo establece los datos de entrada específicos de cada tramo vehicular; como son el largo, ancho, número de carriles, espaciado entre vehículos y realiza los cálculos descritos en los ítems 4, 5, 6, 7 y 8 del algoritmo 2.

Algoritmo 2: Función Iniciar

1. Leer longitud (L), ancho (A) y número de carriles (Lane) de cada tramo.
2. Leer espacio longitudinal (EspaLong) y espacio lateral (EspaLate) entre vehículos.
3. Leer velocidad máxima (Vmax) y velocidad mínima (Vmin) permitida en el tramo.
4. Calcular el área de cada tramo (At) de intersección.

$$At[i] = longitud[i] \times ancho[i]$$
5. Calcular el área ocupada por vehículo (Av).

$$Av = EspaLong \times Espalate$$
6. Calcular el máximo número de vehículos por tramo (MaxVeh).

$$MaxVeh[i] = L[i] / EspaLong \times Lane[i]$$
7. Calcular la densidad máxima del tramo.

$$Dmax[i] = MaxVeh[i] / At[i]$$
8. Calcular el Tiempo máximo (Tmax) del tramo.

$$Tmax = L / Vmin$$

3.6. Algoritmo de operaciones con el número de vehículos y densidad

Este algoritmo calcula la Densidad de Grupo (dg) dividiendo el número de vehículos por el área del tramo. Luego, determina el Índice de Congestión de Tráfico (ICT), que es dg dividido por la densidad máxima del tramo, resultando en un valor en el rango de [0, 1]. Un ICT cercano a 0 indica baja congestión, mientras que cerca de 1 indica alta congestión.

Algoritmo 3: Operaciones con el número de vehículos y densidad

1. Obtener número de vehículos del tramo (numVeh)

$$numVeh[i] = procesosLecturaCamara(i) // \text{Aplica YOLO}$$
2. Calcular la densidad de grupo (dg)

$$dg[i] = numVeh[i] / At[i]$$
3. Calcular el índice de congestión de tráfico (ICT)

$$ICT[i] = dg[i] / Dmax[i]$$

3.7. Algoritmo de operaciones con la velocidad vehicular

Este algoritmo obtiene la suma de las velocidades de los automóviles en aplicación del método de seguimiento IDM. Por otra parte, si el número de vehículos del tramo es cero, entonces la velocidad media será cero; sino, la velocidad media del grupo será la suma de las velocidades del grupo dividido entre el número de vehículos del tramo. Luego, con la velocidad media del grupo (vmg) calculada, se tipifica el tipo de velocidad que puede ser baja, moderada, alta y muy alta.

Algoritmo 4: Operaciones con la velocidad vehicular

1. Obtener la suma de velocidades de los vehículos del grupo (svg)

$$\text{svg}[i] = \text{procesosVelocidadCamara}(i) // \text{Aplica IDM}$$
2. Calcular la velocidad media del grupo (vmg)

$$\text{Si numVeh}[i] = 0, \text{ entonces } \text{vmg}[i] = 0$$

$$\text{Sino } \text{vmg}[i] = \text{sumv}[i] / \text{numVeh}[i]$$
3. Establecer el tipo de velocidad media (tvm)

$$j=0$$

Mientras $j \leq 3$:

$$\text{Si } \text{vmg}[j] \geq 0 \text{ and } \text{vmg}[j] < 10, \text{ entonces}$$

$$\text{tvm}[j][0] = 1 \text{ y } \text{tvm}[j][1] = \text{“Vel. Baja”}$$

$$\text{Sino } \text{vmg}[j] \geq 10 \text{ and } \text{vmg}[j] < 30, \text{ entonces}$$

$$\text{tvm}[j][0] = 2 \text{ y } \text{tvm}[j][1] = \text{“Vel. Moderada”}$$

$$\text{Sino } \text{vmg}[j] \geq 30 \text{ and } \text{vmg}[j] < 50, \text{ entonces}$$

$$\text{tvm}[j][0] = 3 \text{ y } \text{tvm}[j][1] = \text{“Vel. Alta”}$$

$$\text{Sino } \text{tvm}[j][0] = 4, \text{ entonces}$$

$$\text{tvm}[j][1] = \text{“Vel. muy Alta”}$$

$$j += 1$$

fin-mientras

3.8. Algoritmo para establecer el tipo de congestión

Este algoritmo se encarga de verificar en qué rango se encuentra el ICT del tramo actual y así establecer el tiempo de fase al tipo de congestión. El tiempo de fase T_{fase} se determina con la ecuación $T_{fase} = L/vmg$, donde L es la longitud del tramo y vmg es la velocidad media del grupo. Lo cual indica que a menor velocidad media, mayor tiempo de fase y a mayor velocidad media, menor tiempo de fase. Considerando $L=100 \text{ m}$ y $Vmg=Vmin=7.2 \text{ km/h}$, según observación empírica, se ha obtenido la representación matemática que se muestra en la Ecuación 1, donde $f(x)$ es el tiempo asignado para cada tipo de congestión dado el factor x y el Índice de Congestión de Tráfico (ICT). Observar que, a medida que ICT crece, la función $f(x)$ incrementa su valor. $Tmax$ es obtenido del algoritmo 2.

$$f(x) = \begin{cases} T_{max} - 5(2x + 1) & \text{Si } x = 3 \wedge ICT \in [0, 0.25) \\ T_{max} - 5(2x + 1) & \text{Si } x = 2 \wedge ICT \in [0.25, 0.5) \\ T_{max} - 5(2x + 1) & \text{Si } x = 1 \wedge ICT \in [0.5, 0.75) \\ T_{max} - 5(2x + 1) & \text{Si } x = 0 \wedge ICT \in [0.75, 0.9) \\ T_{max} - 5(2x + 1) & \text{Si } x = -1 \wedge ICT \in [0.9, 1) \end{cases}$$

Se usó la estructura TC[Tipo-Congestión, descripción, T_{fase}].

Algoritmo 5: Tipo de congestión

$i=0$

Mientras $i \leq 3$

Si $ICT[i] > 0$ and $ICT[i] < 0.25$, entonces

$x = 3$, $tc[i][0] = 1$, $tc[i][1] =$ "Congestión muy Bajo", $tc[i][2] = T_{max} - 5(2x + 1)$

Sino $ICT[i] > 0.25$ and $ICT[i] < 0.5$, entonces

$x = 2$, $tc[i][0] = 2$, $tc[i][1] =$ "Congestión Bajo", $tc[i][2] = T_{max} - 5(2x + 1)$

Sino $ICT[i] > 0.5$ and $ICT[i] < 0.75$, entonces

$x = 1$, $tc[i][0] = 3$, $tc[i][1] =$ "Congestión Medio", $tc[i][2] = T_{max} - 5(2x + 1)$

Sino $ICT[i] > 0.75$ and $ICT[i] < 0.90$, entonces

$x = 0$, $tc[i][0] = 4$, $tc[i][1] =$ "Congestión Alto", $tc[i][2] = T_{max} - 5(2x + 1)$

Sino $x = -1$, $tc[i][0] = 5$, $tc[i][1] =$ "Congestión muy Alto", $tc[i][2] = T_{max} - 5(2x + 1)$

$i = i + 1$

fin-mientras

3.9. Algoritmo de gestión del semáforo

El algoritmo del semáforo constantemente está recibiendo información del entorno para ajustar la programación óptima de la fase del semáforo en el tramo actual. Siempre tiene en memoria la Fase Actual, el MODO, el ICT respectivo, el tipo de congestión tc , la velocidad media del grupo vmg y el tipo de velocidad media tvm . Además, monitoriza los tramos para determinar cuál tiene mayor ICT para asignar el tiempo de fase. Luego, dependiendo de la densidad y la velocidad media, realiza el reajuste automáticamente aumentando o disminuyendo el tiempo de la fase en segundos adicionales, según sea el caso. A continuación, inicia el conteo regresivo considerando los tiempos asignados y finalmente, efectúa el cambio de fase.

Algoritmo 6: Gestión del Semáforo

1. El sistema inicia su funcionamiento

FASE = 1

Mientras MODO = 1:

2- Monitorizar entorno

IndConTraf = IndiceCongestionTrafico() //ICT

Algoritmo 6: Gestión del Semáforo

```
TipoCon      = TipoCongestion() //tc
VelMedGr     = VelocidadMediaGrupo()//vmg
TipoVelMed   = TipoVelocidadMedia() //tvm
```

3- Monitorizar tramos

```
faseActual = FASE
Si faseActual == 1
    Si IndConTraf[0] > IndConTraf[1], entonces
        posMayorTraficFase = 0, tramo="Sur-Norte"
    Sino, posMayorTraficFase =1, tramo="Norte-Sur"
Si faseActual == 2
    Si IndConTraf[2] > IndConTraf[3], entonces
        posMayorTraficFase = 2, tramo="Este-Oeste"
    Sino, posMayorTraficFase = 3, tramo="Oeste-Este"
TipoTraf = TipoCon[posMayorTraficFase]
tinicio = time()
```

4. Reajuste: Incrementar tiempo de verde x Tipo de congestión

4.1- Si densidad es alta y velocidad promedio es muy baja, aumentar tiempo verde 3s

```
Si TipoCon[posMayorTraficFase][0] == 4 y TipoVelMed[posMayorTraficFase][0] == 1
    tiempoAdicional = 3
```

4.2- Si densidad es muy alta y velocidad promedio es muy baja, aumentar tiempo verde 5s

```
Sino TipoCon[posMayorTraficFase][0] == 5 y TipoVelMed[posMayorTraficFase][0] == 1
    tiempoAdicional = 5
```

4.3- Si densidad es muy baja y velocidad promedio es muy alta, reducir tiempo verde -5s

```
Sino TipoCon[posMayorTraficFase][0] == 1 y TipoVelMed[posMayorTraficFase][0] == 5
    tiempoAdicional = -5
```

4.4- Si densidad es baja y la velocidad promedio es muy alta, reducir el tiempo verde -5s

```
Sino TipoCon[posMayorTraficFase][0] == 2 y TipoVelMed[posMayorTraficFase][0] == 5:
    tiempoAdicional = -5
```

4.5- Se mantiene la duración de verde y no se agrega tiempoAdicional

```
Sino    tiempoAdicional = 0
```

5. Conteo regresivo

```
TiempoRegresivoVerde(TipoCon[posMayorTraficFase][2] + tiempoAdicional)
    tfinal = time()
    tiempoTranscurridoVerde(tfinal-tinicio)
    TiempoRegresivoAmarillo(3 segundos)
```

6. -Cambio de fase

```
Si FASE == 1, entonces FASE = 2
```

Algoritmo 6: Gestión del Semáforo

Sino FASE = 1

fin-mientras

3.10. Despliegue, producción y consumo

El sistema inteligente propuesto se basa en el modelo productor consumidor, ver Figura 7. El servidor central que ejecuta Linux Ubuntu 22.04 LTS, con memoria RAM de 8 GB, procesador Intel Core I7 1.9 GHz, GPU NVIDIA GeForce MX130, está formado por dos nodos principales. Un nodo que se encarga de la lógica de la Gestión del Semáforo y otro nodo Nodejs que se encarga de atender las peticiones web.

El artefacto SemáforoInteligente.py en Python 3.8 usa Yolo4 y OpenCV 4.8 para procesar datos del entorno, ejecutar un algoritmo heurístico, y enviar resultados al nodo semáforo. Este nodo muestra los tiempos de fase asignados y cuentas regresivas. ProductorKafka.py conecta y envía datos json a un servidor Kafka en la nube ConfluentKafka, incluyendo fecha, hora, tipo de tráfico, vehículos por tramo y tiempo de espera. El Servidor Kafka almacena datos en tiempo real en un tópico Intersección01 y maneja peticiones del programa ConsumidorKafka.js para la Aplicación Cliente.

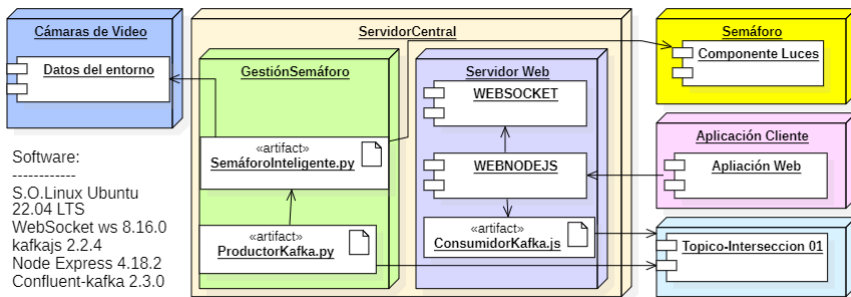


Figura 7 – Diagrama de despliegue para el sistema en tiempo real

4. Resultados

En esta sección se presentan los resultados de la validación del modelo de sistema inteligente propuesto, incluyendo el conteo de vehículos, la medición de velocidad y los resultados del algoritmo heurístico. Para evaluar la precisión del contador de vehículos basado en detección de objetos frente al conteo manual, se usó la fórmula $precisión = (algoritmo/observador)$, donde algoritmo es el número de vehículos detectados y observador es el número de vehículos del conteo manual. Se realizaron 96 conteos, obteniendo una precisión media de 0.9127, una mediana de 1 y una varianza de 0.016, con un nivel de confianza del 95%. Según la prueba de normalidad de Kolmogorv a 96 grados de libertad (gl), se obtuvo un p-valor < 0.05. Por consiguiente, al aplicar la prueba no paramétrica de Wilcoxon para una muestra, se obtiene una significancia de 0.083, o sea p-valor ≥ 0.05. Por tanto, se acepta que la mediana de *precisión* es de 0.95.

Respecto a la velocidad de los vehículos basada en el seguimiento de objetos mediante algoritmo, en contraste con la medición manual, se evaluó la variable *diferencia* expresada como $diferencia = |algoritmo - observador|$, donde *algoritmo* es la velocidad obtenida mediante algoritmo y *observador* es la medida manual de la velocidad. Para 16 conteos se obtuvieron los siguientes estadísticos descriptivos a un 95% de nivel de confianza: una diferencia media de 1.3633, una mediana de 1.35 y una varianza de 0.895. Según la prueba de normalidad de Shapiro-Wilk a 16 grados de libertad, se obtuvo una significancia de 0.521, o sea un p-valor ≥ 0.05 . Lo que implica que, en su conjunto, los datos de la variable *algoritmo* y *observador* respecto a velocidad de vehículos siguen una distribución normal. Además, se evaluó la variable *precisión* = $algoritmo/observador$ donde la prueba normalidad de Shapiro-Wilk indica una significancia de 0.540, es decir, p-valor ≥ 0.05 , y se acepta también que los datos de la variable *precisión* del algoritmo en determinar la velocidad de los vehículos siguen una distribución normal.

La prueba T de una muestra nos permitió comparar la media de la variable *precisión* con el valor de prueba esperado igual o superior a 0,95. Según los resultados que se observan en la Tabla 2, para los valores de prueba de 0.95, 0.98 y 0.99, el valor de prueba no se acepta debido a que el p-valor es menor a 0.05. Solo los valores de prueba de 0.96 y 0.97 se aceptan debido al p-valor ≥ 0.05 .

Por lo tanto, hay evidencia de que la variable *precisión* para la medición de velocidad de vehículos tiene una media aceptable en el rango de *precisión* entre el 96% y el 97%.

Variable	Prueba	t	gl	Sig. Bil.	Dif.medias	Inferior	Superior
PRECISIÓN	0,95	2.238	15	0.041	0.01307	0.0006	0.0255
PRECISIÓN	0,96	0.526	15	0.607	0.00307	-0.0094	0.0155
PRECISIÓN	0,97	-1.187	15	0.254	-0.00693	-0.0194	0.0055
PRECISIÓN	0,98	-2.899	15	0.011	-0.01693	-0.0294	-0.0045
PRECISIÓN	0,99	-4.611	15	0.000	-0.02693	-0.0394	-0.0145

Tabla 2 – Prueba T para una muestra, a un 95% de intervalo de confianza

Para la evaluación de los resultados del algoritmo de semáforo inteligente, se consideró 15000 escenarios simulados por el algoritmo (observar extracto de datos en la Tabla 3). Debido a que la variable de estudio *tiempo_espera_por_ciclo* es la suma del Tiempo de Espera (TE) de un tramo con el TE de su transversal. La ecuación $TEC_k = TE_{2k-1} + TE_{2k}$, donde $k=1,2,\dots, n/2$ expresa que se tomaron cada par de datos de tiempos de espera y se obtuvieron 7500 ciclos de semáforo.

Según la prueba de normalidad de Kolmogórov-Smirnov, el estadístico descriptivo presentó una media de 76.0661 s y una mediana de 76 s, con 7500 grados de libertad. Además, se obtuvo un p-valor < 0.05 . Con la prueba no paramétrica de Wilcoxon para una muestra, la mediana obtenida para la variable *tiempo_espera_por_ciclo* fue de 76 s con una significancia de 0.103, es decir, un p-valor ≥ 0.05 , lo cual nos permite retener la hipótesis de que la mediana es igual a 76 s a un nivel de confianza de 95%.

id	Fase	Tramo	V_NS	V_SN	V_EO	V_OE	TipoVel	Congestión	T.Espera
1	1	Norte-Sur	23	47	14	19	Alta	Medio	38
2	2	Oeste-Este	28	48	50	29	Moderada	muy Alto	58
3	1	Sur-Norte	43	25	26	27	Alta	Medio	38
...
14997	1	Sur-Norte	11	36	14	17	Alta	Bajo	28
14998	2	Oeste-Este	26	40	43	38	Moderada	muy Alto	58
14999	1	Sur-Norte	45	40	26	49	Baja	Medio	38
15000	2	Oeste-Este	15	23	12	40	Baja	Medio	63

Nota. V_NS, V_SN, V_EO, V_OE es el número de vehículos en cada tramo de norte-sur, sur-norte, este-oeste y oeste-este respectivamente, que a cierta tipo de velocidad Tipovel produce un tipo de congestión.

Tabla 3 – Extracto de la vista de datos de tráfico generado por simulación

En la Figura 8A se puede observar el gráfico de frecuencia, donde la mayor frecuencia de los tiempos de espera por ciclo se encuentran cercanos a 80 s y el valor de la media es igual a 76.0661 s. En contraste con el escenario real, donde el ciclo semafórico fijo es de 92 s, se hace notar la diferencia de 14 s. Por tanto, 76 s por cada ciclo semafórico significa 47.36 ciclos/h; mejora que se hace notar frente al número de ciclos semafóricos fijo igual a 39.13 ciclos/h.

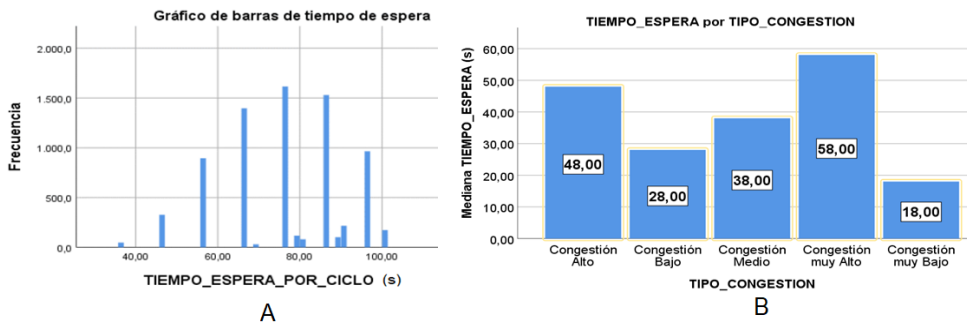


Figura 8 – Tiempo de espera y tipo de congestión

5. Discusión

Haber obtenido una precisión de 95% en cuanto al conteo de vehículos tiene una implicancia significativa en cuanto a confiabilidad del algoritmo de Yolov4, pues esto garantiza su funcionamiento e implementación en sistemas de producción. Sin embargo, los desafíos continúan respecto al problema del solapamiento u oclusión de vehículos, que son precisamente el causal del diferencial entre el conteo real y un conteo mediante

algoritmo. En la Figura 9 se observa un caso de oclusión donde el bus detectado ocluciona a otros vehículos que se encuentran detrás.



Figura 9 – Oclusión de vehículos

Los resultados del conteo de vehículos muestran que la precisión de un modelo de inteligencia artificial puede mejorarse, especialmente al abordar problemas de oclusión. La medición de velocidad alcanza una precisión entre 96% y 97% usando algoritmos de visión computacional, demostrando su viabilidad para aplicaciones en tiempo real. El presente trabajo destaca la importancia del tiempo real en el conteo y seguimiento de vehículos. Aunque YOLOv4 y el algoritmo IDM son precisos, mejorar la precisión podría aumentar la latencia y esfuerzo computacional.

El algoritmo 5 en este estudio añade dos tipificaciones de congestión: muy baja y muy alta, mejorando la segmentación del ICT en el rango (0;1). A diferencia de Ashifuddin et al. (2019), que consideran tres tipos de congestión (libre, media y alta), nuestra propuesta especifica cinco. Como muestra la Figura 8B, las congestiones muy alta y muy baja afectan significativamente el tiempo de espera por tramo vehicular y ciclo semafórico. Por otro lado, el algoritmo de gestión del semáforo ha reducido el tiempo de espera promedio por ciclo a 76 s, mejorando respecto al ciclo fijo de 93 s, usando un modelo heurístico basado en la velocidad y el número de vehículos. Aunque el FNC en Modo 2 o 3 no garantiza resultados alentadores, nuestro modelo supera al modelo de control difuso de Wang et al. (2022), quienes lograron una reducción del 14.59%, y el nuestro logró disminuir el tiempo de espera en un 18.3%.

En cuanto a la validez interna, la metodología utilizada para el conteo de vehículos y la medición de velocidad está claramente definida, asegurando que los datos recopilados son precisos y comparables bajo condiciones controladas. Respecto a la validez externa, la presentación de resultados cuantitativos y estadísticos descriptivos a un nivel de confianza del 95% y la reducción de tiempo de espera en un 18.3% nos permiten generalizar los resultados del algoritmo heurístico a una población más amplia y a otros tipos de escenarios. Sin embargo, también hay limitaciones potenciales, como considerar la precisión del método de detección de objetos en diferentes condiciones ambientales, festividades, accidentes y variables más difíciles de medir y controlar, como el comportamiento imprudente del peatón y/o del conductor en una intersección vial, que serían motivo de otro estudio.

Para una futura aplicación en zonas de alta densidad vehicular como Lima, Perú, se sugiere adoptar nuestro modelo propuesto, enfocándose en mejorar con algoritmos de sincronización entre intersecciones viales.

6. Conclusiones y trabajos futuros

Este trabajo propone un sistema inteligente heurístico para gestionar semáforos en intersecciones viales. Basado en el flujo vehicular, toma decisiones en tiempo real, optimizando el ciclo semafórico a 76 s y reduciendo el tiempo de espera en un 18.3% respecto a los ciclos fijos. Se validó una precisión del 95% con el algoritmo de conteo Yolo y del 96%-97% con el método IDM para medir velocidad, superando el 95% esperado. Futuras investigaciones podrían adaptar el algoritmo a redes LSTM o transformadores para mejorar el proceso del sistema heurístico actual.

Referencias

- Araya-Porras, E., Mora-Calderón, A., & Aguero-Valverde, J. (2022). Pedestrian crossing light violation in Costa Rica: exploring factors affecting mid-block crossing behavior. *Ingeniería*, 32(2), 115-134. <https://dx.doi.org/10.15517/ri.v32i2.50386>
- Ashifuddin, M., Rehena, M.. (2019). Intelligent Traffic Congestion Classification System using. *ACM Library*, 110-116. <https://doi.org/10.1145/3308560.3317053>
- Camargo, E. (2018). El “Pico y Placa” en Bogotá: ¿peor el remedio que la enfermedad? *Universidad Externado de Colombia*, 47. <https://doi.org/10.18601/01236458.n47.07>.
- Cheung, S., & Kamath, C. (2005). Robust background subtraction with foreground validation for urban traffic video. *Journal on Advances in Signal Processing*, 1-11.
- Gonçalves, M., Marques, T., Gaspar, P., Soares, V., & Caldeira, J. (2023). Prototype Solution for Detecting and Signaling Road Pavement Defects Based on Computer Vision Techniques. *RISTI - Revista Ibérica de Sistemas y Tecnologías de Información*, 25-44. <https://doi.org/10.17013/risti.52.25-44>
- Huawei Technologies Co., Ltd. (2023). *Artificial Intelligence*. Springer. <https://doi.org/10.1007/978-981-19-2879-6>
- Inglés-Romero, J., Cano-Baños, M., Doménech-Asens, G., & Morales-Estera, V. (2018). Traffic Metrics at Urban Intersections using a Low-Cost Image Processing Approach. *Association for Computing Machinery*, Article No. 9. <https://doi.org/10.1145/3243394.3243693>
- Javadi, S., Dahl, M., & Pettersson, M. (2019). Vehicle speed measurement model for video-based systems. *Computers & electrical engineering*, 76, 238-248.

- Jeyabharathi, D., & Dejeay, D. (2016). Vehicle Tracking and Speed Measurement system (VTSM) based on novel feature descriptor: Diagonal Hexadecimal Pattern (DHP). *Journal of Visual Communication and Image Representation*, 40, 816-830. Obtenido de <https://www.sciencedirect.com/science/article/abs/pii/S1047320316301705>
- Jiménez-Murillo, D., Henao-Garcés, S. A., Castro-Ospina, A. E., Duque-Muñoz, L., & Nieto-Mora, D. (2023). Segmentación de tejido cerebral mediante redes neuronales convolucionales basadas en U-Net. *RISTI - Revista Ibérica de Sistemas y Tecnologías de Información*, 5-24. <https://doi.org/10.17013/risti.52.5-24>
- Manrique, W. (2018). Estudio del tráfico y tendencias de movilidad urbana en Lima 2018. Lima: Marketwin. Obtenido de <http://willardmanrique.pe/wp-content/uploads/2018/09/Estudio-de-Trafico-y-Tendencias-de-Movilidad-Urbana-2018.pdf>
- Ministerio de Seguridad Pública de China. (2023). The number of motor vehicles in the country has reached 417 million, and the number of drivers exceeds 500 million. Obtenido de https://www.gov.cn/xinwen/2023-01/11/content_5736278.htm
- Ministerio de Transportes y Comunicaciones del Perú. (2018). Manual de dispositivos de control de tránsito automotor para calles y carreteras. Lima: MTC. Obtenido de https://portal.mtc.gob.pe/transportes/caminos/normas_carreteras/MTC%20NORMAS/ARCH_PDF/MAN_6%20DCT-2016.pdf
- Nigam, N., Singh, D., & Choudhary, J. (2023). A Review of Different Components of the Intelligent Traffic Management System (ITMS). *Symmetry*, 15(3), 583. <https://doi.org/10.3390/sym15030583>
- Nikas, A., Koasidis, K., Köberle, A., Kourtesi, G., & Doukas, H. (2022). A comparative study of biodiesel in Brazil and Argentina: An integrated systems of innovation perspective. *Renewable and Sustainable Energy Reviews*, 156, 112022. <https://doi.org/10.1016/j.rser.2021.112022>
- OCDE. (2011). OCDE. Obtenido de: <http://www.oecd.org/greengrowth/greening-energy/49157219.pdf>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779-788. Obtenido de https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf
- Saus, M. (2023). State of the art about urban development and transport: reviewing tools for Latin America. *Economía, Sociedad y Territorio*, 991 - 1016. <https://doi.org/10.22136/est20231894>
- Tomar, I., Sreedevi, I., & Pandey, N. (2022). State-of-Art review of traffic light synchronization for intelligent vehicles: current status, challenges, and emerging trends. *Electronics*, 11(3), 465.

- Wang, M., Wu, X., Tian, H., Lin, J., He, M., & Ding, L. (2022). Efficiency and Reliability Analysis of Self-Adaptive Two-Stage Fuzzy Control System in Complex Traffic Environment. *Journal of Advanced Transportation*, 6007485. <https://doi.org/https://doi.org/10.1155/2022/6007485>
- Wang, S., Shen, Y., Bao, Z., & Qin, X. (2019). Intelligent Traffic Analytics: From Monitoring to Controlling. *Association for Computing Machinery*, 778-781.
- Zuraimi, M., & Zaman, F. (2021). Vehicle Detection and Tracking using YOLO and DeepSORT. *IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 23-29. <https://doi.org/10.1109/ISCAIE51753.2021.9431784>