CMU Journal of Science
The Official Scientific Journal of Central Mindanao University
ISSN Print: 0116-7847 js.cmu.edu.ph ISSN Online: 2704-3703

Research Article

An Internet of Things-based Traffic Light Control Using Computer Vision for Real-time Traffic Density Calculation

Josh Ivan M. Caramat[1], Princess Leila R. Ditchon[2], Julie Ann C. Lontoc[3], Ivan Neil B. Tapel[4]

1 Department of Information Technology Cavite State University - Main Campus, Cavite, Philippines; joshivan.caramat@cvsu.edu.ph

2 Department of Information Technology Cavite State University - Main Campus, Cavite, Philippines; princessleila.ditchon@cvsu.edu.ph

3 Department of Information Technology Cavite State University - Main Campus, Cavite, Philippines; jclontoc@cvsu.edu.ph

4 Department of Information Technology Cavite State University - Main Campus, Cavite, Philippines; ivanneil.tapel@cvsu.edu.ph

ABSTRACT.

The use of traditional traffic light timers in the Philippines has limitations in addressing dynamic fluctuations of traffic density. This observation prompted the study to design an adaptive traffic light system. While most existing research has focused on vehicle counts as a basis for adjusting traffic signals, few studies have utilized computer vision to consider vehicle occupancy for the same purpose. Moreover, the use of emerging computer vision technologies in this context warrants further exploration. In response, the researchers developed a system capable of adjusting traffic light timing based on real-time traffic density calculated through computer vision. The system gathers live traffic feeds and employs a YOLOv8-based model to detect and classify vehicles, thereby identifying current traffic conditions. A prototype was developed to simulate a traffic light system integrated with the Internet of Things (IoT). Results show that the model can detect and classify vehicles into four categories with a mean Average Precision (mAP@0.50) of 0.94. These findings demonstrate the potential of the proposed system as a precursor to the actual implementation of adaptive traffic lights and improvement of urban traffic management.

Keywords: Traffic light control, computer vision, traffic density, YOLOv8, traffic management

Citation: Caramat, J. I., Ditchon, P. L., Lontoc, J. A., & Tapel, I. N. (2026). An Internet of Things-based Traffic Light Control System Utilizing Computer Vision for Real-time Traffic Density Calculation. *CMU Journal of Science*, *30*(1), 63

Academic Editor: Dr. Einstine M. Opiso

Received: Nov. 23, 2025
Revised: June 2, 2026
Accepted: June 9, 2026
Published: July 2, 2026

INTRODUCTION

Efficient transportation plays a crucial role in sustaining economic activity, social connectivity, and overall urban development [17]. However, many major roads in the Philippines experience chronic traffic congestion that reduces travel speed, increases fuel consumption, and degrades air quality. Traffic congestion occurs when the number or movement of vehicles exceeds road capacity, causing extended travel times and inefficiencies in traffic flow [5]. This condition underscores the pressing need for traffic management systems that can adapt to real-time conditions, rather than relying solely on fixed-time control strategies.

Traditional traffic-signal controllers typically operate on preprogrammed timing plans that do not adjust to fluctuating vehicle volumes. As urban traffic becomes increasingly complex, static signal timers often fail to account for the varying vehicle types, occupancy rates, and road demands. These inefficiencies often lead to increased delays and conflict points at intersections [17]. Recent studies have emphasized the importance of adopting proactive and data-driven solutions to improve intersection safety and overall traffic operations [1].

In the Philippine context, [16] reported that Metro Manila consistently ranks among the most congested urban areas in Southeast Asia, with economic and environmental impacts extending to adjacent provinces. Local studies also emphasize the importance of integrating sustainable mobility measures and technological innovations into traffic systems to support long-term development objectives [4].

Meanwhile, research on shared mobility and multimodal integration highlights the role of intelligent transport systems in addressing last-mile and network efficiency issues. [7] emphasized that shared-mobility and data-driven transport strategies can alleviate congestion and promote urban efficiency. Similarly, [6] proposed a conceptual framework for integrating transport demand modeling with emerging technologies, which aligns with adaptive traffic-signal control systems.

With the advancement of computer vision and Internet of Things (IoT) technologies, real-time vehicle detection and classification have become more accurate and accessible. Studies on traffic-density estimation suggest that area-based occupancy and vehicle classification can yield more reliable density metrics than traditional vehicle-count approaches [12]. These technologies can provide dynamic data that can be directly linked to signal timing algorithms.

In response to these developments, this study proposes an IoT-enabled adaptive traffic light control system that utilizes a YOLOv8-based computer vision model to detect and classify vehicles and compute real-time traffic density. The system aims to provide accurate, real-time density estimates that account for vehicle types and sizes; allocate signal time dynamically based on traffic demand; and demonstrate a working prototype integrating a vision-based detection model with an Arduino-based controller. This project serves as a proof-of-concept for an adaptive signal control system that can potentially improve traffic efficiency and serve as a precursor for real-world implementation.

METHODOLOGY

To enable the model to detect and classify vehicles accurately, several development stages were undertaken. The process involved data collection, preprocessing, model training, and system integration using the YOLOv8 algorithm. This section describes each stage of development and the construction of the IoT-based traffic-light control prototype.

Model Development

Data and Data Collection

The researchers used a modified classification from the Toll Regulatory Board (TRB) to include motorcycles with under 400 cubic capacities (cc), since the program will be tested at an intersection with no cubic capacity limitations. The researchers created a new classification of vehicles, as shown in Table 1. The researchers manually recorded videos of the incoming vehicles at the intersection to collect data. A smartphone (iPhone 14 Pro Max) was used to record the videos for the dataset. The video resolution was set to 1280x720 pixels with 30 frames per second. The researchers gathered 1000 images of each vehicle classification and incremented by 200 more images in each increment depending on the performance and accuracy of the model. The dataset was split based on training, validation, and test set to be fed in training a model for vehicle classification. This dataset size is sufficient in comparison with [15] where they collected 800 images for six (6) vehicle classifications totaling to 4800 images created.

Additionally, [9] conducted similar study utilizing 425 images per 10 vehicle models per approach (fine-tuning: Inception-v3, MobileNet-v2, and ResNet-18; and extraction: AlexNet). These approaches yielded 95.7% - 97.4% accuracy for fine-tuning and 93.5% accuracy for extraction. Given these, 1000 dataset per vehicle classification is valid and sufficient to yield high accuracy in the performance of the model. In Table 2, dataset was divided into training, validation, and test sets to be used to train the model for vehicle classification. This is in line with [14] that 70%/20%/10% split for train, validation, and test set is recommended for machine learning.

Table 1. Vehicle Classification

Class	Description	Examples
1	Lightweight vehicles with only two wheels.	Motorcycles, Bicycles, Scooters
2	Vehicles with two axles and a height not exceeding 7 feet 5 inches (2,286 mm).	Cars, Jeepneys, Vans, Pick-ups
3	Vehicles that exceed 7 feet 5 inches in height with two axles.	Buses, Trucks
4	Trucks with three or more axles and a height greater than 7.5 feet.	Large Trucks, Large Trucks with Trailers

Table 2. Dataset split

TRAIN SET	70%
VALIDATION SET	20%
TEST SET	10%

Data Preprocessing

The researchers utilized Roboflow [11] for the annotation of the datasets. The video was split into a one-frame-per-second format, creating one image for every second of the video. The researchers manually labeled the datasets by drawing bounding boxes around each vehicle. The images then underwent preprocessing procedures through Roboflow. Each image was resized to 1080 x 640 pixels to allow smaller file sizes and faster training. Lastly, the images were augmented by adjusting the exposure by -20% and +20%.

Model Training

The preprocessed datasets were exported and uploaded to Google Colab [3] for model training. Google Colab is hosted by the Jupyter Notebook service, which provides free access to computing resources, including Tensor Processing Units (TPUs) and Graphics Processing Units (GPUs). It is widely used in machine learning, data science, and education. Inside Google Colab, a Jupyter notebook will be uploaded for training the model. The YOLOv8 network has been designed to meet different needs, resulting in models of various sizes. The small-size model (yolov8s.pt) was chosen for training because it offers a good balance between accuracy and computational efficiency [7]. Additionally, [10] argued that YOLOv8 achieves superior inference speed while maintaining accuracy, enabling frame-by-frame vehicle detection in live video streams, and studies conclude that YOLOv8 outshines other YOLO variants (v3, v4, v5, and v7) in many aspects including speed and accuracy for traffic purposes. Moreover, [9] demonstrated the effectiveness of yolov8-based vehicle detection model trained on a 2,500-image Veh5 dataset achieving score results of mAP50 = 0.994 and mAP50–95 = 0.915.

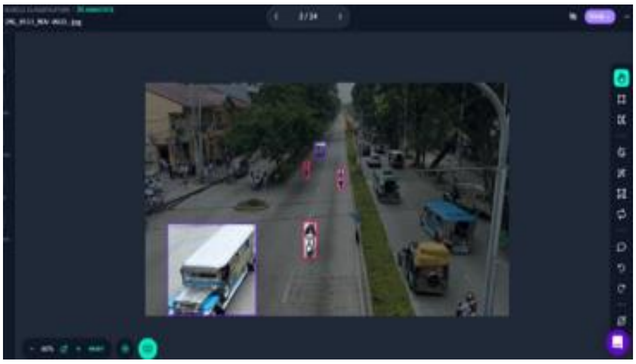

Figure 1. Roboflow Annotation Interface

IoT Traffic Light Control Prototype

System Architecture

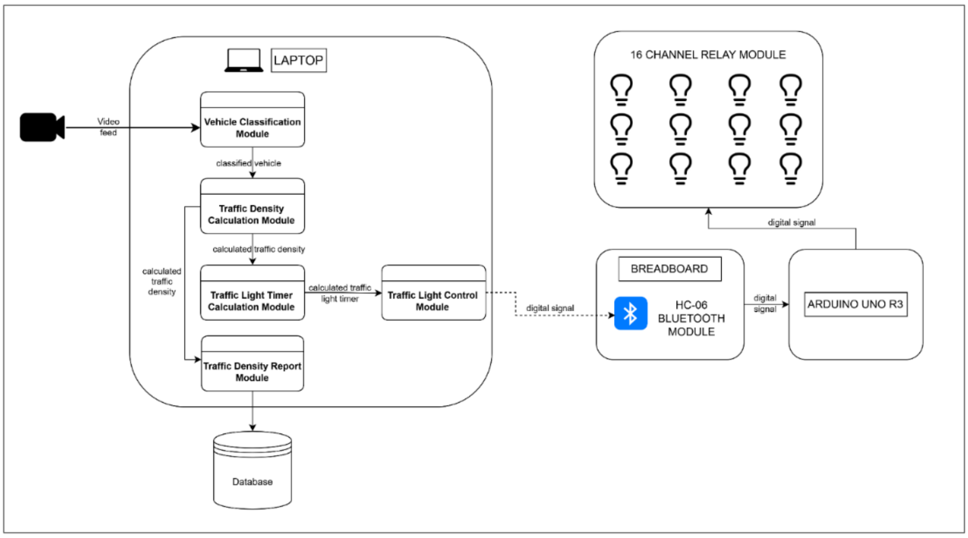

Figure 2. System Architecture

Figure 2 presents the system architecture of the study. It was developed to be modular and layered to ensure efficiency in real-time analysis and control of traffic. The system consists of several modules primarily responsible for traffic light control, traffic density calculation, and traffic light timing. This architecture includes a laptop for processing and an Arduino Uno R3 for control.

The laptop handles the software modules, while Arduino Uno R3 and the Bluetooth module operate the traffic lights through a relay module. The modules within the system are discussed as follows:

Vehicle classification module: Processes video feeds from real-time camera input and recorded traffic footage to classify vehicle types using the developed model. This module identifies the classified vehicles and their occupied area, which serves as input for calculating traffic density. The ROI must be defined to limit detection and classification within a specified area.

Traffic density calculation module: Calculates the traffic density. The traffic density (D_1) is calculated by the total area occupied by the vehicle ($TA_{(veh)}$) over the area of the region of interest (AROI) multiplied by 100 (Eq. 1).

$$D_i = (TA_{veh}/A_{ROI})100$$

To determine the area of each vehicle, the researchers gathered information about average vehicle dimensions and utilized the data shown in Figure 3, produced by the study of [9].

Class	Description	Pictorial representation of vehicle class	Length (m)	Width (m)	Area (m^2)	Ave. Class Area (m^2)
1	Motorized Two-Wheeler		1.65	0.76	1.25	1.25
2	Small Car / Passenger Car		3,84	1.45	5.57	6.6
	Big Car, Sports Utility Vehicles, etc.		4.60	1.80	8.28	
	Small-Sized Commercial Goods Vehicles		4.36	1.38	6.02	
3	Light-weight Commercial Goods Vehicles		5.62	1.52	8.54	13.38
	2-Axle Rigid Trucks/ Buses		8.28	2.20	18.2 2	
4	3-Axle Rigid Trucks/ Buses		9.10	2.30	20.9 3	25.74
	4-Axle, 5-Axle and 6-Axle Trucks		12.62	2.42	30.5 4	

Figure 3. Average Area of Each Vehicle Classification

Traffic light timer module. Sets the red and green light timers based on the calculated traffic density. It produces computed timer values for both red and green lights, which are sent to the Arduino Uno R3 microcontroller via the HC-07 Bluetooth module. Green light durations are determined according to the level of traffic congestion (see Table 2).

Traffic light control module: Composed of hardware components such as Arduino Uno R3 microcontroller, HC-06 Bluetooth module, 16-channel 5V relay module, and E27 light bulbs. The allocated timer is received by the microcontroller through the Bluetooth module, which controls the lightbulb duration using the relay module.

Traffic density report module: Create records of traffic density patterns. The database logs information such as lane number, vehicle classes, traffic density level, hour, minute, day, date, month, year, and the calculated traffic density at a specific time.

Table 3. Lane Green Light Timers

Level of Traffic	Traffic Density Percentage	Green Light Timer
	Lane 1	
Low	$D1 \leq 33.333\%$	21
Moderate	$33.333\% < D1 > 66.666\%$	42
High	$66.666\% < D1 > 100\%$	65
	Lane 2	
Low	$D2 \leq 33.333\%$	35
Moderate	$33.333\% < D2 > 66.666\%$	70
High	$66.666\% < D2 > 100\%$	105
	Lane 3	
Low	$D3 \leq 33.333\%$	11
Moderate	$33.333\% < D3 > 66.666\%$	22
High	$66.666\% < D3 > 100\%$	35
	Lane 4	
Low	$D4 \leq 33.333\%$	13
Moderate	$33.333\% < D4 > 66.666\%$	26
High	$66.666\% < D4 > 100\%$	40

Set-up

The prototype IoT traffic light control system is composed of different hardware materials, including the Arduino UNO, Bluetooth Relay Module, 16 Channel 5V Relay Module, and bulbs that serve as the traffic light signals, as shown in Figure 4.

Arduino and Bluetooth Module: The Bluetooth module allows a wireless connection with the Python program. The module sends data to the Arduino to trigger the bulbs. The digital pins are connected to the RXD and TXD pins of the Bluetooth module for data transfer to the Arduino, while the VCC and GND pins provide power. Since the Arduino uses a 5V trigger, the researchers used two 1k Ohm resistors as a voltage divider to stabilize the input voltage for the Bluetooth module.

Arduino and 16 Channel 5v Relay: Since the relay module is a low-level trigger, it is activated using a low signal from the microcontroller. To supply the Arduino with power, the researchers utilized the 5v output from the relay module and connected it to the VIN port of the Arduino. Additionally, controlling the relay required connecting the input pins of the relay module to the Arduino's digital pins.

16 Channel 5v Relay Module to 5v Bulbs: Each positive wire of the bulbs was connected to the Normally Open (NO) terminal of the relay module. The ground wires were connected sequentially to one another until the last bulb. Lastly, the Common ports of the relay module were interconnected, resulting in two wires extending from the last common port and joining with the ground wire of the bulb. These two wires were then connected to a power plug for the AC power source.

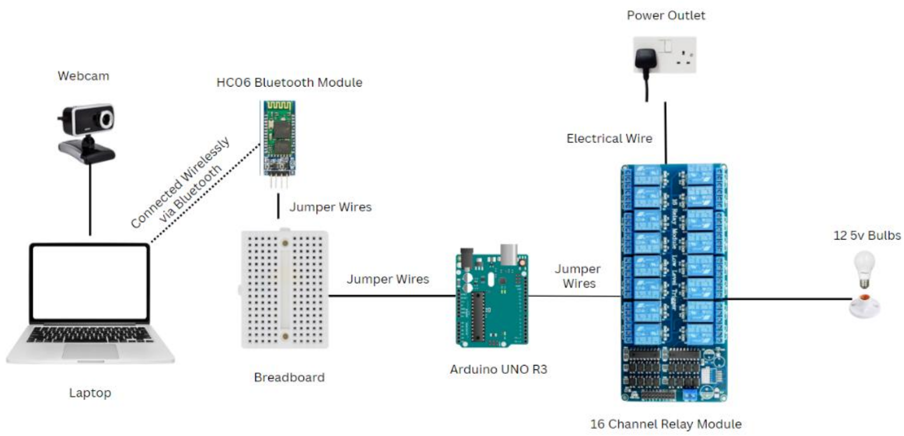

Figure 4. Diagram of prototype IoT Traffic Light Control

RESULTS AND DISCUSSION

3.1 YOLOv8-Based Vehicle Classification Model

The latest version of the developed model successfully performed its intended function of detecting vehicles. The model consistently classified the detected vehicles from a traffic video within the ROI and provided these classifications as input variables to the program, which then calculated the traffic density and adjusted the traffic light timer accordingly, as shown in Figure 5.

(a) (b)

Figure 5. Raw traffic image (a) and traffic image in the system with the corresponding light timer (b)

There were instances of vehicle occlusion observed during testing in the earlier versions of the model. Specifically, Class 1 vehicles were more likely to be occluded, suggesting that the model requires additional datasets containing more instances of Class 1 vehicles, as shown in Figure 6.

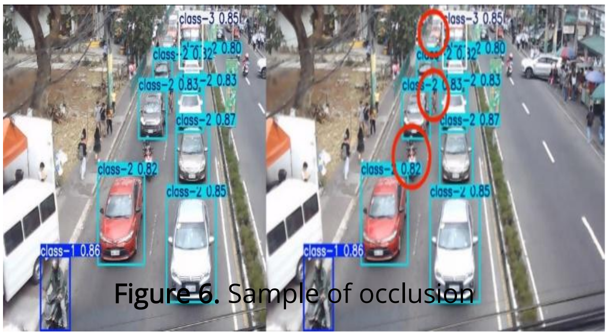

Figure 6. Sample of occlusion

To address this issue, the researchers gathered additional datasets that included more instances of Class 1 vehicles and applied hyperparameter tuning, which helped improve the model's detection performance in the later versions. In addition, the most recently trained model was fine-tuned to further enhance its overall performance.

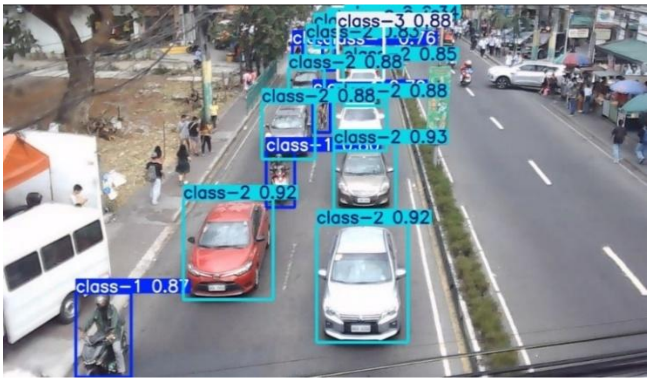

Figure 7. Sample of the detection of the final version

The trained model gained at most 91.4% mAP@50 and 67.1% mAP@[50:95] which shows low loss values and high precision and recall. To further improve the model, the researchers fine-tuned the latest model which resulted to 94% mAP@50 and 69.7% mAP@[50:95] based on the prepared validation data sets gathered in Dasmarinas with four lane intersection. Figure 8 presented the evaluation metrics result of the latest version before and after fine-tuning.

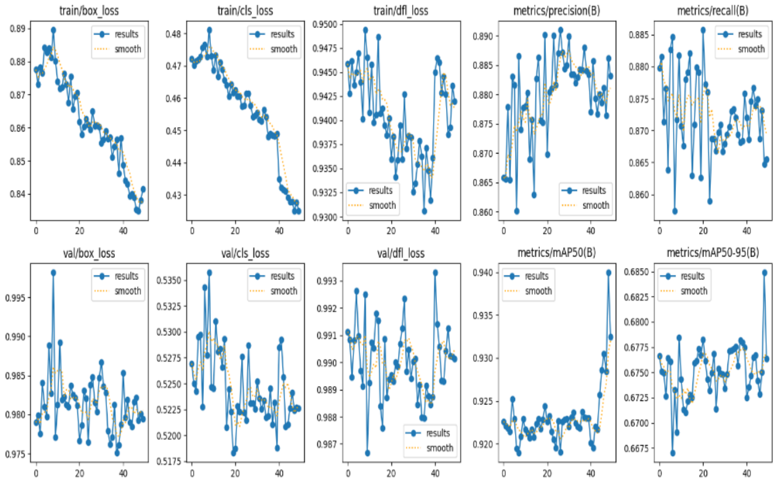

Figure 8. Evaluation of the latest version of the developed YOLOv8 model

Prototype IoT Traffic Light Control

The ROI utilized in the system is based on the camera's viewing range from the footbridge overlooking the traffic, as shown in Table 3. The ROI is set within the vehicle classification module. The Arduino Uno is programmed to fetch and display the appropriate light pattern. Figures 9 and 10 present the IoT Traffic light control prototype.

The prototype of the Internet of Things (IoT)-enabled traffic light control system functioned as designed. The camera operated as the "eye" of the system, capturing live traffic that was integrated into the program, which then produced a video output with vehicle detection and classification within the ROI.

Table 3. Adjusted ROI for each lane

Lane	Length (m)	Width (m)	Area (m^2)
1	25	6	150
2	25	6	150
3	25	9	225
4	25	6	150

Figure 9. A prototype of Internet of Things-based traffic light control showing ROI

Figure 10. A prototype of Internet of Things-based traffic light control showing output based on video feed

The prototype of Internet of Things-enabled traffic light control fulfills the purpose of simulating traffic light. The prototype underwent unit testing and integration testing that ensures the suitability of the assembled output for the expected outcome. System testing was done to evaluate the capability of the system in terms of speed to test its operation capacity in real-time environment. Results showed the highest frame per second (FPS) recorded is 20 FPS. On the other hand, the best inference speed is 20.77ms, and the average inference time is 28.49ms but appears to be lower in a lower GPU. This shows that the capacity of the system (FPS and inference speed) to operate is dependent on the GPU used.

CONCLUSION

This study demonstrated that utilizing computer vision enables adaptive traffic light management. The YOLOv8 algorithm was employed to create a vehicle detection model that can categorize vehicles into distinct types. Using the model's output, the system determined the average area of vehicles to calculate traffic density. The study produced an Internet of Things (IoT)-based traffic light control prototype that incorporated the vehicle detection model. The camera, which served as the "eye" of the prototype, transmitted live traffic feeds to the program, allowing it to classify detected vehicles within the Region of Interest (ROI).

The results highlighted that the YOLOv8 algorithm was suitable for detecting and classifying vehicles in real traffic conditions. In addition, the quantity and quality of datasets significantly influenced the model's performance, particularly its ability to accurately identify vehicle classes and the effectiveness of dataset preprocessing. The performance of the model presented an accuracy score of 94% using the metric mAP@50. This result was obtained by continuously updating the datasets based on the previous performance of the model, hyperparameter tuning was also performed to determine the best algorithm settings in training the model. The model's evaluation indicated a strong relationship between dataset characteristics and the mean Average Precision (mAP) achieved for each class.

Overall, the study serves as a precursor to enhancing the mechanism of traditional traffic lights. It is recommended to partner with a municipal traffic authority to run a pilot at one or two intersections with diverse traffic patterns (peak/off-peak, mixed vehicle classifications, pedestrians); and integrate edge computing to meet the low-latency, reliability, and privacy requirements of an operational traffic-light control system. This innovation has the potential to enhance traffic management efficiency and contribute to the development of smarter and more sustainable urban road systems.

Author Contributions: All authors were actively involved in collecting data and took part in the writing, reviewing, and editing of the manuscript. Each author has read and approved the final version for publication.

Funding: This research received no external funding.

Data Availability Statement: The data supporting this study will be made available upon request by contacting the authors.

Acknowledgments: The researchers would like to thank Cavite State University for its continuous support, which made this research possible. The authors extend their gratitude to the municipality of Dasmariñas and Trece Martires City Traffic Management Office for enabling the researchers to gather information, conduct interviews, and test for this study.

Conflicts of Interest: The researchers declare no conflicts of interest.

REFERENCES

1. Agramon, P.M.G., Lim, M. B. B., Uaje, M. J. B., & Maunahan, H. S. (2024). Proximal surrogate measures for proactive safety evaluation of right-angle vehicle collisions in an intersection. *CMU Journal of Science, 28(1). https://doi.org/10.52751/cmujs.2024.v28i1.tjn2g562*
2. George, R., Vanajakshi, L. D., & Subramanian, S. C. (2019). Area Occupancy-Based Adaptive density Estimation for mixed road traffic. *IEEE Access*, *8*, 5502–5514. https://doi.org/10.1109/access.2019.2962373
3. *Google Colab*. (n.d.). https://colab.research.google.com/drive/1LVKFKYT6TBrYGzG1_hb-sHwyHm8_3gM5?usp=sharing
4. Frande Echaluce, & Macabeo, M. (2024). Evaluation of Traffic Accidents in the Province of Albay: Basis for Sustainable Traffic Management Framework. *JPAIR Multidisciplinary Research*, 55(1), 137–161. https://doi.org/10.7719/jpair.v55i1.487
5. Kumar, M., Kumar, K., & Das, P. (2021). Study on road traffic congestion: A review. *CRC Press eBooks*, 230–240. https://doi.org/10.1201/9781003139883-48
6. Lim, M. B., Lim, H. J., Agramon, P. M., Vitales, G., & Anabo, J. M. (2023). A proposed conceptual framework for the integration of agent-based microsimulation to activity-based demand models. *CMU Journal of Science, 27(2). https://doi.org/10.52751/GQKL6394*
7. Lim, M. B., Lim, H. J., & Rojas, A. W. M. (2022). Bridging the last mile through shared mobility towards an integrated mobility system. *CMU Journal of Science, 26(2). https://doi.org/10.52751/lvpo5704*
8. Ma, M., & Pang, H. (2023). SP-YOLOv5s: an improved YOLOV5 model for remote sensing image tiny object detection. *Applied Sciences, 13*(14), 8615. https://doi.org/10.3390/app13148615
9. Najeeb, S. A., Raza, R. H., Yusuf, A., & Sultan, Z. (2022). Fine-Grained vehicle classification in urban traffic scenes using deep learning. *Lecture notes in electrical engineering* (pp. 902–908). https://doi.org/10.1007/978-981-16-8129-5_138
10. Ngo, H. (2023, June 30). Vehicle-detection-based traffic density estimation at road intersections. *International Journal of Open Information Technologies*. http://injoit.org/index.php/j1/article/view/1554
11. Roboflow. (n.d.). notebooks/notebooks/train-yolov8-object-detection-on-custom-dataset.ipynb at main · roboflow/notebooks. GitHub. https://github.com/roboflow/notebooks/blob/main/notebooks/train-yolov8-object-detection-on-custom-dataset.ipynb
12. Saklani, S., Dhondiyal, S. A., & Singh, D. (2025). Real-Time Traffic Management System using YOLOV8: An analysis of various YOLO models. *4th OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry*. 5.0, 1–6. https://doi.org/10.1109/otcon65728.2025.11071056
13. Singh, S., & Santhakumar, S. M. (2021). Empirical analysis of impact of multi-class commercial vehicles on multi-lane highway traffic characteristics under mixed traffic conditions. *Journal of Traffic and Transportation Engineering: English Edition, 11*(3), 545–562. https://doi.org/10.1016/j.jtte.2021.07.005
14. Solawetz, J.. (Sep 4, 2020). Train, Validation, Test Split for Machine Learning. Roboflow Blog: https://blog.roboflow.com/train-test-split/
15. Tas, S., Sari, O., Dalveren, Y., Pazar, S., Kara, A., & Derawi, M. (2022). Deep Learning-Based Vehicle Classification for Low Quality Images. *Sensors*, *22*(13), 4740. https://doi.org/10.3390/s22134740
16. *Traffic Congestion in Metro Manila* (2021, In Congressional Policy and Budget Research Department (No. IF2021-16). https://cpbrd.congress.gov.ph/images/PDF%20Attachments/Factsheets/2021/Factsheet%202021-16-Traffic-congestion-in-metro-manila.pdf
17. Victoria, A. P. C., O. G. D., Biances, J. R. A., & Muhi, M. M. (2018). Analysis of road traffic congestion in selected intersections. *International Journal of Civil Engineering, 1049–1058*.

https://doi.org/10.1007/978-981-97-0751-5_86