Vehicle Detection and Traffic Data Generation

Using YOLOv8 in Metro Manila Highways

A Thesis Presented to the
Faculty of the College of Science
Technological University of the Philippines
Ayala Blvd., Manila

In Partial Fulfillment of the
Requirements for the Degree
Bachelor of Science in Computer Science

Imee Q. Compra

Simon Daniel M. Dela Cruz

Ronan M. Esguerra

Joshuel Ernest Q. Simbulan

Andrew James S. Tejerero

June 2024

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines | Tel No. +632-5301-3001 local 608 Fax No. +632-8521-4063 | Email: cos@tup.edu.ph | Website: www.tup.edu.ph	Index No.	TUPM-F-COS-16-TAU
		Revision No.	00
	THESIS APPROVAL SHEET FOR THE UNDERGRADUATE PROGRAMS OF THE COS	Date	07012022
		Page	1 / 1

This thesis hereto entitled:

VEHICLE DETECTION AND TRAFFIC DATA GENERATION USING YOLOV8 IN METRO MANILA HIGHWAYS

prepared and submitted by **IMEE Q. COMPRA, SIMON DANIEL M. DELA CRUZ, RONAN M. ESGUERRA, JOSHUEL ERNEST Q. SIMBULAN,** and **ANDREW JAMES S. TEJERERO** in partial fulfillment of the requirements for the degree **BACHELOR OF SCIENCE IN COMPUTER SCIENCE** has been examined and is recommended for approval and acceptance.

PROF. EDWARD N. CRUZ
Adviser

Approved by the Committee on Oral Examination with a grade of **PASSED** on MAY 20, 2024.

PROF. JAN EILBERT LEE
Member

PROF. ARIEL TOMAGAN
Member

PROF. PRISCILLA SOTELO-BATOR
Member

PROF. DOLORES L MONTESINES
Department Head/Chair

Accepted in partial fulfillment of the requirements for the degree **BACHELOR OF SCIENCE IN COMPUTER SCIENCE.**

Date: June 13, 2024

DR. JOSHUA T. SORIANO
Acting Dean

Transaction ID	TUPM-COS-TAU-ELS-07012022 -0258PM
Signature	

Transaction ID Legend: TUPX-AAA (Office Code)-BBB (Type of Transaction)-CCC (Initial of employee)-MMDDYYYY (month day year)-HHMMAM/PM (hourminutesAM/PM)

Vehicle Detection and Traffic Data Generation using YOLOv8 in Metro Manila Highways

ABSTRACT

The worsening traffic conditions in urban cities reflect neglect in proper preparation and a lack of studies about traffic management. This problem can be pointed out as one of the effects of having insufficient comprehensive data, specifically in Metro Manila. The primary objective of this study is to develop a project that can replace the traditional ways of collecting and generating traffic data with a more robust, automated, and scalable web application. YOLOv8 – the main algorithm for vehicle detection – was implemented on a diverse dataset of traffic images in highway environments. The model's accuracy, speed, and robustness were assessed through precision, recall, and score metrics. Data generation techniques are also employed to export traffic count values from either live or recorded video inputs. The results demonstrate that the YOLOv8 achieved a high detection accuracy with a mean average precision (mAP) of 86.76% and a real-time processing speed of 20-30 frames per second (FPS), and successfully generating compact traffic data that can be used for data-driven scientific studies. That concludes that this project is a potential stepping stone towards understanding the underlying crisis involving road and transportation systems as it takes advantage of modern technologies to resolve and provide a faster and more flexible solution for the lack of extensive traffic volume data in cities.

Keywords: Vehicle Detection, YOLOv8, Data Generation,

ACKNOWLEDGEMENT

This research project was a collaborative effort, and we would like to express our deepest gratitude to all those who contributed to its completion.

First and foremost, we extend our heartfelt thanks to our research adviser, Professor Edward Cruz, for his support, guidance, and vital feedback throughout this study.

Special thanks go to the Disaster Risk Reduction Management Office of Parañaque City, especially to Dr. Ted Gonzales for authorizing our request and providing the necessary video traffic resources for our research.

We would like to acknowledge the financial support received from the DOST scholarship program, which greatly helped this research. Their good and deserved funds allowed us to dedicate our time and efforts to making this project.

Lastly, we deeply thank our families for their help and understanding throughout this journey. To our parents, who supported us, and to our partners, who provided endless love and encouragement, thank you.

This project is dedicated to all those who helped make this possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

Chapter 1

The Problem and Its Setting

Background of the Study

Traffic congestion has been a problem plaguing cities since the invention of modern vehicles. It has spiraled out of control as car ownership has expanded yearly, resulting in much time lost and millions to even trillions worth of economic resources. As defined by the *Encyclopedia of the City,* congestion results from the calculation of average journey times for routes under congested conditions, resulting in the time loss of each vehicle per kilometer ("Congestion," 2005).

Traffic congestion, or as more known in our country as heavy traffic, has become a part of contemporary Filipino culture due to its prevalence. It has been used as a plot point for Filipino films, signifying that the problem has become deeply rooted in society. It even has a segment every morning and evening in a news program that updates the public about the condition of traffic on major roads. Memes relating to heavy traffic always persist, going against the short attention spans and quick lifespans of memes from gen Zs. It has gone up to a point where Epifanio de los Santos Avenue, or EDSA itself, has become synonymous with heavy traffic. It transformed into an instrument for entertainment, allowing Filipinos to use it as a vehicle unbothered by the congested traffic to escape the horrific realities of Filipino commuting (Ancheta, 2015).

The Philippine government has implemented various possible solutions to solve the issues surrounding the traffic problem. One of the major imposed solutions is the Unified Vehicular Volume Reduction (UVVRP) policy of the Metro Manila Development Authority (MMDA). It has restricted vehicles with a plate number ending in 1 and 2 on

Mondays, 3 and 4 on Tuesdays, 5 and 6 on Wednesdays, 7 and 8 on Thursdays, and 9 and 0 on Fridays in driving along the major roads in Metro Manila from 7 a.m. to 10 a.m. and 5 p.m. to 10 p.m. on weekdays, except on holidays. MMDA stated that it could reduce traffic volume by 20% during peak hours based on their data as they reimplemented the UVVRP through the release of MMDA Resolution No. 22-14, series of 2022, due to it being postponed temporarily due to the pandemic (MMDA, 2022 Aug.). Although the MMDA stated it reduced the traffic volume, no study backed up with empirical evidence has verified their claim. The lack of intensive traffic data influenced these sets that detail the amount of vehicular volume given a major road in a set period of time. As such, data will be used to conduct studies that can verify whether the MMDA's claim is true or not. Aside from the UVVRP, the MMDA also implemented the MMDA Resolution No. 22-16, which reimplements the truck ban along Roxas Boulevard, Manila. Trucks with a gross capacity weight of more than 4,500 kilograms (about half the weight of a school bus) are not allowed to go through Roxas Boulevard. Instead, they need to reroute their path to the Port Area by cruising through the South Luzon Expressway or vice versa (MMDA, 2022 Oct.).

Even with the imposed policies to reduce vehicular volume along the major roads in Metro Manila, no signs of heavy traffic reduction were felt. The congested traffic only persisted, catching the attention of the whole world. Based on the Tomtom Traffic Index report, out of 390 cities across 56 countries, Manila City is ranked 9th in the worst traffic-congested city in 2022 (TomTom Traffic Index, 2023). It set an average record of 27 mins of travel time per 10km distance covered. This resulted in precious time being lost due to heavy traffic. Aside from time lost, heavy traffic also results in economic losses. According to a report from the Japan International Cooperation Agency (JICA), the Philippines lost ₱3.5 billion a day in 2017 due to traffic congestion in Metro Manila (CNN Philippines Staff, 2018). It displayed the worsened state of traffic conditions, where it was ₱2.4 billion losses daily from a report in partnership with the National Economic Development Authority (NEDA) back in 2014. It is now projected that if no proper solutions are placed, the Philippines will lose ₱5.4 billion a day by 2035.

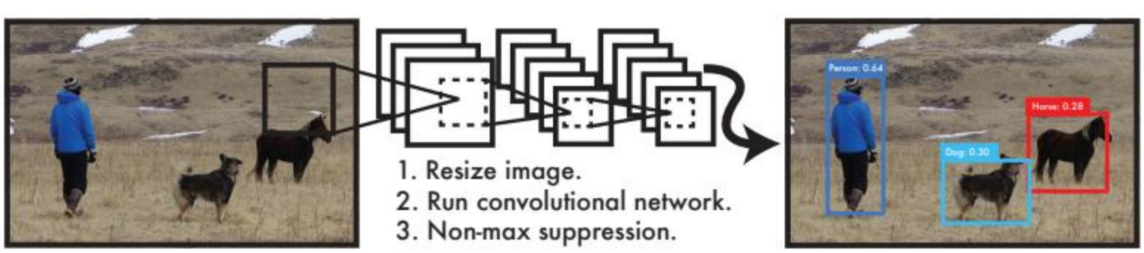
The applications of traffic modelling should accompany the formulation of solutions for traffic congestion. Through traffic modelling, a representation of a traffic system is simulated digitally via software to accurately depict traffic flow equal to what is observed and measured on road networks. This, in turn, can be used as a tool to assist in planning and managing traffic within the scope of what traffic system was simulated, allowing traffic engineers to formulate and visualize plans without spending more money and time (Azlan & Rohani, 2018). Real-world data is needed to verify and validate the accuracy of traffic models relative to the traffic system they simulate. This pertains to the hourly traffic volume data used to generate the Annual Average Data Traffic (AADT) yearly.

Figure 1*. MMDA Personnel Manually Counting Vehicles atop a Pedestrian Overpass*

For the purpose of gathering the traffic volume data along the major roads in Metro Manila, MMDA conducts a manual count from pedestrian overpasses, according to a statement from chief Bong Nebrija (Laurel, 2022), as shown in Figure 1. The manual method of physically counting the passing cars along the major roads in Metro Manila is a very intensive task. MMDA needs to record thousands of vehicles in an extended period while also enduring the erratic tropical weather in the Philippines. This does not also include the possibility of human errors, as some people may miss some vehicles or are too tired and distracted to focus on counting.

Artificial intelligence (AI) has generated a massive impact that is now referred to as the Industrial Revolution (IR) 4.0. It brought new ground-breaking technologies that revamped our approach to various things but were not limited to different functions of AI: automation, machine learning and vision, natural language processing, robotics, and self-driving (Tai, 2020). For this study, the researchers put great emphasis on automation, machine learning, and vision, as these will be applied to empowering traffic data gathering. Machine learning and vision will facilitate vehicle detection, and the counting of vehicles detected will be automated to generate a comprehensive report of vehicles detected.

Figure 2. *YOLO Detection System*



You Only Look Once (YOLO) is an object detection algorithm. As its name implies, it only looks at an image once to classify and detect objects. It revolutionized object detection algorithms as one of the first single-stage object detection algorithms. The proposal step in two-stage object detection algorithms involved the extraction of candidate boxes, which are the potential regions in an image where an object was possibly located. This step was removed in YOLO as it looked at the entire image during training and test time. It generalized representations of objects, making it faster and more accurate than Fast R-CNN in objects that required more context (Redmon et al., 2016).

In this study, the researchers intended to create a Vehicle Detection and Traffic Data Generation using YOLOv8 on Metro Manila Highways. This study enhanced the data gathering of MMDA, traffic engineers, researchers, and even hobbyists by enabling them to gather intensive traffic data with artificial intelligence. Users were empowered by artificial intelligence, specifically through the YOLO algorithm, to detect, tally, and vehicles. This generated intensive traffic data that can be used for empirical studies that necessitate such data. For example, in a study conducted by Castro et al. (2003) titled "*A STUDY ON THE IMPACT AND EFFECTIVENESS OF THE TRUCK BAN SCHEME IN METRO MANILA*," they highlighted the effects of the truck ban scheme in Metro Manila. This was done by showing intensive traffic data, where records of vehicle volume are displayed. Figure 3 shows the summary of the 16-hour truck volume count and 24-hour classified volume count survey for all vehicles of their study.

***Figure 3**. Vehicle Volume data for "Increase of vehicle traffic at night."*

Station	Period	Dir	Bus	Jeep	Car	Truck	Others	Total
CH04	24hr	Inbound	355	3211	5897	1156	1587	12206
		Outbound	400	3093	4723	979	1597	10793
	16hr	Inbound	287	2624	4816	830	1407	9965
		Outbound	284	2591	3893	723	1386	8876
	24/16hr	Inbound	1.2	1.2	1.2	1.4	1.1	1.2
		Outbound	1.4	1.2	1.2	1.4	1.2	1.2
CH09	24hr	Inbound	460	3088	13441	2246	1639	20874
		Outbound	440	3068	12767	2116	1641	20032
	16hr	Inbound	367	2334	10378	1576	1406	16061
		Outbound	384	2545	10338	1460	1424	16151
	24/16hr	Inbound	1.3	1.3	1.3	1.4	1.2	1.3
		Outbound	1.1	1.2	1.2	1.4	1.2	1.2
CH012	24hr	Inbound	1593	3748	20242	2593	1085	29260
		Outbound	1373	3768	16700	2343	856	25040
	16hr	Inbound	1399	3006	16945	1658	975	23983
		Outbound	1246	3253	13633	1272	775	20180
	24/16hr	Inbound	1.1	1.2	1.2	1.6	1.1	1.2
		Outbound	1.1	1.2	1.2	1.8	1.1	1.2
EX02	24hr	Inbound	2954	3393	28267	6159	27	40799
		Outbound	3501	3699	28197	6284	20	41703
	16hr	Inbound	2477	2903	25003	4652	17	35053
		Outbound	3125	3252	24130	4116	15	34637
	24/16hr	Inbound	1.2	1.2	1.1	1.3	1.6	1.2
		Outbound	1.1	1.1	1.2	1.5	1.4	1.2

These collected data and other related traffic data helped their study scientifically prove that the truck ban scheme reduces traffic. Although it minimized traffic, additional analysis was needed to properly estimate the benefits of reduced congestion concerning passenger trips and the environment (Castro et al., 2003).

This further proved that such data and studies could be the foundation for a scientific solution against traffic congestion. The intensive traffic volume data gathered by this project could contribute to the promotion of data-driven scientific studies by equipping fellow researchers with the tools to gather actual data that does not need as much effort as previous methods.

Objectives of the Study

The main objective of this study was to develop the project vehicle detection and traffic data generation using the YOLO algorithm, which could allow its users to gather intensive traffic data for data-driven scientific studies.

Specifically, it aimed to:

1. Design and develop the web application with the following features:
    a. Ability to detect, track, and classify the type of vehicle given a video input. To enable its users to filter the data they collected. It must recognize the type of vehicle under these classifications:
        i. Bus
        ii. Bicycle
        iii. Tricycle
        iv. Car
        v. Jeepney
        vi. Motorcycle
        vii. Truck
    b. Ability to generate traffic data report from the detected vehicles using computer vision. The traffic data report should contain information regarding the detected vehicle, specifically:
        i. Type of vehicle
        ii. Time of count
        iii. Exported in JSON API response format
        iv. Download the API response as CSV format
    c. Ability to connect an external camera that will serve as the video input where the AI will detect and record the vehicles.
    d. Ability to execute API functionalities properly, across different programming languages.

e. Ability to access the generated traffic data uploaded through the project's website or API through URL request.

f. Ability to download a summary report of the tallied vehicles stored in the database.

g. Ability to filter the accessed data via dynamic parameters such as address, class, date, time, and type (Public Vehicle, Private Vehicle).

h. Provide a user-friendly experience and allow easy learning of the API documentation.

2. Design and develop the system using Vite Bootstrap for the UI of the web application, Flask for developing the API, and MongoDB for the storage of the generated traffic data.
3. Test and improve the project.
4. Evaluate the project based on the characteristics set by the ISO/IEC 25010.

Significance of the Study

Various parties involved in traffic management may find the study useful. The following were some of the groups that could benefit from the study:

a) Traffic Management Authorities - The study could help traffic management authorities to monitor traffic flow and congestion in real-time. With the use of the YOLO algorithm, the authorities could automate the process of vehicle detection, making it more efficient and accurate. This could help traffic management authorities to make informed decisions about traffic management.

b) Developers - The study's use of open-source APIs could help developers create new applications that improve traffic management. They could use the data generated by the YOLO algorithm to create new applications that helped drivers navigate through congested areas, find parking spots, and avoid accidents.

c) Drivers - The study's contribution to the digitalization of manual tasks like vehicle counting could help drivers by reducing road congestion making it easier for vehicles to maneuver through jammed areas.

d) Researchers – The system may benefit students, researchers, and people in the academe in terms of data gathering and inference. The traffic data generated by the application were stored in a database that was publicly available for online queries using the web application or the API.

Scope and Delimitations

Vehicle Detection and Traffic Data Generation were created to enable gathering traffic volume to help formulate solutions for traffic congestion, and these gathered data were also used for other research purposes.

The researchers used the Vite web tool to easily bundle and serve front-end files. Flask, a backend framework, was then used to develop the API. Then, MongoDB is implemented for the system's database management, specifically for the storage of the traffic data that will be generated. To detect vehicles, the YOLO Algorithm was used. It identifies objects within an image by placing a bounding box around them and labels the detected objects based on their categories. The researchers trained the model with a well-equipped dataset to classify the detected vehicles under the types specified by the researchers. The custom model training was done by manually creating a custom dataset annotated via Roboflow, which was then trained with Ultralytics. To evaluate and test the web project in a real-world application, it will focus on the highway within Metro Manila in detecting the following:

- Vehicles
- Type of vehicle

Using a local connection with the city surveillance cameras, the output will be tallied on how many vehicles have passed through.

In the evaluation and testing of the project in relation to real-world applications, the researchers first had a trial test using a minimum of 1 hour and a maximum of 4 hours to test whether the web application can accurately detect a moving vehicle. If the test was successful, the researchers could gather 12 or more hours of footage from the City's Disaster Risk Reduction and Management Office (the said office is the one that records all the traffic footage around the city) to further test the effectiveness of the web application.

The project's functionalities were limited to:

- Vehicle detection
- Vehicle classification
- Traffic data generation
- Traffic data access through JSON API request and download the report from the project's website.
- Traffic data access through a downloadable report.
- Live camera feed input for vehicle detection.
- A web application with all the functionalities mentioned.

Chapter 2

CONCEPTUAL FRAMEWORK

This chapter displays the related literature, scholarly articles, academic papers, and studies that helped in building the foundation of knowledge that will be applied to this study. This chapter also contains a conceptual model of the study and the operational definition of terms.

Review of Related Literature and Studies

Traffic Congestion

According to the *Encyclopedia of the City*, traffic congestion is quantified by the time lost per vehicle per kilometer, calculated through a comparison between average journey times for regular routes and those affected by traffic congestion. This has been a persisting issue since the 1950s, and traffic congestion has been getting worse since the rapid expansion of car ownership in the early 1950s. By 1990, the United States and Western Europe had 85% and 70% of their journeys accounted for private cars only. The rapid increase in private car volume was caused by the *suburbanization* and scattering of commercial, leisure, and residential areas. This has caused mass transit systems to be less viable in comparison to private cars as city layouts have been heavily favoring private vehicle usage. This problem further intensifies during peak hours, or ‘rush hour’, when almost all working people go home simultaneously. Expansion of roads was done to combat this issue, but its capacity to hold vehicular volume consistently congestion-free was never met. This issue's persistence causes problems such as noise, greenhouse gas emissions, and lower air quality. Aside from environmental problems, it also heavily impacts the economy as it wastes time and fuel, thereby causing reduced work efficiency.

In the 1990s, over US$180 billion and US$30 billion per annum were estimated to be lost due to congestion (“Congestion,” 2005).

Congestion is a common word that can be understood by almost anyone. It pertains to “clog,” “impede,” and “excessive fullness.” For example, a congested nose is a nose full of mucous that hinders your normal breathing. Thus, under the context of traffic congestion, it pertains to the extra number of vehicles on a selected road, resulting in longer travel times in comparison to congestion-free traffic. Congestion is rooted in various factors that could be categorized under traffic-influencing events, traffic demand, and physical highway features. Traffic-influencing events are natural and man-made situations that cause congestion, such as traffic incidents, construction activities, weather, and disasters. Traffic demand is the traffic volume from fluctuations in normal traffic (day-to-day variability) and special events (sales, holidays, etc.). Physical highway features are the tangible characteristics of a road that it exhibits, for example, traffic control devices (traffic lights, signs, pedestrian lanes, etc.) and capacity – the physical maximum of the amount of vehicular volume a road can contain (Cambridge Systematics, Inc., 2005).

Traffic congestion can be simply defined as the extended travel time of vehicles from point a to point b. Whenever the maximum vehicular volume capacity of a road or road system has been reached, traffic congestion occurs. Level of Service (LOS) is used to classify the severity of congestion. Table 1 below is the Level of Service and its corresponding general operating condition:

Table 1. *Level of Service (LOS) of a road and its corresponding general operating conditions*

Level of Service	General Operating conditions
A	Free Flow
B	Reasonably free flow
C	Stable Flow
D	Approaching Unstable Flow
E	Unstable Flow
F	Forced or Breakdown Flow

Solutions against traffic were concocted, such as increasing the physical capacity of roads by adding more lanes. However, to achieve long-term, sustainable solutions, the implementation of Intelligent Transport Systems (ITS) was recommended. ITS provides smart solutions by monitoring traffic conditions 24/7, including traffic volume (number of vehicles that passed), weather, and accidents (*What Is Traffic Congestion? Traffic Congestion Meaning*, 2023).

Intelligent Transport System (ITS)

According to Choudhary (2019), the Intelligent Transport System (ITS) has become an integral smart city component. It enhances the quality of life and efficiency of travelling within the smart city. It applies the information gathered about local traffic, accidents, public transport schedules, etc., in enriching the efficiency of the smart city. ITS mainly gathers data and uses the data collected in analysis for various concepts of traffic management. Such applications are:

- Advance Traffic Management System
- Advanced Traveler Information System
- Advanced Vehicle Control System
- Advanced Public Transportation System
- Advanced Rural Transportation System
- Advanced Commercial Vehicles Operations System

According to Shaheen and Finson (2004), Intelligent Transport Systems (ITS) are technologies that apply wireless, electronic, and automated technologies. These can then help optimize road trips, reduce unnecessary distances travelled, reduce traffic congestion, etc.

Traffic Flow Theory

Elefteriadou (2014) stated that traffic flow is an area of study in transportation that concerns itself with the analysis of the capacity and traffic operational quality of transportation facilities. It aims to evaluate the operational quality of a traffic stream under a set of situations or conditions—for example, road network design and the vehicular volume of various types of vehicles on the road.

Traffic-flow theory aims to understand the mathematical intricacies of the interactions between vehicles, drivers, pedestrians, and the infrastructure. Infrastructure is represented by control devices (e.g., traffic lights), signage, and markings. To this day, traffic flow theory has become a foundation for all traffic models and analysis tools applied in designing and operating streets and highways (Lieu, 1999).

Traffic Flow

According to Wu (2021), traffic flow, also called traffic streams, is the analysis of the motion of different drivers and their vehicles between two points and the relationship between them. Under the context of math and civil engineering, traffic flow is the study of how various individual entities interact with structures (roads, signs, and traffic control tools) in order to understand and formulate solutions for traffic congestion problems (Shanmukha, 2016).

Traffic Flow Parameters

As stated by Oregon State University et al. (2001), traffic flow has a set of parameters that help describe and understand traffic flow more easily. These parameters or common sets of terms are Speed (v), Volume, Flow (q), Peak Hour Factor (PHF), Density (k), Headway (h), Spacing (s), Gap (g), and Clearance (c). Speed refers to the rate of time a vehicle travels. Volume refers to the number of vehicles that have gone through a set point on the road within a specific time range. Flow refers to the rate of vehicles that have gone through a set point on the road, normally set under a range of vehicles per hour. PHF refers to the hourly flow rate divided by the peak 15-minute rate of flow expressed as an hourly flow.

Density also refers to the amount of number of vehicles, but it is specified under a set length of road—for example, the number of vehicles across 15 kilometers. Headway is the time difference between the arrival of the first vehicle and its corresponding one given a designated point. Spacing refers to the length of the distance between the front bumper of the first vehicle and the front bumper of the second vehicle. Gap refers to the time between the first vehicle's exit and the second. Finally, clearance refers to the length of the distance between the rear bumper of the first vehicle and the front bumper of the second vehicle.

Traffic Flow Analysis

To analyze traffic flow, necessary traffic data must be gathered. Through sensing technologies and AI, traffic monitoring systems can be developed to automate the gathering of traffic data, such as traffic volume. The collected traffic data will then be studied to find solutions for traffic congestion. One glaring problem is the access to accurate traffic data in developing countries, where traffic congestion and lack of access to accurate traffic data bleed developing countries into millions and even trillions of economic damages (Tsuboi, 2021).

Urban Traffic Analysis

According to Li et al. (2022), the analysis of urban traffic is a crucial component of urban development. This analysis contributes valuable data for urban planning, traffic management, and the allocation of resources. The progressive evolution of Intelligent Transportation Systems, incorporating diverse urban sensing technologies as highlighted by Buch et al. (2011), resulted in an array of data regarding traffic dynamics. This encompasses data types such as Close-circuited television (CCTV) images, cycling counts, and traffic volume data, all contributing to the real-time monitoring of urban traffic conditions at a higher resolution. In comparison to conventional travel survey methods, these datasets furnish a better understanding of traffic patterns within our urban landscapes, ushering in a new era of insight into the intricacies of the urban transportation system.

Ways to Measure Traffic Data

In all levels of transport planning and decision-making, a comprehensive understanding of current conditions with respect to traffic flows is imperative (Yurshevich and Yatskiv 2012). This necessitates a detailed grasp of various aspects of traffic flows, including the types and speeds of vehicles, insights into trip length and frequency, the purpose of trips, as well as the origin and destination for each trip. Traffic counts serve as a primary method for acquiring a portion of this crucial data.

Presently, engineering tools for the measurement of traffic flow parameters (such as intensity, speed, and direction) primarily involve the use of video, laser, radio frequency, induction, sound, and pressure sensors (Kabashkin 1999).

1. Manual Count

    According to Zheng (2012), manual counting is the oldest form of measuring traffic volume. A traffic enforcer may tally using a piece of paper or an electronic handheld counter. With this method, enforcers usually count the number of vehicles at a certain time in a single day (i.e., during rush hour). This method is the easiest and most affordable method but not the best way and is labor intensive. In tests, manual counting is proved to be 99%.

2. Video Vehicle Detection

    Video Systems can automatically analyze video frames as cars pass by a certain road where the video camera is placed. These systems work almost as well as people watching the videos. The results come in quickly and can be used in city systems. This way of counting cars has advantages. It is not expensive because one camera can count cars in many directions simultaneously. You can change or add counting areas easily from a regular office computer. Using this system doesn't use up a lot of internet space. The car counts are sent online in real-time so traffic experts can see current and past counts from their web browsers. You can double-check the counts by watching the video and comparing them to automatic ones. This video counting happens all the time, throughout the year, and gives accurate numbers (Zheng, 2012).

3. Pneumatic Road Tube Counting

According to McGowen (2011), in Pneumatic Road Tube Counting, rubber hoses are put across the road and connected to a special device. When a pair of wheels goes over the hose, it squashes, making the device record the time. You can use a pair of tubes across different lanes of traffic. The device can determine which way the vehicle goes by first seeing which tube it crosses. However, if two vehicles cross the tubes at the exact same time, it's hard to know the direction. Also, if two cars are close together, the system might think they're one big vehicle. Companies say this method is about 99% accurate, but studies show the actual mistake is closer to ten percent for a typical 15-minute count. This means there are errors, but sometimes they cancel each other out. To get the counts, you must physically connect the device to a computer by the roadside. You need at least one tube for each direction on every road or junction you want to count. Setting this up means working on the actual road. These tubes are good for short counts on less busy roads. But they don't work as well on busier, multi-lane highways.

4. Piezoelectric Sensor

   Piezoelectric sensors operate by converting the mechanical energy from passing cars into electrical power. Installed in grooves on the road surface, these sensors get compressed when a vehicle drives over them, generating an electric signal or voltage. The strength of this signal corresponds to the extent of compression, and when the vehicle moves away, the voltage reverts. This change in voltage is harnessed for vehicle detection and counting. The counting device, connected to these sensors, is housed in a roadside enclosure, allowing on-the-spot data collection through a laptop connection or remote transmission via a modem (U.S. Department of Transportation, 2013).

5. Inductive Loop

   Caruso (1999) states that magnetic sensors find vehicles by checking how the Earth's magnetic field changes when cars go over it. The sensor can be on the road or in a box at the roadside. The sensor might have trouble telling them apart if cars are close together. It can notice all the different parts of a car passing over it. Sometimes, these sensors can even figure out details about the traffic, like what kind of vehicle it is.

6. Acoustic Detector

   According to Szwoch, G. (2021), Acoustic sensors offer an interesting option for traffic monitoring, even though they are not the most commonly used. Instead of sending out signals, these sensors analyze the sounds of vehicles, like the noise from their tires and engines. While they may not be the best choice for busy roads, they can be set up in places where cars naturally slow down, like around corners.

7. Passive Infrared Sensors

   Passive sensors do not emit their own energy. Instead, they pick up energy from two places: (1) energy from vehicles, roads, and other things in their sight, and (2) energy from the atmosphere that bounces off objects and comes into the sensor. In passive infrared sensors, an optical system focuses the captured energy onto a special material. The signals are then analyzed in real-time to see if there's a vehicle around. These sensors are usually placed above to watch traffic coming or going, but they can also be put on the side. Infrared sensors are used to control signals, measure traffic volume, speed, and type, find pedestrians in crosswalks, and give traffic info to drivers. The downside of this sensor is that one device can only measure up to two lanes maximum (Klein, 2006).

8. Doppler and Radar Microwave Sensors

   According to Klein (2006), the constant frequency signal (with respect to time) helps determine the speed of a vehicle using the Doppler principle. As a vehicle moves away from the radar, the received signal's frequency decreases, while it increases when a vehicle approaches the radar. Detecting the frequency shift indicates the passage or count of a vehicle. However, the constant frequency waveform can't measure vehicle presence because it only detects moving vehicles.

Artificial Intelligence (AI)

In computer science, artificial intelligence (AI) is the study of building machines capable of functions like speech recognition, visual perception, decision-making, and language translation—tasks that normally require human intellect. This is also known as a computer's or a machine's capacity to perform jobs that are typically performed by people since they call for human intelligence and judgment. While no AI can accomplish the vast array of things that a typical person can, certain AIs are comparable to humans in some skills. The earliest attempts to create machines that could think and learn like humans were made by computer scientists in the 1950s, which is when artificial intelligence (AI) initially came into being. Roser (2023) mentioned that artificial intelligence (AI) has advanced significantly and is now employed in many fields, such as robotics, image identification, and natural language processing. Healthcare, banking, transportation, and entertainment are just a few domains in which artificial intelligence has many uses. Personalized marketing, self-driving cars, chatbots, and virtual assistants are a few of the common uses of AI. Artificial intelligence emerged from science fiction and is now used in almost every business daily.

Furthermore, AI has the power to fundamentally alter the world in various ways, including cybersecurity and healthcare findings. The Encyclopedia Britannica's editors (n.d.) noted that it also poses several difficulties, including job displacements, privacy issues, and moral dilemmas. Many individuals contend that artificial intelligence (AI) enhances the standard of living by surpassing people in ordinary and complex jobs, hence simplifying, enhancing safety, and optimizing efficiency. Many contend that AI presents significant privacy hazards, exacerbates racism by normalizing individuals, and causes job losses for workers, thus increasing unemployment rates.

Artificial Intelligence is a broad discipline that includes several subfields, such as computer vision, robotics, and natural language processing. AI is an intriguing and diverse area to study, as each subfield has its own applications and issues. Moreover, computer vision will be the study's main focus. According to Sawtell-Rickson's (2022) definition, computer vision is a branch of artificial intelligence that focuses on giving robots the ability to interpret and comprehend visual input from their surroundings. It entails creating methods and algorithms that enable computers to identify and evaluate images and videos. Computer vision is receiving more attention as the need for computers to understand their environment grows. By simulating human vision, computers can view the world more clearly than humans. This is becoming increasingly important as we develop more sophisticated AI systems with capabilities that are closer to the abilities of humans. The field of computer vision began to take shape in the 1960s when scientists started to investigate the idea of building robots that could interpret visual information. Computer vision has advanced dramatically, according to Srivastava (2021), and it is now utilized in many different applications, such as augmented reality, face recognition, and self-driving automobiles. A wide range of businesses, including healthcare, retail, and entertainment, have used computer vision for several decades. Current technological capabilities and uses are a small portion of this technology's almost infinite potential. Finding out how computer vision may help people and enhance their quality of life—both in terms of behavior and form—is the first step in accepting and utilizing technology, according to Szeliski (2022).

Face recognition, image segmentation, and object identification are a few of the widely used computer vision applications. Ipopba (n.d.) stated that Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Support Vector Machines (SVMs) are among the techniques utilized in Computer Vision. These algorithms are employed in the training of models for the recognition and analysis of visual data.

Using computed tomography (CT) images, Hassan et al. (2022) provided a comprehensive overview of artificial intelligence (AI) and computer vision techniques for the diagnosis of the coronavirus illness of 2019 (COVID-19). Upon analyzing the earlier review studies, they discovered that all of them disregarded the classification and categorization of the COVID-19 literature according to computer vision tasks, including detection, segmentation, and classification. Segmentation and classification tasks are used extensively in the majority of COVID-19 CT diagnosis techniques. Furthermore, the majority of the review studies cover both CT and X-ray images and are varied. As a result, they have concentrated on the CT image-based COVID-19 diagnostic techniques. Well-known search engines and databases like Google, Google Scholar, Kaggle, Baidu, IEEE Xplore, Web of Science, PubMed, ScienceDirect, and Scopus were used to gather pertinent research. After a thorough examination, they gathered 114 studies and presented extremely detailed data for each chosen study. Their study indicates that immediate COVID-19 diagnosis is possible since AI and computer vision may greatly help automate the diagnosis procedure. Though more study is still needed, accurate and effective models will have real-time therapeutic effects. Future studies on the classification of literature according to computer vision tasks may be beneficial, and this study will serve as a solid basis for this research.

Object Detection and YOLO Algorithm

Object detection is a computer vision task that involves identifying and locating objects in images or videos. It is an important part of many applications, such as surveillance, self-driving cars, or robotics. Object detection algorithms can be divided into two main categories: single-shot detectors and two-stage detectors. Single-shot detectors use a single pass of the input image to make predictions about the presence and location of objects in the image. They are computationally efficient and suitable for real-time applications, but they are generally less accurate than two-stage detectors, which use two passes of the input image to make predictions.

YOLO (You Only Look Once) is a popular single-shot object detection algorithm that uses a single-shot convolutional neural network (CNN) to process images and classify objects. It is fast, accurate, and has learning capabilities. YOLO was first introduced by Redmon and Farhadi in 2016 and has since undergone several iterations, the latest being YOLOv8 by Ultralytics in 2023.

YOLO processes an image using a fully convolutional neural network (CNN). It divides the input image into a grid of cells and predicts a vector for each cell, consisting of the class probabilities and bounding box coordinates of the objects in that cell. It also uses non-max suppression to select the best bounding box per class. YOLO has several advantages over other object detection algorithms, such as:

- **Speed:** YOLO can process images in real-time, achieving up to 60 frames per second (FPS) on a GPU. This is much faster than other algorithms, such as R-CNN, which can only process a few frames per second.

- **Accuracy:** YOLO is accurate and robust, especially for detecting small objects and objects in motion. It also has low false positives and high recall, meaning it can detect most objects in an image without missing any.
- **Learning capabilities:** YOLO can learn the representations of objects and apply them in object detection. It can also be trained on different datasets and tasks, such as segmentation, pose estimation, tracking, and classification.

YOLOv8 is the latest version of YOLO by Ultralytics. It builds on the success of previous versions and introduces new features and improvements to further boost performance and flexibility. Some of the new features and improvements of YOLOv8 are:

- **Spatial attention:** YOLOv8 uses a spatial attention module to enhance the feature maps of the CNN. This module learns to focus on the most relevant regions of the image and suppress the background noise, resulting in more accurate and robust detection.
- **Feature fusion:** YOLOv8 uses a feature fusion module to combine the features from different layers of the CNN. This module learns to fuse the low-level and high-level features, capturing both the fine details and the global context of the objects, resulting in more precise and consistent detection.
- **Context aggregation:** YOLOv8 uses a context aggregation module to enrich the features of the CNN. This module learns to aggregate the features from neighboring cells, capturing the spatial relationships and dependencies of the objects, resulting in more comprehensive and diverse detection.

YOLOv8 is one of the key object detection algorithms in the field. It offers unparalleled performance in terms of speed and accuracy and supports a full range of vision AI tasks, such as detection, segmentation, pose estimation, tracking, and classification. It is suitable for various applications and domains and easily adaptable to different hardware platforms, from edge devices to cloud APIs.

API

An API, or Application Programming Interface, is a set of rules, protocols, and resources that enable communication between software programs. According to Reddy (2011), APIs provide the formats for data transmission, methods for gaining access to features, and the structure of requests and responses. Based on how they are utilized, APIs are categorized into several categories. This includes operating system APIs, library APIs, and web APIs. This project will revolve around the utilization of the API online or on the web.

1. Web API (HTTP APIs).

   Web APIs, often referred to as HTTP APIs, allow communication and data exchange over the internet using the HTTP protocol. They are commonly used for building web services and enabling interaction between web applications.

2. Endpoints

   Specific endpoints, which are distinct URLs or paths representing different functions or resources, are available via APIs. These endpoints function as specific locations within an API that accept requests and send back responses.

3. Data Format

APIs use specific formats to transmit data; the most common formats are JSON (JavaScript Object Notation) and XML (eXtensible Markup Language). JSON is preferred because of its ease of use, readability by humans, and lightweight structure. The data formats determine the structure of the information in API requests and answers.

4. Authentication and Authorization

To maintain security and privacy, APIs often require authentication and authorization mechanisms. These mechanisms validate the identity of users or applications accessing the API. Common methods include API keys, OAuth tokens, and token-based authentication.

5. Rate Limiting

Rate limiting is implemented to prevent overuse or abuse of API resources. It restricts the number of requests a client can make within a specified time frame, helping maintain fair usage and resource availability.

6. Documentation

API providers offer comprehensive documentation to assist developers in effectively utilizing their APIs. Documentation outlines available endpoints, request and response structures, authentication requirements, and other essential information for seamless integration.

7. REST and SOAP

Simple Object Access Protocol (SOAP) and Representational State Transfer (REST) are two popular architectural paradigms for creating APIs. Statelessness and the use of lightweight HTTP methods for interaction are characteristics of REST. Conversely, SOAP is a more rigorous, protocol-focused methodology. REST is becoming increasingly popular because it's easy to use and works well with online services.

JavaScript

A high-level programming language commonly applied in web development, web applications, game development, etc. It enhances web pages by allowing developers to implement dynamic features that cannot be achieved with HTML and CSS. Features such as click-to-show dropdown menu, interactive maps, form validation, and many more are done through JavaScript, and if JavaScript was not used, 90%, if not more, of the webpages we use would only have static animations. What makes JavaScript dynamic compared to HTML and CSS? As their name suggests, HTML and CSS are markup languages that can only provide document markups. This does not offer dynamic features and functions, unlike in JavaScript, where math calculations, fetching contents from another website, adding HTML contents to the Document Object Model (DOM), and lots more (Megida, 2021).

Python

Developed by Guido van Rossum in the late 1980s, Python is a popular general-purpose, interpreted, interactive, object-oriented programming language. It also allows third-party modules to vie the Python Package Index (PyPI), exceptions, dynamic typing, very high-level dynamic data types, and classes. Python includes functionalities and features such as high-level data structures, dynamic binding, extended system calls, run code written in C or C++, “glue code” that connects components of complex applications, and many more. Due to its ability to run on almost every system architecture, Python can be called a universal language, and it can be found in different applications. Python has also become an open-source programming language, and in 2018, van Rossum retired from his position as the Benevolent Dictator For Life (BDFL), and the future development of the Python language will be handled by the steering council off PEP 13 – Python Language Governance (Opensource.com, n.d.).

Vite

Vite, pronounced as *vit,* is derived from the French word that means “fast.” It is a new front-end web development build tool that combines a dev server and bundle combo, yet it is still leaner and faster. This is done by leveraging native ES module support and tools in browsers that are written in compile-to-native languages such as bundler build. It was released on April 20, 2020, and its latest version, Vite 5.0, was released last November 16, 2023 (Vite, n.d.).

Flask

A Pythonic web framework developed by Armin Ronacher. It provides developers tools to help them build their web applications easily with features such as URL routing and template engine. Compared to Django, Flask has an easier learning curve that allows developers to learn it easily. It has become one of the most popular web frameworks and is now used for developing complex applications (What Is Flask Python - Python Tutorial, n.d.).

MongoDB

MongoDB is an open-source NoSQL data management program. NoSQL databases serve as an alternative to traditional relational databases where instead of using tables and rows, NoSQL databases use either key-value pairs, column-oriented, or graph databases. This allows NoSQL databases to store unstructured or semi-structured data. In the case of MongoDB, its architecture is built upon a number of documents made up of key-value pairs. MongoDB is utilized in high-volume data storage, ad-hoc queries, indexing, load balancing, aggregation, server-side JavaScript execution, and other features. MongoDB supports many programming languages, such as C, C++, C#, Go, Java, Python, Ruby, and Swift. MongoDB works by storing the documents in a way that allows them to have field and value pairs. These are the basic units of data in MongoDB, and they are similar to JavaScript Object Notation (JSON) but use another version, the Binary JSON (BSON). This allows MongoDB to store more data types. The stored data can vary in types, such as documents, arrays, and arrays of documents. This stored data will then be incorporated with a primary key that will serve as their unique identifier. A set of documents are called collections, which have the same functionalities as a relational database table. It can store any type of data but cannot be spread across different databases. The architecture of a NoSQL database management system (DBMS) uses a single master for data consistency while having secondary databases that copy the primary databases for redundancy and automatic failover (Gillis & Botelho, 2023).

ISO/IEC 25010

The ISO/IEC 25010 serves as a quality evaluation system for software products. It defines the characteristics that should be taken into consideration when assessing the quality of software products in relation to the satisfaction of its intended stakeholders. To properly evaluate a software's quality using ISO/IEC 25010, eight characteristics are used to properly measure the degree of quality of a software product. The eight characteristics are:

- *Functional Suitability*
    - It describes the degree of functionalities provided by a product or system that satisfies the stakeholders' stated and implied needs under different use cases. This can be broken down into the following sub-characteristics:
        - Functional completeness: the measure of functionalities covering all the specified tasks and user objectives.
        - Functional correctness: the extent to which the product or system outputs correct and accurate results. In relation to machine learning and AI methods, correctness and incorrectness should be measured and observed carefully since such methods often do not provide functional correctness in all observed circumstances.
        - Functional appropriateness: the degree to which the functions facilitate the accomplishment of specified tasks and objectives.
- *Performance efficiency*
    - It describes the performance of the product or system in relation to the degree of resources it uses under a set of specific conditions. This can be broken down into the following sub-characteristics:

- Time behavior: the extent to which, during the execution of functions, the response and processing times and throughput rates of a product or system have met the predetermined requirements.
- Resource utilization: the measure in which, during the execution of functions, the amount and variety of resources used by a product or system meet the predetermined requirements.
- Capacity: the degree of the maximum limits of a product or system variable that has met the predetermined requirements.

- *Compatibility*
    - It describes the extent to which a product, system, or component can communicate information with other products, systems, or components and/or execute the required functions under situations such as sharing the same hardware and software environments. This can be broken down into the following sub-characteristics:
        - Co-existence: the extent to which a system or product executes its required functions properly while the environment and resources it currently uses are shared with other products, without any negative impact on any other product or system or itself.
        - Interoperability: the measure in which more than one system, product, or component can interchange information and utilize the exchanged information.

- *Usability*
    - It describes the extent to which different user groups utilize a product or system to conduct specific functions with relative effectiveness, efficiency, and satisfaction under specific situations. This can be broken down into the following sub-characteristics:
        - Appropriateness recognizability: the intended users can properly identify if the product or system satisfies their needs.
        - Learnability: the extent to which the users of the product or system can understand and learn how to use it with relative effectiveness, efficiency, freedom from risk, and satisfaction under different situations.
        - Operability: the degree to which a product or system has characteristics that allow its users to operate and control it easily.
        - User error protection: the product or system's capability to prevent users from committing errors.
        - User interface aesthetics: the measure in which the user interface of a product or system provides a pleasing and satisfying experience and interaction with the user.
        - Accessibility: the extent to which the product or system is usable for a wide variety of user groups and achieves specified goals under different situations.

- *Reliability*
    - It describes the capability of a system, product, or component to perform functions under different specified conditions in a set amount of time. This can be broken down into the following sub-characteristics:
        - Maturity: the system, product, or component meets the required reliability under normal operating conditions.
        - Availability: the component, product, or system is accessible and operable when required to be used.
        - Fault tolerance: the extent to which a component, product, or system can function properly, regardless of hardware or software issues.
        - Recoverability: the extent to which the product or system can directly recover the data and return to normal operating conditions when an event of interruption or failure occurs.
- *Security*
    - It describes the ability of the product or system to protect information and data from unauthorized access by implementing proper data access appropriate to the level of authorization a product or system has. This can be broken down into the following sub-characteristics:
        - Confidentiality: specifies that the information and data can only be accessed by entities with the appropriate authorization level.
        - Integrity: specifies that a component, product, or system can prevent malicious intentions such as unauthorized access to computer programs or data modification.

        - Non-repudiation: specifies that the component, product, or system can record actions or events to prove that such actions and events have taken place in order for those actions and events to be not repudiated in the future.
        - Accountability: specifies the degree to which an action of an entity can be uniquely traced back to them.
        - Authenticity: specifies the degree to which the identity of the subject or resource who claimed it can be verified.

- *Maintainability*
    - It describes the ability of the product or system to be modified, improved, adapted, and corrected to the changes in environment and requirements. This can be broken down into the following sub-characteristics:
        - Modularity: the ability of the product or system to be broken down into distinct components so that when one change is done to a component, it creates minimal impact on other distinct components.
        - Reusability: the ability of an asset to be reused in more than one system or product or use it to build other assets.
        - Analyzability: the ability of a product or system to identify the influence of an intended change effectively and efficiently on one or more components of a product or system, identify the deficiencies and causes of failures, or identify the parts to be modified.

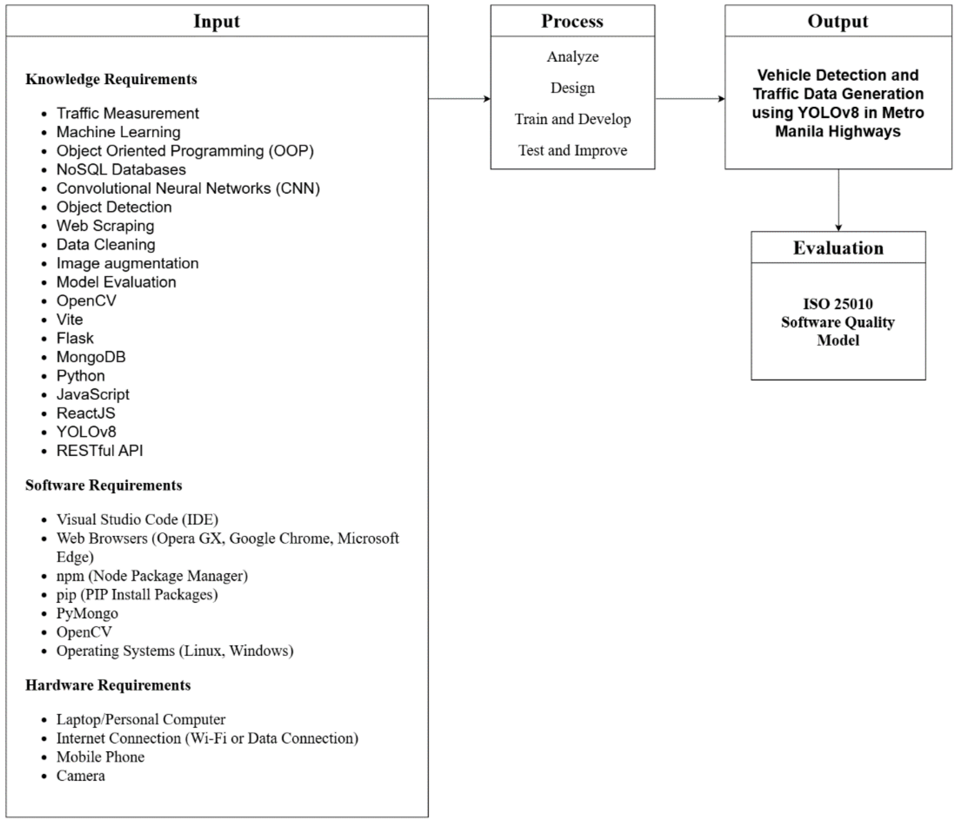
        - Modifiability: the ability of a product or system to modify itself or its components effectively and efficiently without creating any detrimental effects or lowering the quality of the product or system.
        - Testability: the ability of a product or system to efficiently test criteria for a system, product, or component and how effectively these tests can verify if the set criteria have been fulfilled.

- *Portability*
    - It describes the extent of effectiveness and efficiency of a component, product, or system relative to its ability to be transferred from its hardware, software, or another operational or usage environment to another. This can be broken down into the following sub-characteristics:
        - Adaptability: the extent to which a product or system adapts effectively and efficiently to the ever-changing hardware, software, or other operational or usage environments.
        - Installability: the extent to which a product or system is properly installed and uninstalled effectively and efficiently in a set environment.
        - Replaceability: the extent to which a product or system can be switched to another specific software product or system to perform the same functions under the same purpose and environment.

These eight characteristics that the ISO/IEC has set become the cornerstone of the product quality evaluation system. It continues to set quality standards for software products and should employ all aspiring software developers and researchers (*ISO 25010*, n.d.).

Conceptual Model of the Study

Figure 4*. Input-Process-Output Diagram of the Web Application Project*



The conceptual Model of the Study is presented below using an Input-Process-Output (IPO) diagram. The IPO diagram depicts a brief overview of the flow and concept of the project.

Input

The input block encapsulates the prerequisite knowledge, software, and hardware requirements that the researchers should have an adequate understanding or necessity to learn to design and develop the project. The knowledge requirements specify fields of information such as traffic measurement, machine learning, OOP, NoSQL databases, CNN, object detection, web scrapping, data cleaning, image augmentation, model evaluation, MongoDB, Python, JavaScript, ReactJS, YOLOv8, and RESTful API. The software requirements specify the software tools and technology required to design and develop the project. These software requirements are Visual Studio Code (IDE), web browsers, npm, pip, PyMongo, and OpenCV Operating Systems. The hardware requirements specify the required tangible or physical tools required to design and develop the project. These hardware requirements are a laptop or personal computer, internet connection, mobile phone, and camera.

Process

The process block is the summary of the flow of the development of this project. These processes are:

Analyze is the formulation of the problem and its solution. This is where an investigation of the problem is conducted to determine if the project is feasible. A series of studies and data collections are done to solidify the feasibility of this project. Literature is collected, and various software and hardware tools are listed as needed to proceed with the project's development.

Design is the process in which the ideation of the project's system and software architecture is conceptualized. This involves the initial concept of how the UI will look and how the process of data is conducted. Design of the flow of functions and data. Essentially, this process encapsulates the initial planning and conceptualization of the core functionalities of the project.

Train and develop is the process in which the project is developed. The AI model that facilitates the detection of vehicles given a video input will then be trained. This is done to enable the AI model to correctly identify a vehicle when given video input and to categorize it properly. The API endpoint, database that will store the traffic data, and other core functionalities of the project will be developed using the software tools and technologies specified under the software requirements in the *input* block.

Test and improvement are done after all the core functionalities and set goals for training and development are met. The project will be subjected to various testing procedures to ensure that it meets the quality standards and properly executes the specified functional requirements the project entails. The results from the tests will then be used as a basis for which areas of the project require improvement or additional features to be added.

Output

The output block represents the *Vehicle Detection and Traffic Data Generation using YOLOv8 in Metro Manila Highways* as the final product. It allows API requests for the AI detection model, generation of traffic data, and upload and storage of traffic data, allowing the recorded data to be available to the public.

Evaluation

The evaluation block specifies the utilization of the ISO 25010 Software Quality Model, where ISO 25010 is the amended edition of ISO 25010. It defines the quality characteristics that should be noted when evaluating a software system. The researchers intended to evaluate the project under the characteristics of *Functional Suitability, Performance Efficiency, Usability*, *Reliability,* and *Maintainability*.

Operational Definition of Terms

- **Bounding Line** – a line built from two user-specified points that serve as a counting line whenever the detected vehicles pass through it.
- **Traffic Flow -** refers to the movement of vehicles or pedestrians along a specific route or within a particular transportation network. It encompasses the patterns, volume, speed, density, and behavior of the movement of vehicles or individuals within a given area or roadway system.
- **Traffic Data** – refers to the information collected and analyzed about the movement of vehicles, pedestrians, or transportation systems within a specific area or along particular routes. This data includes various metrics such as the volume of vehicles passing through a location, vehicle speed, density of traffic, types of vehicles, traffic patterns, congestion levels, travel times, and other relevant information.
- **Road Network** –refers to the interconnected system of roads, streets, highways, and transportation routes within a geographic area, region, or country. It comprises the infrastructure designed to facilitate the movement of vehicles, pedestrians, and goods from one location to another.
- **pip** – stands for *"pip Install Package,"* developed in 2008, is a standard packages installer tool for Python. It is a powerful and user-friendly tool allowing users to install and manage Python packages via various commands. It has now been preinstalled in Python 2.7.9, Python 3.4, and later versions.
- **npm** – stands for *"Node Package Manager,"* a package manager and software register that also allows its users to find, build, and manage code packages. It now contains over 800,000 packages for various applications such as front-end, robotics,

web development, mobile apps, etc. It now serves as the primary support for JavaScript packages.

- **Computer Vision -** enables computers to extract meaningful information from digital images and videos, making informed decisions or recommendations. Unlike human vision, which can easily recognize images, computer vision rapidly learns to perceive, observe, and understand using cameras, data, and algorithms. This accelerated process allows systems to analyze thousands of products or processes per minute, surpassing human capabilities in tasks like defect detection in manufacturing.

Chapter 3

METHODOLOGY

This chapter outlines the research methodology of the study, which comprises segments on project design, project development, operational and testing procedures, and the evaluation process.

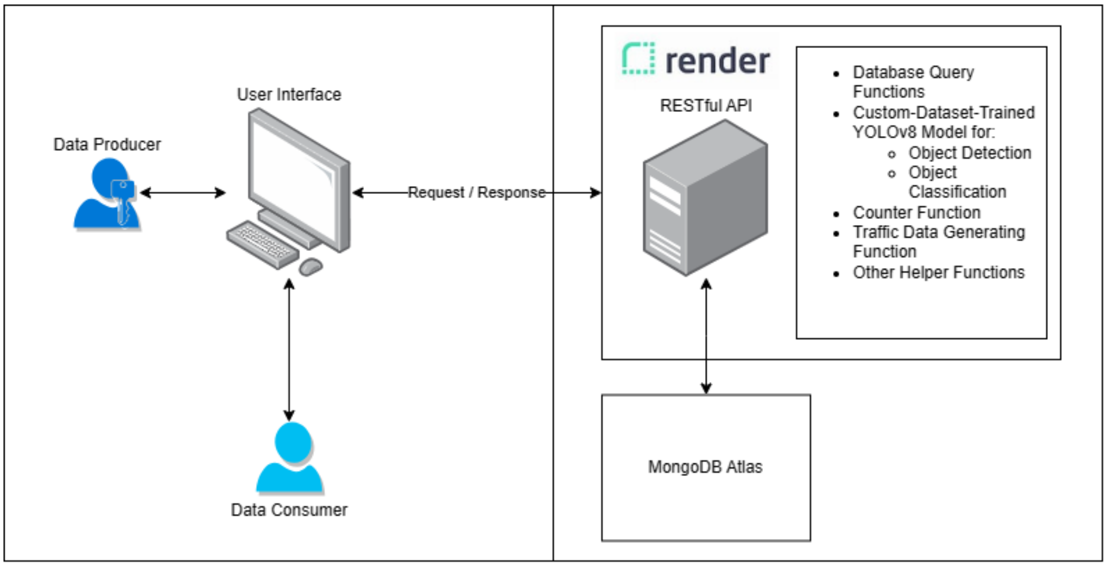
Project Design

The researchers of this study designed a web application with a user interface to facilitate access to the RESTful API endpoint. These endpoints enable users to retrieve real-time traffic data generated from live or file inputs. The web application also allows users to add their own live camera input, such as a webcam, to the system. The web application will then use YOLOv8, the eighth iteration of the YOLO algorithm, to detect, track, and count vehicles in the video stream and generate traffic data accordingly. The traffic data includes the count, type, and time of entry and exit of each vehicle. The web application will store the traffic data in a MongoDB database and update it in real-time. Users can access the traffic data through the API endpoint or the user interface. The user interface will display the traffic data in a graphical way, such as charts, tables, and maps. The web application aims to provide a convenient and efficient way for users to monitor and manage traffic conditions in different locations.

System Design

The system architecture in Figure 5 illustrates the system model and how two different types of users can use it. A user who wants to produce data can connect their camera to the system through the web application's user interface, built with web technologies such as HTML, CSS, and JavaScript, as well as frameworks and tools such as React and Vite.

***Figure 5**. System Architecture of the System Model*

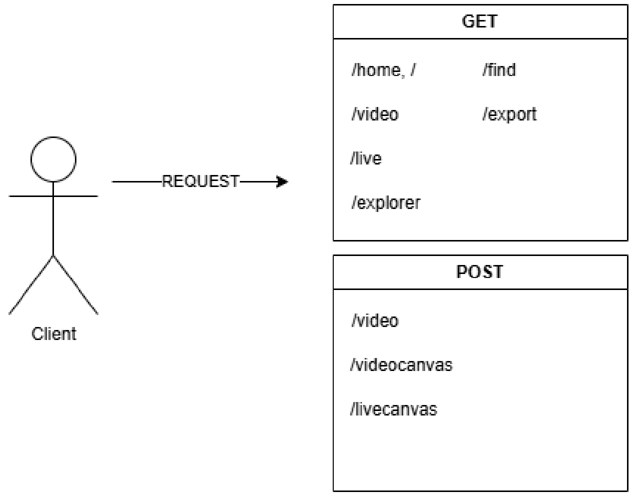


When the camera is connected, the system will start generating data based on the vehicles passing the camera's view. The data will be stored in the database and can be accessed by another type of user who wants to consume data. The data-consuming user can view the traffic data using the same web application.

Software Design

Figure 6 shows the Use Case Diagram of the system API. It presents the project's scope by determining the route and endpoint that the system user can expect. The route and endpoint illustrate the API's anticipated functionalities, behavior, and requirements.

Figure 6*. Routes and Endpoints*



The system API will have the following endpoints:

- "get /home, /"
- "get /video"
- "get /live"
- "get /explorer"
- "get /find"
- "get /export"
- "post /video"
- "post /videocanvas"
- "post /livecanvas"

Assuming the routes for endpoints have been navigated through the web application's home page, these endpoints correspond to the scope of the study's functionalities.

Project Development

The custom vehicle detection model was developed using the Ultralytics library and the YOLO algorithm architecture. The YOLO algorithm architecture consists of three main parts: "Backbone," "Neck," and "Head." The "Backbone" uses the "New CSP-Darknet53" structure, a modification of the Darknet architecture used in previous versions of the algorithm. The “Backbone” is the main body of the network. The “Neck” utilizes “SPPF” and “New CSP-PAN” structures to connect the backbone and the head. The “Head” is responsible for the final output.

The following steps were taken to develop the real-time vehicle detection model:

a. Images of vehicles prevalent in Philippine roads containing nine classes were scraped.
b. The images were manually labelled using Roboflow’s labelling tool.
c. The dataset was split into train, validation, and testing sets in the 75%, 15%, and 10% ratios, respectively.
d. A pre-trained model of YOLOv8 was used as a base for the vehicle detection model.
e. The model was trained, and the parameters such as epochs, image size, patience, and batch were adjusted.
f. The model was validated and evaluated using mean average precision (mAP), confusion matrix, and training and validation loss.
g. Inferences are created with the model.
h. The traffic data is deployed and can be accessed and exported through the API.

The images were manually labelled using Roboflow’s labelling tool to ensure the model was trained on accurate data. Splitting the dataset into training, validation, and testing sets helps to ensure that the model is not overfitting to the training data and can generalize well to new data.

Using a pre-trained model like YOLOv8 as a base for the vehicle detection model is a great way to speed up the training process and improve the accuracy of the model. Adjusting the parameters such as epochs, image size, patience, and batch can help to fine-tune the model and improve its performance.

Evaluating the model using metrics such as confusion matrix, training and validation loss, and mean average precision (mAP) is also important. These metrics are important for assessing the performance of the model.

The RESTful API was developed using the Flask framework for Python. The simplicity of the framework enables the researchers to set up endpoints for the application with ease. MongoDB, a NoSQL database, is the storage used for the generated traffic data. The following steps were taken to develop the API and web application:

1. Set up a virtual environment to create an isolated set of dependencies from other projects.
2. Install Flask, Ultralytics, and other dependencies using PIP, a Python package manager.
3. Create the main application that contains references to the vehicle detection model and methods responsible for inferences.
4. Program the routes and endpoints of the API.
5. Program the middleware methods that are responsible for logic and database queries.
6. Setup the database storage in MongoDB Atlas.
7. Create the view template for each route.

8. Create the controller template for each route.
9. Run the program and test using a video file or input camera.

Operation and Testing Procedure

Operation Procedure

The following steps can be taken to operate the web application and API locally:

1. Create a Python virtual environment for the project.
2. Clone the repository containing the codebase of the application.
3. Install the dependencies in “requirements.txt”
4. Run the app using Flask.
5. Visit the local host URL provided by the framework.
6. Navigate the webpage to upload a video or use an input camera such as a webcam.
7. Connect to the internet to upload the generated traffic data into the centralized database in MongoDB Atlas.

Testing Procedure

The following procedure is the steps taken on how the web application, RESTful API, and vehicle detection model were tested.

A. Testing the RESTful API

To find and debug the errors, as well as ensure the functionalities of the features, the RESTful API was tested using Postman and different programming languages (see Figures 19 - 22).

B. Testing the Web Application

To identify and debug the errors, as well as ensure the functionalities of the user interface, the web application was tested using manual testing by the researchers. The test cases, steps undertaken, and their expected output can be seen in Table 2.

Table 2. *Operations and Testing Procedure for the Web Application*

Test Cases	**Step Undertaken**	**Expected Results**
Access the *Home* landing page	1. Access the web application via the URL provided	Display the intended home landing page
Access the *Video Page*	1. Access the web application via the URL provided. 2. From the landing page, click the *Video* link in the navbar.	Display the Video page
Submit a *Video* forms	1. Access the web application via the URL provided. 2. From the landing page, click the *Video* link in the navbar. 3. From the *Video page*, click the Choose *file* button.	Display file explorer window where the user is allowed to choose a video file to submit
Submit *Video* details forms	1. Access the web application via the URL provided. 2. From the landing page, click the Video link in the navbar. 3. From the *Video page*, click the *Choose file* button. 4. Submit video.	Display the submitted video, while also showing the bounding line

	5. Input the information related to the uploaded video, such as its date of record, time of record, address, and specified bounding line.	
Track, detection, classification, and count of the vehicles using submitted video	1. Access the web application via the URL provided. 2. From the landing page, click the Video link in the navbar. 3. From the *Video page*, click the *Choose file* button. 4. Submit video. 5. Input the information related to the uploaded video, such as its date of record, time of record, address, and specified bounding line. 6. Submit the form.	Display the submitted video while also showing the bounding line and object inferences done with the developed custom model
Access *Live page*	1. Access the web application via the URL provided. 2. From the landing page, click the *Live* link in the navbar.	Automatically connect the available camera connected to the user's device. Get a screengrab from the detected camera.
Live page submit	1. Access the Live page via the navbar from the home page. 2. Input the address of the location of the live camera, and the	Display what the connected camera sees, with the bounding line drawn by the user

	two points of the bounding line.	
Track, detection, classification, and count of the vehicles using connected live camera	1. Access the Live page via the navbar from the home page. 2. Input the address of the location of the live camera, and the two points of the bounding line. 3. Submit the forms generated from *Live page submit*	Display the connected live camera, while also showing the bounding line, and object inferences done with the developed custom model
Access the *API Explorer* page	1. Access the web application via the URL provided. 2. From the landing page, click the Live link in the navbar.	Display the Explore API page.
Input filter queries via the API Explorer	1. Access the *API Explorer* page via the navbar from the *Home* page. 2. Input desired filter queries in the form fields displayed.	Display the API call string, with the input filters
View JSON format of the database	1. Access the *API Explorer* page via the navbar from the *Home* page. 2. Input desired filter queries in the form fields displayed. 3. Click the View JSON button.	New window pops up showing the full view of the database in JSON format
Download the CSV file containing the counted vehicles	1. Access the *API Explorer* page via the navbar from the *Home* page.	A CSV file containing the database will be downloaded

	2. Input desired filter queries in the form fields displayed. 3. Click the download CSV button.	
Download *Summary Report*	1. Access the *API Explorer* page via the navbar from the *Home* page. 2. Input desired filter queries in the form fields displayed. 3. Click the *Download Summary Report* button.	A file containing the traffic data report will be downloaded

C. Testing the Model

The vehicle detection model was tested by creating inferences using samples from the dataset, as well as new ones, and manually comparing the results from the ground truth. The results were also evaluated using metrics, such as confusion matrix, mean average precision (mAP), accuracy, and loss.

Evaluation Procedure

The researchers utilized the evaluation tool, ISO 25010, "Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) - Systems and Software Quality Models," to assess the system's acceptability.

To evaluate the acceptability of the developed project, the following procedures are followed:

1. Purposively invited 30 respondents divided between 15 non-expert individuals and 15 expert individuals with a background in the IT/CS industry.

2. A Google Form link, embedded with a video presentation of the demonstration of how the project works, use case examples such as when it detects the vehicle and its type through an input video and how the API functionalities will be requested, will be sent to the evaluators. The Google form also contains a questionnaire that encompasses the ISO 25010 software characteristics, which are used as a standard for evaluating the acceptability of the developed project. Specifically, the included characteristics are *Functional Suitability, Performance Efficiency, Compatibility, Usability, Reliability, Security,* and *Maintainability*. The evaluators will then be asked to evaluate the project based on the questionnaire described using a 4-point Likert Scale shown in Table 3.
3. The duly filled evaluation sheets are gathered and processed, and the data is tabulated to identify the mean ratings.
4. The weighted mean ratings will then be interpreted for their corresponding qualitative interpretation, as shown in Table 3.

Table 3*. Likert Scale and its corresponding weighted mean rating and qualitative interpretation*

Scale	Range	Qualitative Interpretation
4	3.26 – 4.0	Highly Acceptable
3	2.51 – 3.25	Very Acceptable
2	1.76 – 2.5	Acceptable
1	1.00 – 1.7	Not Acceptable

Chapter 4

RESULTS AND DISCUSSION

This chapter contains the project description, structure, capabilities and limitations, and results of the project's testing and evaluation.

Project Description

This study developed a vehicle detection and traffic data generation web application using YOLOv8, which can help users gather traffic-intensive data. This is done by detecting, tracking, classifying, and counting the detected vehicles. Additionally, it has an API endpoint that allows users to access the database via a GET request, which displays the exported database in JSON format.

The web application contains various pages for different functionalities such as *Home*, *Video,* and *Live*. *The home* page includes information regarding the project and a couple of screenshots displaying the project. *The video* page allows users to upload and submit a video that will be used by the web application to detect, classify, and count vehicles. *The live* page allows users to use a live camera as input for detecting, classifying, and counting vehicles.

The data can be exported via a GET request API action */find,* which takes a dynamic number of parameters such as address, date, time, and type. The response body to this request is in JSON format, containing the traffic data filtered by the query parameters in the GET request.

Figure 7*. Web Application Landing Page*

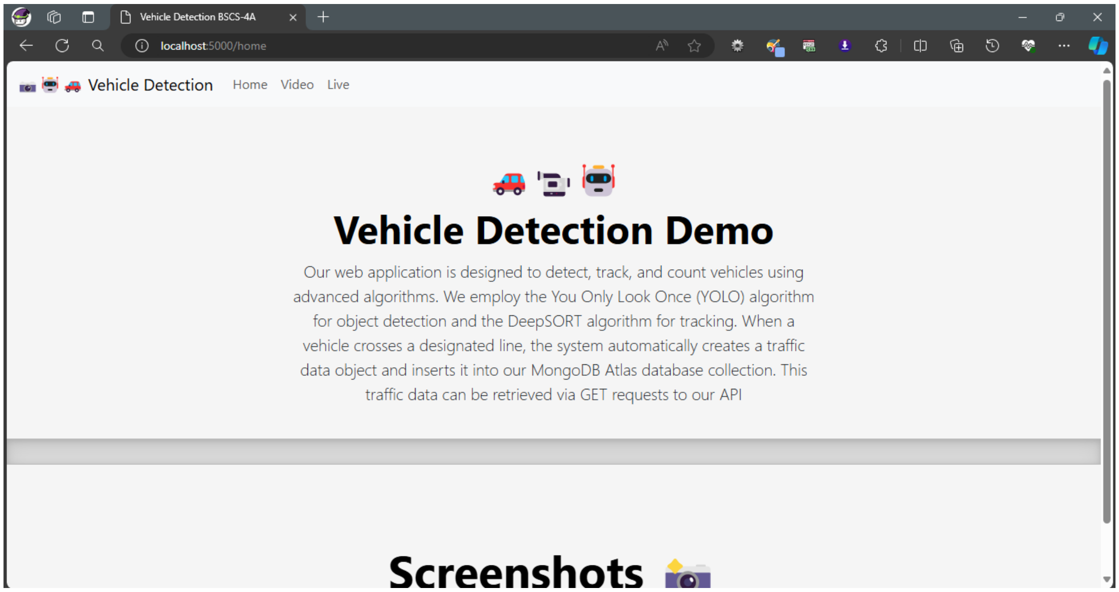
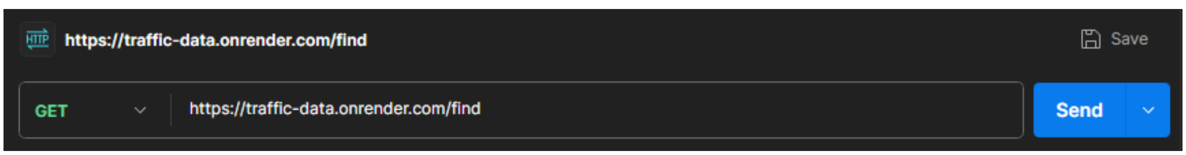


Figure 8*. API URL Request*



Project Structure

The web application can be divided into four main routes '/home', '/video', '/live', and '/export'. The '/home' route serves as the landing page of the project. It displays a brief description of the project. It also shows a few screenshots of the project, providing more previews of the project.

The /video' route displays a page in which users are asked to upload a video, as shown in Figure 10, that will be used for vehicle detection and traffic data generation. Once a video is uploaded, a helper route '/canvas' will then grab a screen capture of the uploaded video, which will help the users draw the bounding line on the video. It will also ask for required information, such as the video's date, time, and address, as shown in Figure 11. After submission, the '/video' route will display the inferences generated with the researchers' custom-trained model, as shown in Figure 12.

***Figure 9**. Video page asking users to upload a video file.*

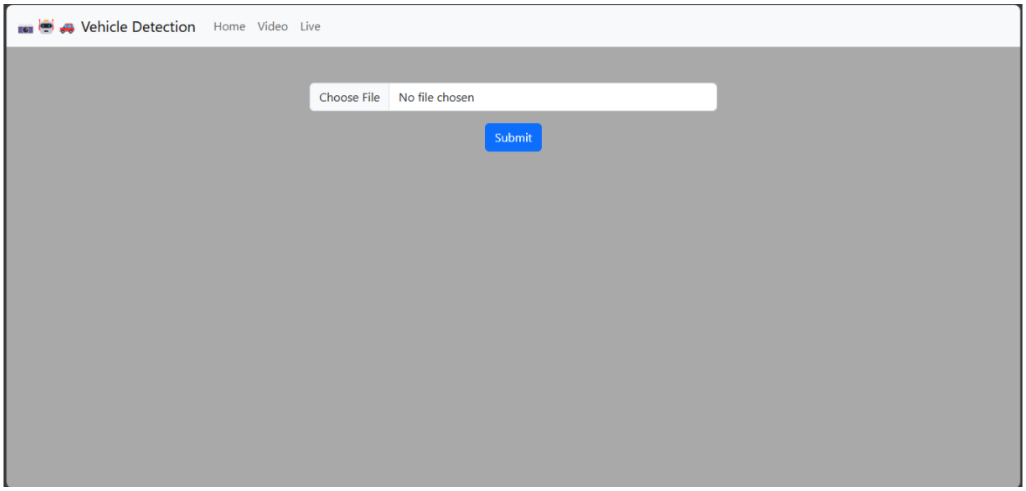


***Figure 10**. Video file upload*

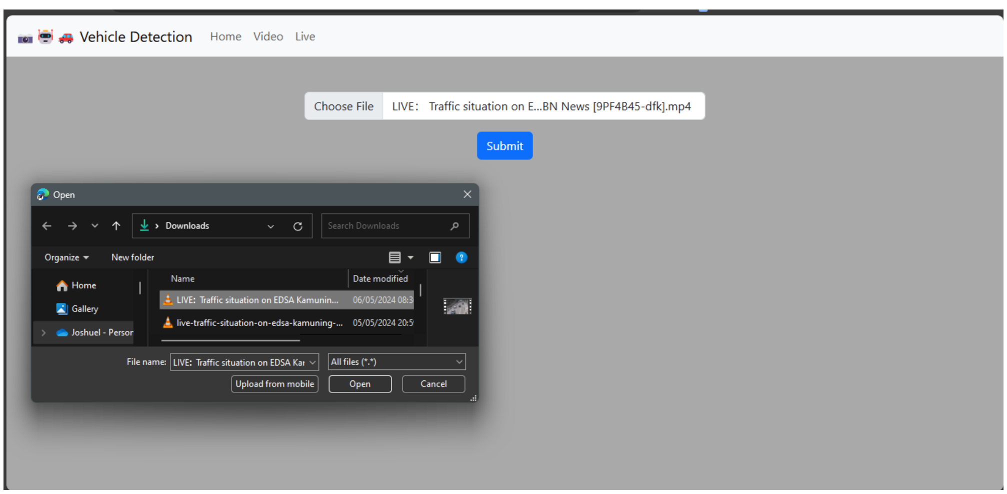


***Figure 11**. Helper route '/canvas'*

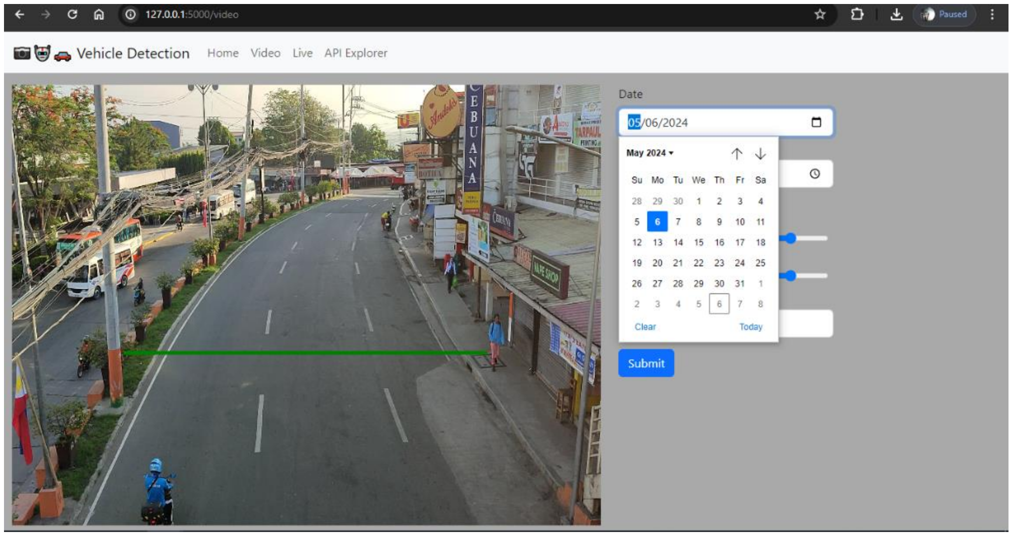
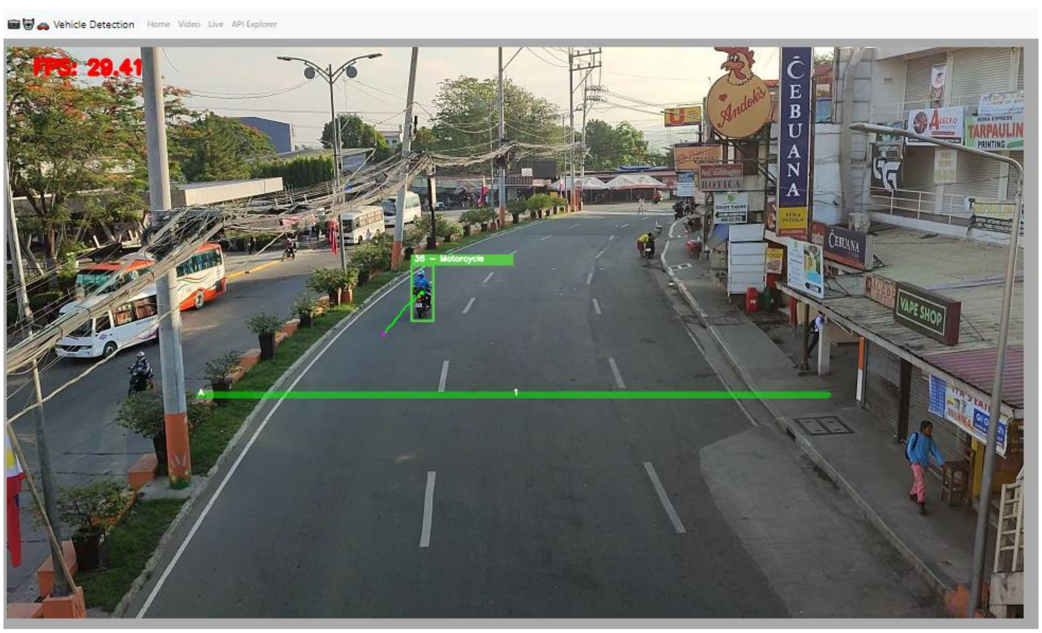


***Figure 12**. '/video' route where a video has been uploaded and is currently doing its tasks.*



The '/live' route renders a page where users can use the connected camera as input for vehicle detection and traffic data generation, as shown in Figure 13. With the help of the '/canvaslive' route, the first frame of the connected live camera will immediately be captured, which will be used to help the users draw the bounding line. It will then ask for required information, such as the address of the current location of the live camera, while the input for the date and time will be fetched automatically to the system's corresponding one. Once submitted, the '/live' route displays the inferences generated with the researchers' custom-trained model, as shown in Figure 14.

***Figure 13**. '/canvaslive route for Vehicle Detection and Traffic Data Generation*

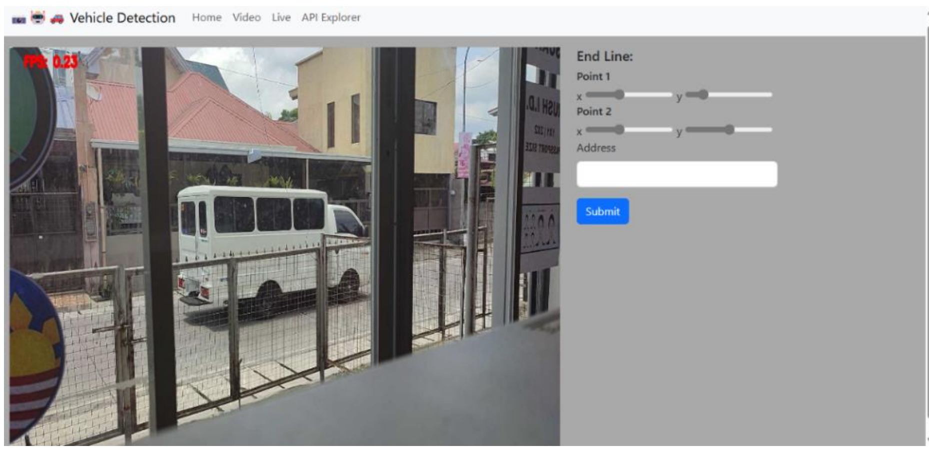
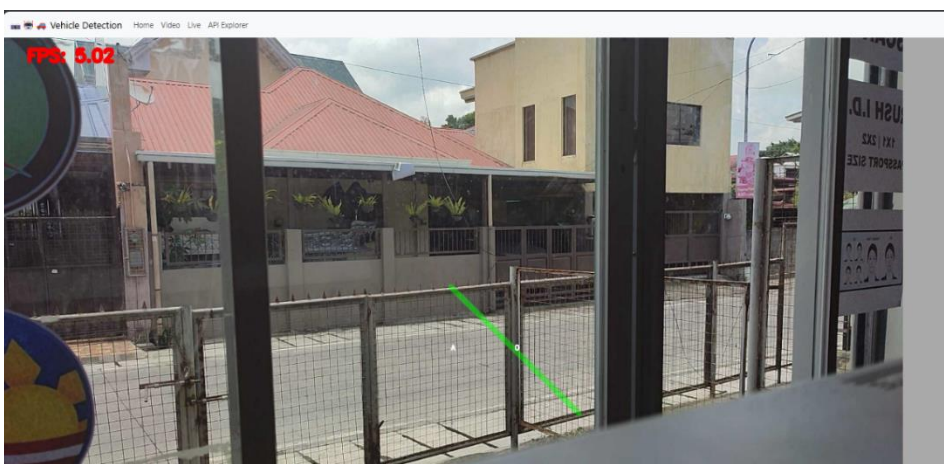


***Figure 14**. '/live' route in action gathering inferences via the live camera connection.*



The '/export' route presents a page allowing users to download the data, which will be exported in a csv format, as shown in Figure 15. The downloaded data can be filtered dynamically with arguments such as address, class (kind of vehicle), date, and type (public or private vehicle) with its query.

Figure 15. *'/export' route preview*

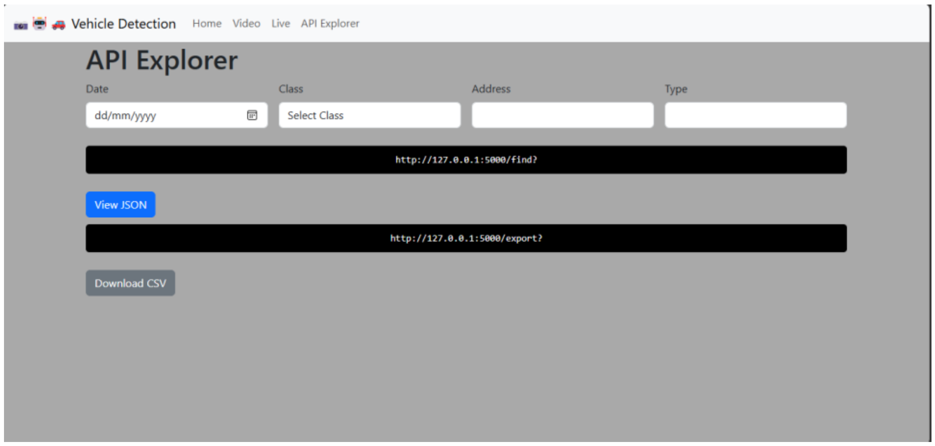


***Figure* 16**. '/export' route with an address filter added

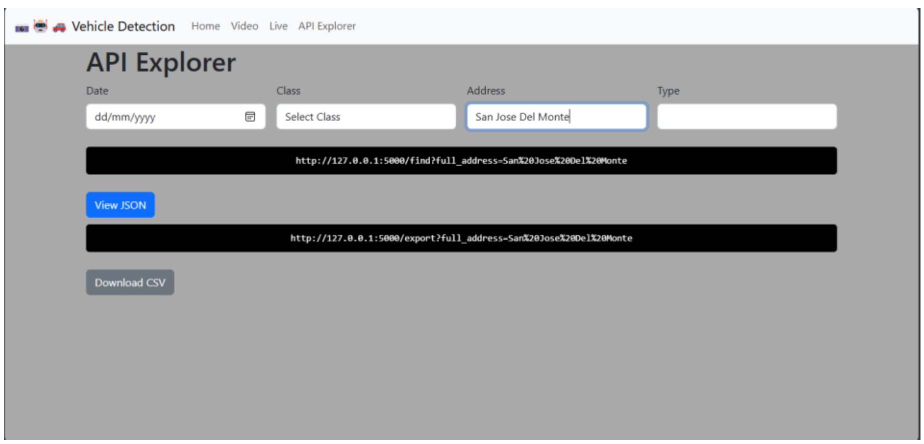


Figure 17. *'/export' route with a date filter added*

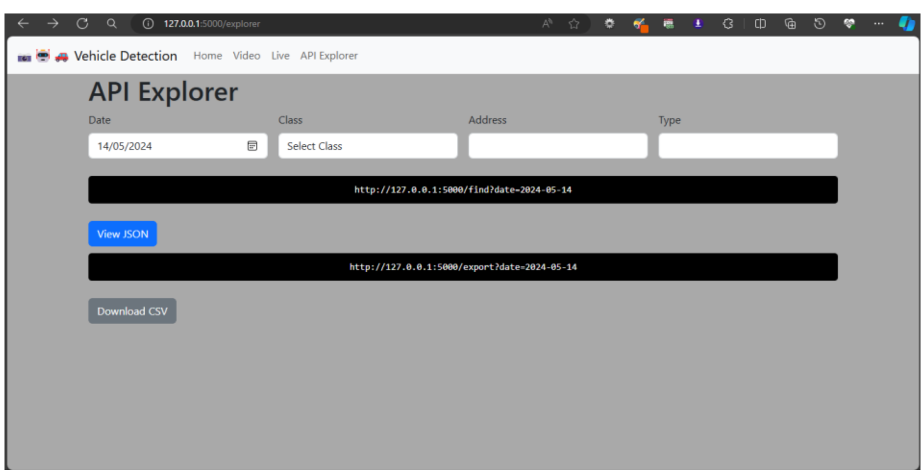
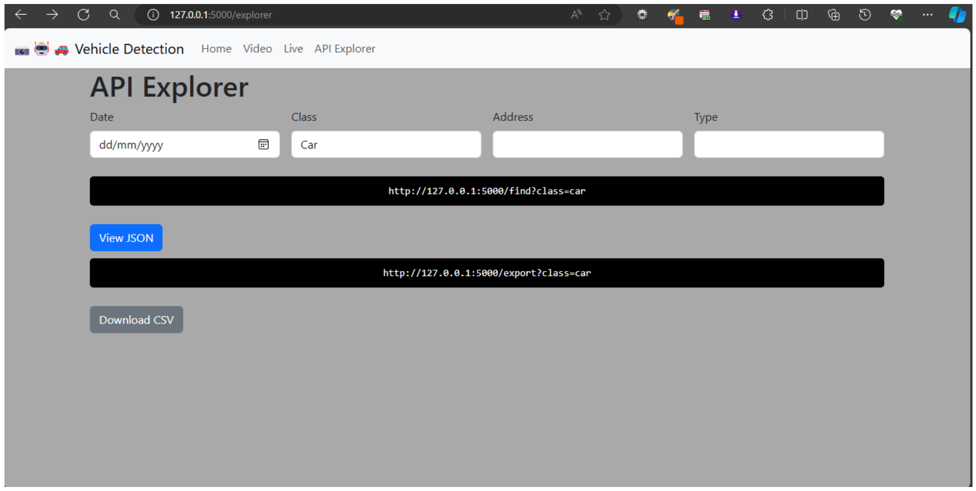


***Figure 18**. '/export' route with class filter added.*



The API GET request to endpoint /find allows developers to fetch the recorded data from the MongoDB database. They can include filters within their query via dynamic arguments such as address, class (kind of vehicle), date, type (public or private vehicle), and time.

***Figure 19**. POSTMAN test for the API query with a filter for class=car*

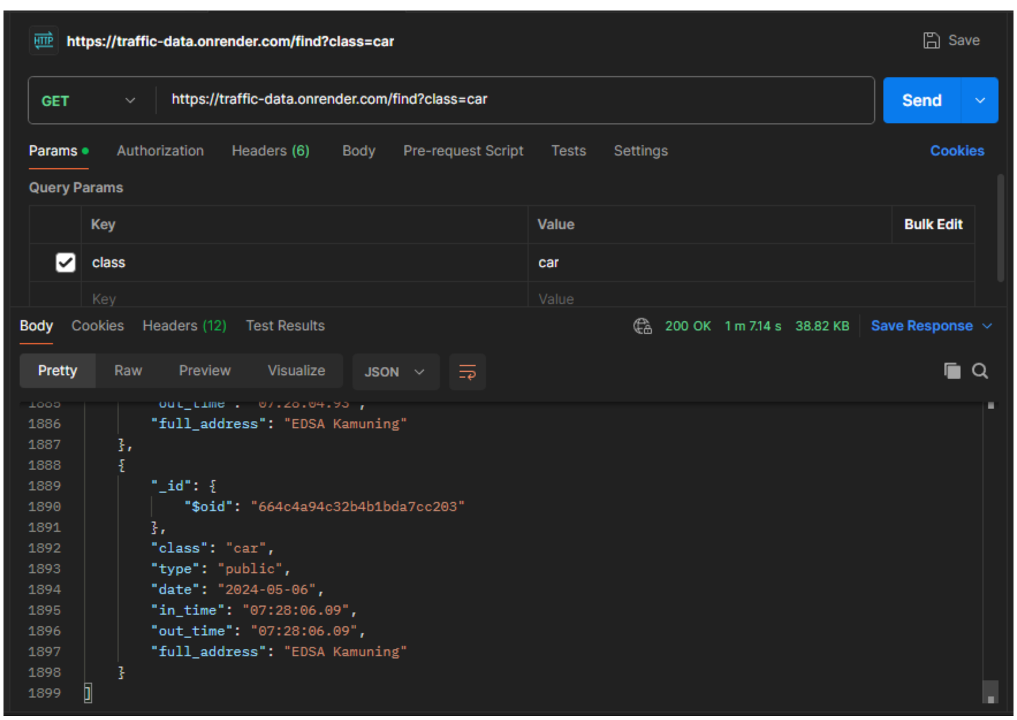


Figure 20. *API use case in JavaScript*

```javascript
async function fetchTrafficData(url) {
    try {
        const response = await fetch(url);
        if (!response.ok) {⋯
        }
        const data = await response.json();
        return data;
    } catch (error) {⋯
    }
}
const url = "https://traffic-data.onrender.com/find?class=%22Car%22";
fetchTrafficData(url)
```

Figure 21. *API use case in Python*

requestsample.py

```python
import requests

url = "https://traffic-data.onrender.com/find?"

response = requests.get(url)

if response.status_code == 200:
    data = response.json()
    for obj in data:
        print(obj)
else:
    print("Error fetching data :(")

```

Figure 22. *API use case in PHP*

```php
<?php
    $url = 'https://traffic-data.onrender.com/find?class=car';
    $jsonString = file_get_contents($url);
    $jsonObject = json_decode($jsonString);
    var_dump($jsonObject)
?>
```

Project Capabilities and Limitations

The following are the capabilities of the developed web application for vehicle detection and traffic data generation:

Web Application:

1. The web application can detect vehicles and identify their class. Specifically:
    a. Bicycle
    b. Bus
    c. Car
    d. Jeep
    e. Motorcycle
    f. Tricycle
    g. Truck
2. The web application tracks vehicles as they cross the counting line and records the data in a centralized cloud database.
3. The web application allows users to customize the vehicle counting line.
4. The web application allows the connection of a camera to feed the system with live video input, which is used for detecting vehicles.
5. The web application can output generated traffic data in JSON and CSV format using REST API.

Hosted API:

1. The hosted API can output generated traffic data in JSON and CSV format using REST API.
2. The hosted API's page can be visited at (https://traffic-data.onrender.com/)

The following are the limitations of the developed web application for vehicle detection and traffic data generation:

Web Application:

1. The web application is currently not hosted on the web.
2. The developed custom YOLOv8 detection model can only infer about 80 detections per frame.
3. The web application's performance is based on the resources of the machine it is running into.

Test Results

A. API Testing Results

To test the API's proper responses e.g., returning code 200, Postman was employed, as seen in Figure xx, the GET method is selected, and the URL containing the query can be seen. The URL query string *https://traffic-data.onrender.com/find* fetches all the data inside the database. Several GET parameters can be added to filter the downloaded data. Such parameters are:

- *https://traffic-data.onrender.com/find?address=<value>*
- *https://traffic-data.onrender.com/find?class=<value>*
- *https://traffic-data.onrender.com/find?date=<value>*
- *https://traffic-data.onrender.com/find?time=<value>*
- *https://traffic-data.onrender.com/find?type=<value>*

Figure 23*. API Postman Testing*

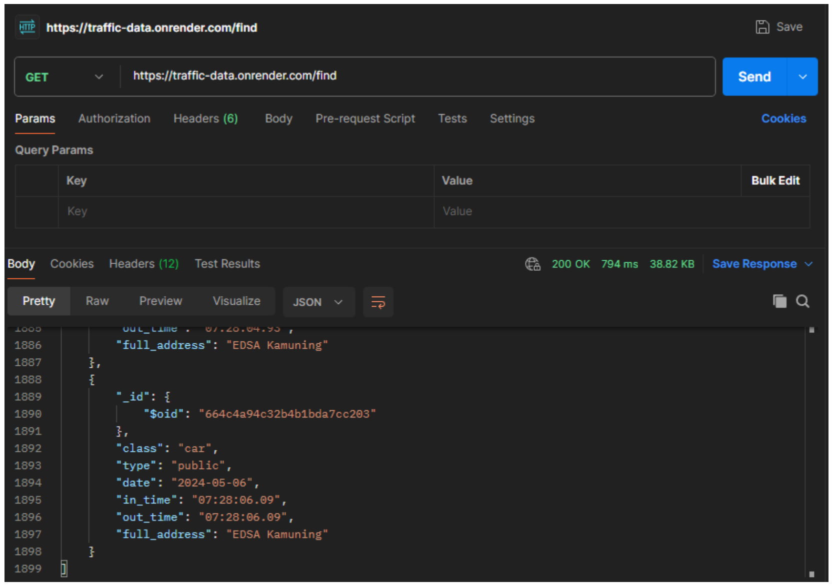


Figure 24*. API Postman testing with filter for class=car*

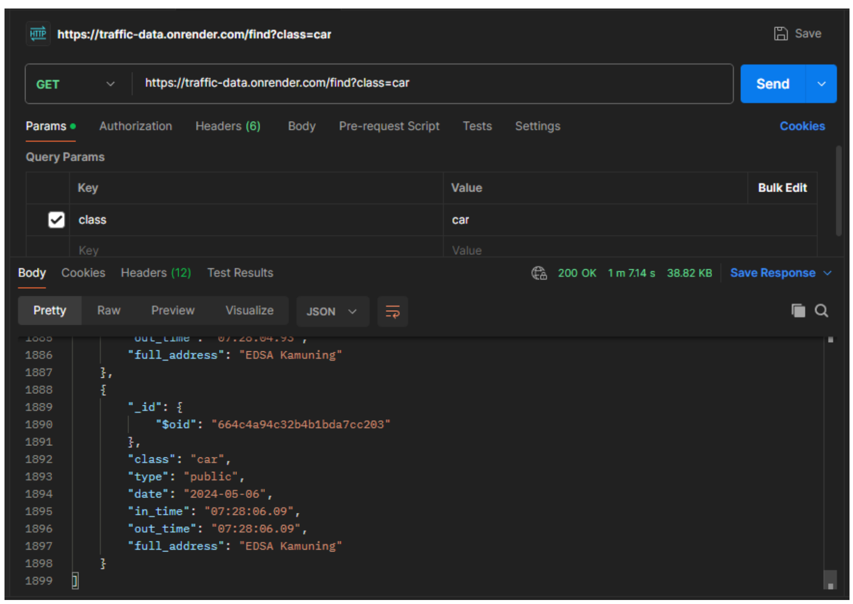
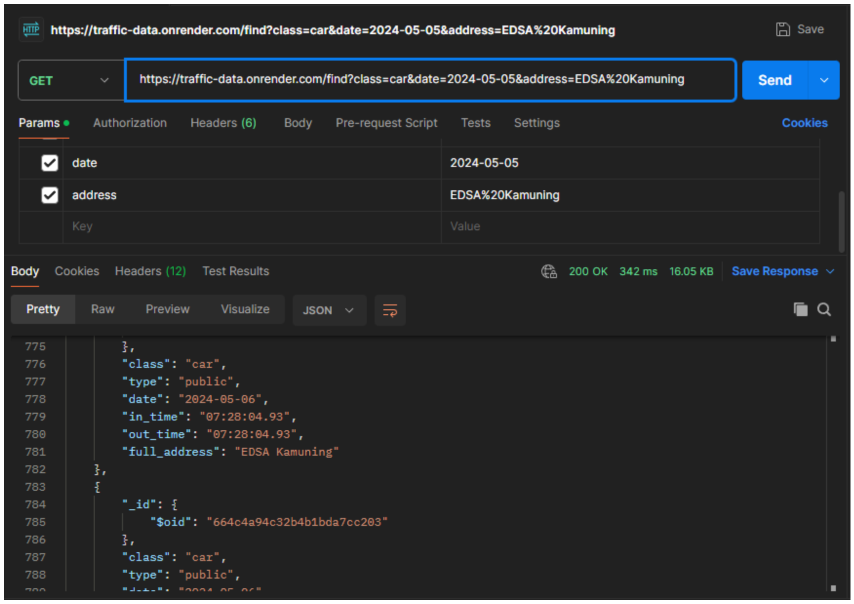


Figure 25. *API POSTMAN testing with filters with Address*



B. Operations and testing results

Table 4. *Operations and Testing Results*

Test Cases	Step Undertaken	Observed Results
Access the *Home* landing page	1. Access the web application via the URL provided	The *home* page has been displayed accurately and properly
Access the *Video Page*	1. Access the web application via the URL provided. 2. From the landing page, click the *Video* link in the navbar.	The *video* page successfully loaded and displayed the intended forms.
Submit a *Video* forms	1. Access the web application via the URL provided. 2. From the landing page, click the *Video* page, click the *Video* link in the navbar. 3. From the *Video page*, click the *Choose file* button.	The *Choose file* button properly redirected the user to upload a video file by displaying a new file explorer

Submit *Video* details forms	1. Access the web application via the URL provided. 2. From the landing page, click the Video link in the navbar. 3. From the *Video page*, click the *Choose file* button. 4. Submit video. 5. Input the information related to the uploaded video, such as its date of record, time of record, address, and specified bounding line.	A screengrab of the first frame of the submitted video has been displayed. The bounding line has also been displayed properly and is changing dynamically as the values of the two points are manipulated. Other form fields have also been displayed properly
Track, detection, classification, and count of the vehicles using submitted video	1. Access the web application via the URL provided. 2. From the landing page, click the Video link in the navbar. 3. From the *Video page*, click the *Choose file* button. 4. Submit video. 5. Input the information related to the uploaded video, such as its date of record, time of record, address, and their specified bounding line. 6. Submit the form.	The submitted video is played. Inferences of the developed model alongside the set bounding line by the user is displayed
Access *Live page*	1. Access the web application via the URL provided. 2. From the landing page, click the *Live* link in the navbar.	A screengrab of the first frame captured by the connected live camera has been displayed alongside the required form fields and bounding line that can be dynamically modified

		via the values of two points
Live page submit	1. Access the Live page via the navbar from the home page. 2. Input the address of the location of the live camera, and the two points of the bounding line. 3. Submit the forms.	It properly displays the dynamic change of the points of the bounding line, alongside the screengrab of the live camera and other required form fields
Track, detection, classification, and count of the vehicles using connected live camera	1. Access the Live page via the navbar from the home page. 2. Input the address of the location of the live camera, and the two points of the bounding line. 3. Submit the forms generated from *Live page submit*	It displays live footage captured by the connected camera. Inferences of the developed model alongside the set bounding line by the user is displayed
Access the *API Explorer* page	1. Access the web application via the URL provided. 2. From the landing page, click the Live link in the navbar.	The *API Explorer* page was displayed properly alongside the form fields for the filter, the API calls, and buttons for *View JSON* and *Download CSV*
Input filter queries via the API Explorer	1. Access the *API Explorer* page via the navbar from the *Home* page. 2. Input desired filter queries in the form fields displayed.	The display of the API call is changed based on the filter applied via the user input inside the form fields
View JSON format of the database	1. Access the *API Explorer* page via the navbar from the *Home* page.	A new window popped up alongside the filter input by the user in the form fields

	2. Input desired filter queries in the form fields displayed. 3. Click the View JSON button.	
Download the CSV file containing the counted vehicles	1. Access the *API Explorer* page via the navbar from the *Home* page. 2. Input desired filter queries in the form fields displayed. 3. Click the download CSV button.	A CSV file was downloaded with filters applied based on the input of the user
Download *Summary Report*	1. Access the *API Explorer* page via the navbar from the *Home* page. 2. Input desired filter queries in the form fields displayed. 3. Click the *Download Summary Report* button.	A file containing the traffic data report will be downloaded

C. Model Testing and Evaluation

A custom object detection model is developed and trained with YOLOv8, with a dataset scrapped from the internet and annotated manually via Roboflow. The result of the evaluation of the custom object detection model is presented in Figure 26. The custom-trained model, trained for 158 epochs, showed the true positive (TP) predictions and the true negative, false positive, and false negative predictions of the model. It highlights the high TP predictions where the class bicycle has the highest TP prediction score at 0.92. Followed by classes bus and jeep tied at 0.91, motorcycle at 0.88, truck at 0.86, tricycle at 0.83, and at last the car with a score of 0.71.

Figure 26. *Custom Trained Model Confusion Matrix*

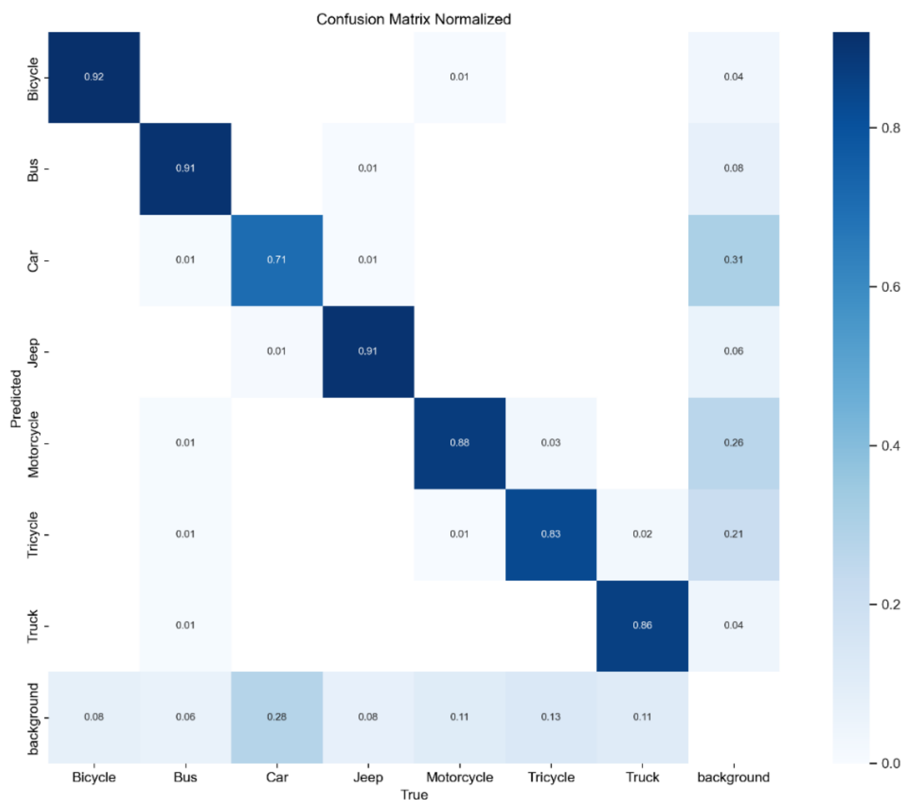


Figure 27 shows the high precision, recall, and mean Average Precision (mAP) across the seven classes (bicycle, bus, car, jeep, motorcycle, tricycle, and truck), where on each epoch trained, its metrics improved until it plateaued around 80th epoch in which no significant improvements where observed. It recorded an 86.76% mAP50 and a 67.48% mAP50-95. Figure 28 shows a time series graph containing the values generated by the web application and the manually counted ground truth. The graph shows the count on the y-axis and the time on the x-axis.

Figure 27*. Custom Trained Model Results*

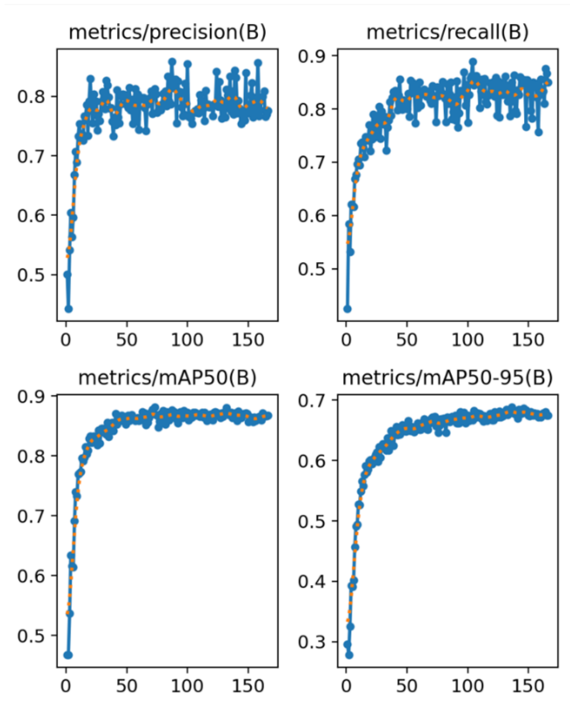
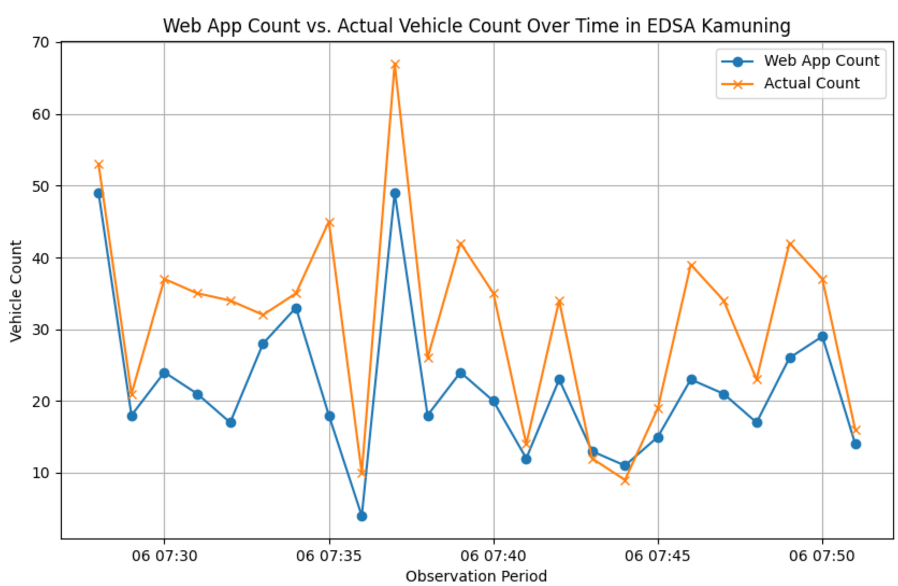


Figure 28*. Web App Count vs Actual Count (Ground Truth)*



Project Evaluation

An evaluation was done to determine the acceptability of the project. Thirty respondents were selected, divided into 15 expert individuals in the IT/CS industry and 15 non-expert individuals. Table 5 displays the summary of the respondents' responses.

Table 5. *Summary of the Evaluation Responses*

Characteristic	Mean	Evaluation
Functional Suitability		
Functional Completeness	3.93	Highly Acceptable
Functional Correctness	3.87	Highly Acceptable
Functional Appropriateness	3.93	Highly Acceptable
Functional Suitability Total Average	3.91	Highly Acceptable
Performance Efficiency		
Time Behavior	3.77	Highly Acceptable
Resource Utilization	3.80	Highly Acceptable
Capacity	3.93	Highly Acceptable
Performance Efficiency Total Average	3.83	Highly Acceptable
Usability		
Learnability	3.90	Highly Acceptable
Operability	3.87	Highly Acceptable
Usability Total Average	3.88	Highly Acceptable
Reliability		
Availability	3.80	Highly Acceptable
Fault Tolerance	3.80	Highly Acceptable
Reliability Total Average	3.80	Highly Acceptable
Maintainability		
Reusability	3.90	Highly Acceptable

Modifiability	3.83	Highly Acceptable
Testability	3.93	Highly Acceptable
Maintainability Total Average	3.89	Highly Acceptable
Total Average	3.86	Highly Acceptable

The project's evaluation, titled Vehicle Detection and Traffic Data Generation using YOLOv8 in Metro Manila Highways, based on the ISO 25010 Software quality model, has shown positive results.

Its highest rating for a criterion was recorded in "*Functional Suitability,*" with a total rating of 3.91, which evaluates to *"Highly Acceptable."* This showcases the project's exceptional performance in terms of satisfying the criterion pertaining to "Functional *Suitability."* Specifically, under its sub-criteria, such as "Functional Completeness," "Functional Correctness," and "Functional Appropriateness," it has delivered its intended purpose, which is to ensure that users can expect accurate and reliable service from its functions.

Its lowest rating for a criterion was recorded in *"Reliability"* with a total rating of 3.80, which evaluates to *"Highly Acceptable."* It has been able to demonstrate a favorably received rating where both of its sub-criterion, *"Availability*" and *"Fault Tolerance,"* has garnered a rating of 3.80, which translates to *"Highly Acceptable."* Regardless of being the lowest rated among all the criteria, it still exhibited a widely accepted ratings, meaning the project provides a reliable service in terms of its *"Availability"* and *"Fault Tolerance."*

Chapter 5

SUMMARY OF FINDINGS, CONCLUSION, AND RECOMMENDATION

This chapter encapsulates the totality of the project. It includes a summary of findings, conclusion, and recommendations.

Summary of Findings

The developed project titled *"Vehicle Detection and Traffic Data Generation using YOLOv8 in Metro Manila Highways"* has been deemed capable of detecting, classifying, and counting the inferred classes. After a series of tests and evaluations, it was concluded that the project's acceptability had been assessed. A summary of the findings regarding the project can be found after the following:

Test Results

1. **Functionality:** Operating and test results showed that the project titled *"Vehicle Detection and Traffic Data Generation using YOLOv8 in Metro Manila Highways"* demonstrated a degree of functionality that exhibits a reliable and accurate inferences with custom developed and trained model. Various test cases were executed to test the web application's functionalities, while the API function in which the database can be downloaded via an HTTP request was tested using a tool named Postman.
2. **Maintainability:** Operating and test results showed how scalable the project titled *"Vehicle Detection and Traffic Data Generation using YOLOv8 in Metro Manila Highways"* is. Based on the results, the project exhibited excellent reusability, and users could just upload a new video or change the location of the live camera. Although it did not perform as well as the reusability aspect of the project, its modifiability in

terms of how well it can scale is almost limitless. By adding more live cameras and being equipped with a more powerful computer, the scale and performance of the project can improve vastly. Lastly, its testability also covers a lot of scope as many major roads in the Philippines are already overripe in intensively harvesting their traffic data.

3. **Web Application Operations and Testing:** The operating and testing results validated the accuracy and reliability of the project's capability to execute its designated test cases properly. The project is exposed to various test cases in which steps are taken to reach a desired outcome. The project resulted in the accurate and reliable execution of its test cases. Exhibiting the project's capabilities as a data collection tool for gathering intensive vehicular volume data.

4. **Object Detection Model testing and evaluation:** The custom-developed and trained model using the YOLOv8 architecture has demonstrated high accuracy in the classification of classes. The classes in which the model is trained to detect are Bicycle, Bus, Car, Jeep, Motorcycle, Tricycle, and Truck, in which the classes recorded confidence scores of 0.92, 0.91, 0.71, 0.91, 0.88, 0.83, and 0.86 respectively. It recorded a total average of 0.86 confidence score for its true predictions. It also recorded a high overall mAP50 and mAP50-95, equal to 86.76% and 67.48%, respectively. This further showcases the high accuracy of the custom-developed and trained model in inferencing objects for vehicle detection.

Evaluation Results

1. **Functional Suitability.** The evaluators rated the Vehicle Detection and Traffic Data Generation using YOLOv8 as highly acceptable for its capacity to classify different

types of vehicles and be able to generate traffic data accurately based on industry-standard applications and mathematics.

2. **Performance Efficiency.** The evaluators rated the Vehicle Detection and Traffic Data Generation using YOLOv8 as highly acceptable in terms of its utilization and efficient use of resources (such as CPU, memory, storage, network devices, energy, materials...) in terms of its response and processing time.
3. **Usability.** The evaluators rated the Vehicle Detection and Traffic Data Generation using YOLOv8 as highly acceptable when operating and using the software itself.
4. **Reliability.** The evaluators rated the Vehicle Detection and Traffic Data Generation using YOLOv8 as highly acceptable in terms of operating the software effortlessly and its consistency and efficiency concerning handling errors without compromising its performance.
5. **Maintainability.** The evaluators rated the Vehicle Detection and Traffic Data Generation using YOLOv8 as highly acceptable in terms of its usage for other projects and its consistency in showing results regardless of how it's modified and used for other testing criteria.

Conclusion

The development of the project titled *"Vehicle Detection and Traffic Data Generation using YOLOv8 in Metro Manila Highways"* has led the researchers to derive conclusions such as:

1. The project titled "*Vehicle Detection and Traffic Data Generation using YOLOv8 in Metro Manila Highways"* successfully developed with these functions:

a. Detect vehicles and identify their class, such as bicycle, bus, car, jeep, motorcycle, tricycle, and truck.
b. Tracks vehicles as they cross the counting line and records the data in a centralized cloud database.
c. Allows users to customize the counting line.
d. Allows the connection of a camera to feed the system with live video input, which is used for detecting vehicles.
e. Output generated traffic data in JSON and CSV format using REST API.
f. Allows users to upload and submit a video file for detecting and counting vehicles.
g. Access a web page that allows users to easily generate a filter for the HTTP query string.

2. The project titled *"Vehicle Detection and Traffic Data Generation using YOLOv8 in Metro Manila Highways"* was developed using the following tools:
   a. Front-end tools and Frameworks:
      i. HTML
      ii. CSS
      iii. JavaScript
      iv. Bootstrap
   b. Back-end tools and Frameworks:
      i. Python
      ii. Flask
   c. Database Management System(s)

    i. MongoDB

  d. Cloud hosting services for the API

    i. Render

  e. Version Control GIT.

    i. Git

    ii. GitHub

3. The project titled *"Vehicle Detection and Traffic Data Generation using YOLOv8 in Metro Manila Highways"* successfully executed the test cases set in the operational and testing of the system.

4. The project titled "Vehicle Detection and Traffic Data Generation using YOLOv8 in Metro Manila Highways" was evaluated successfully by 15 experts in IT/CS fields and 15 non-experts based on ISO/SEC 25010 in terms of:

  a. Functional Suitability
  b. Performance Efficiency
  c. Usability
  d. Reliability
  e. Usability
  f. Maintainability

Recommendations

The following recommendations are suggested for further improvement of the developed web application:

1. **Client-side inference:** The speed at which frames are processed per second depends significantly on the computing power available. Given that inference takes place on the server side of the web application, its performance relies heavily on the server's computational capacity. Therefore, it is advisable to consider

configuring the web application to conduct inference on the client side for optimal results.

2. **File compression:** Compressing the video file before uploading it to the web application offers numerous advantages beyond optimized memory usage and bandwidth efficiency. Firstly, it reduces the time required for uploading and inferencing. Additionally, compressed files consume less storage space on servers, which can result in cost savings for hosting services. Moreover, smaller file sizes are generally easier to manage and distribute, facilitating smoother operations and reducing the likelihood of technical issues. Overall, implementing compression techniques for video files significantly improves web application performance and functionality.
3. **Additional traffic data information:** Including extraneous details can enhance the quality of generated traffic data. For instance, incorporating speed data can assist researchers in analyzing the average speeds at which vehicles traverse specific roads. Furthermore, integrating optical character recognition (OCR) tools into the system to capture plate numbers could prove beneficial for various purposes.
4. **Creative data visualization:** Integrating creative and configurable data visualization into the web application is recommended to augment its functionality. For example, the incorporation of interactive charts and graphs enables users to visually interpret intricate data sets more effectively, thereby facilitating enhanced analysis and informed decision-making. Furthermore, offering configurable visualization options enables users to change their view according to their specific requirements and preferences

REFERENCES:

Ancheta, M. R. G. (2015). Halakhak: Defining the "National" in the Humor of Philippine Popular Culture. Thammasat Review, 14(1), 35–60. Retrieved from https://sc01.tci-thaijo.org/index.php/tureview/article/view/40801

CNN Philippines Staff. (2018, February 23). *PH loses ₱3.5B a day due to Metro Manila traffic – JICA*. CNN Philippines. Retrieved October 29, 2023, from https://www.cnnphilippines.com/transportation/2018/02/23/JICA-P3.5-billion-traffic.html

Laurel, D. (2022, March 8). Think you can make a living manually counting cars on EDSA all day? *https://www.topgear.com.ph*. https://www.topgear.com.ph/news/motoring-news/mmda-edsa-car-count-a962-20220308

Metro Manila Development Authority. (2022, August 15). RE-IMPLEMENTING THE UNIFIED VEHICLE VOLUME REDUCTION PROGRAM IN THE NATIONAL CAPITAL REGION STARTING ON 15 AUGUST 2022. https://mmda.gov.ph/images/pdf/resolution/2022/22-14_s_2022_Re-implementing_the_UVVRP_in_the_National_Capital_Region_.pdf

Metro Manila Development Authority. (2022, October 19). REIMPLEMENTING THE TRUCK BAN ALONG ROXAS BOULEVARD MANILA

Nor Azlan, N. N., & Md Rohani, M. (2018). Overview Of Application Of Traffic Simulation Model. *MATEC Web of Conferences*, *150*, 03006. https://doi.org/10.1051/matecconf/201815003006

Redmon, J., Divvala, S., Girschick, R., Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 779-788). IEEE. https://doi.org/10.1109/CVPR.2016.91

Tai, M. C. (2020). The impact of artificial intelligence on human society and bioethics. *Tzu Chi Medical Journal*, *32*(4), 339. https://doi.org/10.4103/tcmj.tcmj_71_20

TomTom Traffic Index. (2023). *Traffic Index Ranking 2022*. TomTom. Retrieved October 29, 2023, from https://www.tomtom.com/traffic-index/ranking/

Congestion. (2005). In R. Caves (Ed.), *Encyclopedida of the City* (1st ed., Vol. 1). Taylor & Francis.

Choudhary, M. (2019, January 14). *What is Intelligent Transport System and how it works?* Geospatial World. https://www.geospatialworld.net/blogs/what-is-intelligent-transport-system-and-how-it-works/

Lieu, H. (1999, January). Traffic-Flow Theory. *Public Roads, 62(4)*

Elefteriadou, L. (2014). *An Introduction to Traffic Flow Theory*. Springer New York, NY.

Tsuboi, T. (2021). Traffic Flow Analysis and Management. IntechOpen. doi: 10.5772/intechopen.95087

Opensource.com. (n.d.). *What is Python?* https://opensource.com/resources/python

Wu, C. (2021). *Traffic Flow Theory: Macroscopic perspective on traffic flow fundamentals* [Slide show]. MIT - Massachusetts Institute of Technology. https://web.mit.edu/1.041/www/fall2021/www/lectures/L5-traffic-flow-theory-2021fa-pre.pdf

Shaheen, S., & Finson, R. S. (2004). Intelligent transportation systems. In *Elsevier eBooks* (pp. 487–496). https://doi.org/10.1016/b0-12-176480-x/00191-1

Shanmukha. (2016, December 13). Traffic Flow, Speed, Density. *Civil Engineering*. Retrieved November 27, 2023, from https://knowledge4civil.wordpress.com/2016/12/13/traffic-flow-speed-density/

Gillis, A. S., & Botelho, B. (2023, March 7). *MongoDB*. Data Management. https://www.techtarget.com/searchdatamanagement/definition/MongoDB

Megida, D. (2021, April 28). *What is JavaScript? A Definition of the JS Programming Language*. freeCodeCamp.org. https://www.freecodecamp.org/news/what-is-javascript-definition-of-js/

Li, Yue & Zhao, Qunshan & Wang, M.. (2022). ANALYSIS THE INFLUENCING FACTORS OF URBAN TRAFFIC FLOWS BY USING NEW AND EMERGING URBAN BIG DATA AND DEEP LEARNING. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. XLIII-B4-2022. 537-543. 10.5194/isprs-archives-XLIII-B4-2022-537-2022.

N. Buch, S. A. Velastin and J. Orwell, "A Review of Computer Vision Techniques for the Analysis of Urban Traffic," in IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 3, pp. 920-939, Sept. 2011, doi: 10.1109/TITS.2011.2119372.

Yurshevich, E. and Yatskiv, I. (2012). Consideration of the Aspects of the Transportation Systems Microscopic Model Application as Part of a Decision Support System. Riga: Transport and Telecommunication, 13(3), 209-218.

Batenko, A., Grakovski, A., Kabashkin, I., Petersons, E., and Sikerzhicki, Y. (2011).

Weight-in-Motion (WIM) Measurements by Fiber-Optic Sensor: Problems and Solutions. Riga: Transport and Telecommunication, 12(4), 27-33.

Zheng, PJ & Mike, McDonad. (2012). An Investigation on the Manual Traffic Count Accuracy. Procedia - Social and Behavioral Sciences. 43. 226–231. 10.1016/j.sbspro.2012.04.095.

McGowen and Sanderson. (2011). Accuracy of Pneumatic Road Tube Counters. A report prepared for the 2011 Western District Annual Meeting Institute of Transportation Engineers Anchorage. http://www.westernite.org/annualmeetings/alaska11/Compendium/Moderated%20Session%20Papers/8C-Patrick%20McGowen.pdf

U.S. Department of Transportation. (2013). Traffic Monitoring Guide. http://www.fhwa.dot.gov/policyinformation/tmguide/tmg_fhwa_pl_13_015.pdf

Caruso, M.J., & Withanawasam, L. (1999). VEHICLE DETECTION AND COMPASS APPLICATIONS USING AMR MAGNETIC SENSORS. https://aerospace.honeywell.com/content/dam/aerobt/en/documents/learn/products/sensors/technical-articles/VehicleDetectionandCompassApplicationsusingAMRMagneticSensors_ta.pdf

Szwoch, G., & Kotus, J. (2021). Acoustic Detector of Road Vehicles Based on Sound Intensity. Sensors (Basel, Switzerland), 21(23), 7781. https://doi.org/10.3390/s21237781

Klein L.A., Mills M.K., Gibson D.R.P. Traffic Detector Handbook. 3rd ed. Federal Highway Administration Research and Technology; Washington, DC, USA: 2006. Report No. FHWA-HRT-06-108.

Roser, M. (2023, October 29). The brief history of artificial intelligence: The world has changed fast – what might be next? Our World in Data. https://ourworldindata.org/brief-history-of-ai

Castro, J., Hyodo, T., & Kuse, H. (2003). A STUDY ON THE IMPACT AND EFFECTIVENESS OF THE TRUCK BAN SCHEME IN METRO MANILA. Journal of the Eastern Asia Society for Transportation Studies, 5, 2177. https://www2.kaiyodai.ac.jp/~hyodo/047.pdf

Oregon State University, Portland State University, & University of Idaho. (2001). *Traffic Flow Parameters*. Transportation Engineering Online Lab Manula. Retrieved November 27, 2023, from https://www.webpages.uidaho.edu/niatt_labmanual/chapters/trafficflowtheory/theoryandconcepts/TrafficFlowParameters.htm

History of Artificial intelligence - Artificial intelligence - www.coe.int. (n.d.). Artificial Intelligence. https://coe.int/en/web/artificial-intelligence/history-of-ai

Marr, B. (2023, May 10). 15 Amazing Real-World Applications Of AI Everyone Should Know About. Forbes. https://www.forbes.com/sites/bernardmarr/2023/05/10/15-amazing-real-world-applications-of-ai-everyone-should-know-about/?sh=3e452fea85e8

The present and future of AI. (2021, October 19). https://seas.harvard.edu/news/2021/10/present-and-future-ai

The Editors of Encyclopedia Britannica. (n.d.-b). What is the impact of artificial intelligence (AI) on society? Encyclopedia Britannica. https://www.britannica.com/question/What-is-the-impact-of-artificial-intelligence-AI-on-society

Sawtell-Rickson, J. (2022, December 21). What is Computer Vision? Built In. https://builtin.com/machine-learning/computer-vision

The Editors of Encyclopedia Britannica. (2023, October 30). Computer vision | Definition, Examples, Applications, & Facts. Encyclopedia Britannica. https://www.britannica.com/technology/computer-vision

Srivastava, A. (2021, October 14). The Evolution Of Computer Vision And Its Impact On Real-World Applications. Forbes. https://www.forbes.com/sites/forbestechcouncil/2021/10/14/the-evolution-of-computer-vision-and-its-impact-on-real-world-applications/?sh=27c9ffefc6ab

Szeliski, R. (2022). Computer Vision. In Texts in computer science. https://doi.org/10.1007/978-3-030-34372-9

Maione, I. (2023b, May 2). Computer Vision: definition, applications, tools, history & future [Video]. clickworker.com. https://www.clickworker.com/ai-glossary/computer-vision/

Copeland, B. (2023, November 3). Artificial intelligence (AI) | Definition, Examples, Types, Applications, Companies, & Facts. Encyclopedia Britannica. https://www.britannica.com/technology/artificial-intelligence

Hassan, H., Ren, Z., Zhao, H., Huang, S., Li, D., Xiang, S., Kang, Y., Chen, S., & Huang, B. (2022). Review and classification of AI-enabled COVID-19 CT imaging models based on computer vision tasks. Computers in Biology and Medicine, 141, 105123. https://doi.org/10.1016/j.compbiomed.2021.105123

Ipopba. (n.d.). AI, Machine learning, Hands of robot and human touching on big data. . . Getty Images. https://www.gettyimages.com/detail/photo/machine-learning-hands-of-robot-and-human-touching-royalty-free-image/1206796363

Reddy, M. (2011) API design for C++. Burlington: Elsevier Science. https://books.google.com.ph/books?id=IY29LylT85wC&lpg=PP1&pg=PP1#v=onepage&q&f=false.

What is Flask Python - Python Tutorial. (n.d.). https://pythonbasics.org/what-is-flask-python/

Scott, G. (n.d.-b) Application programming interface (API): Definition and examples, Investopedia. https://www.investopedia.com/terms/a/application-programming-interface.asp

Web API basics – definition, technologies, features, and uses. (n.d.-b) Gravitee. https://www.gravitee.io/understanding-web-apis-and-their-use

Bryan, A. and Yusuf, B. (n.d.-b) 'What is an API endpoint?', https://www.contentful.com/blog/api-endpoint/#:~:text=In%20summary%2C%20an%20API%20endpoint,and%20instructions%20via%20the%20endpoint.

Cambridge Systematics, Inc. (2005). Traffic Congestion and Reliability: Trends and Advanced Strategies for Congestion Mitigation. In *Federal Highway Administration*. Retrieved November 26, 2023, from https://ops.fhwa.dot.gov/congestion_report/chapter2.htm

Data Formats (n.d.-b) NYU Developer Support. Available at: https://wp.nyu.edu/developers/data-formats/#:~:text=Data%20provided%20by%20APIs%20can,RSS%2C%20CSV%20Files%2C%20etc.

'*Difference between authentication and authorization?*' (2023) KongHQ, 9 June. https://konghq.com/blog/engineering/api-authentication-vs-api-authorization.

Ellis, M. (2023) '*API Rate Limiting - Everything You Need to Know*', HubSpot, 22 August. https://blog.hubspot.com/website/api-rate-limit.

Fitzgerald, A. (2023) '*What is API documentation'*, HubSpot, 26 September. https://blog.hubspot.com/website/api-documentation#:~:text=API%20documentation%20is%20essentially%20an,to%20work%20with%20the%20API.

'*SOAP vs. REST: A Look at Two Different API Styles*' (2021) UpWork, 12 August. https://www.upwork.com/resources/soap-vs-rest-a-look-at-two-different-api-styles.

Redmon, J., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

Ultralytics. (2023). YOLOv8: The ultimate object detection algorithm. Retrieved from [Ultralytics website].

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

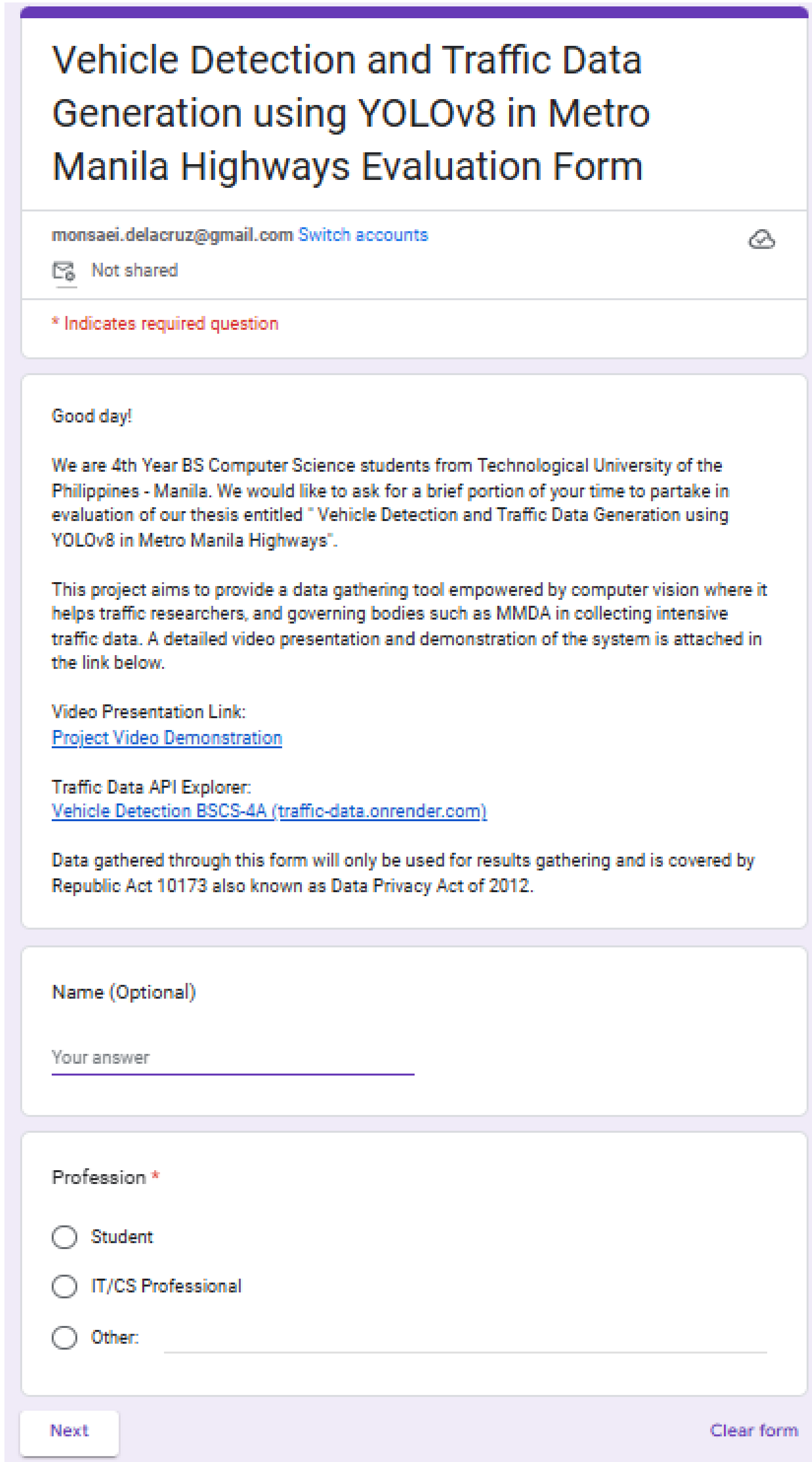
Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., … & Tang, X. (2017). Residual attention network for image classification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3156-3164).]

What is Traffic Congestion? Traffic Congestion Meaning. (2023, September 23). Isarsoft. Retrieved November 26, 2023, from https://www.isarsoft.com/knowledge-hub/traffic-congestion

Vite. (n.d.). *Why Vite*. Vite.dev. Retrieved November 28, 2023, from https://vitejs.dev
ISO 25010. (n.d.). ISO 25000 Software and Data Quality. Retrieved December 1, 2023, from https://iso25000.com/index.php/en/iso-25000-standards/iso-25010

APPENDIX A

Survey Questionnaire Form



Vehicle Detection and Traffic Data Generation using YOLOv8 in Metro Manila Highways Evaluation Form

monsaei.delacruz@gmail.com Switch accounts

Not shared

* Indicates required question

Good day!

We are 4th Year BS Computer Science students from Technological University of the Philippines - Manila. We would like to ask for a brief portion of your time to partake in evaluation of our thesis entitled " Vehicle Detection and Traffic Data Generation using YOLOv8 in Metro Manila Highways".

This project aims to provide a data gathering tool empowered by computer vision where it helps traffic researchers, and governing bodies such as MMDA in collecting intensive traffic data. A detailed video presentation and demonstration of the system is attached in the link below.

Video Presentation Link:
Project Video Demonstration

Traffic Data API Explorer:
Vehicle Detection BSCS-4A (traffic-data.onrender.com)

Data gathered through this form will only be used for results gathering and is covered by Republic Act 10173 also known as Data Privacy Act of 2012.

Name (Optional)

Your answer

Profession *

- Student
- IT/CS Professional
- Other:

Next

Clear form

Functional Suitability

This characteristic represents the degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions.

Functional Completeness *

The software functionalities cover all specified tasks and user objectives

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Functional Correctness *

The software provides results that are accurate based on the industry standard applications and mathematics

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Functional Appropriateness *

The software functionalities can be used for accomplishment of specified tasks and objectives.

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Performance Efficiency

This characteristic represents the degree to which a product performs its functions within specified time and throughput parameters and is efficient in the use of resources (such as CPU, memory, storage, network devices, energy, materials...) under specified conditions.

Time Behavior *

Time taken for the software to process data and give responses to its users.

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Resource Utilization *

Software utilization and management of resources when performing its functions.

	1	2	3	4	
Not Acceptable	◯	◯	◯	◯	Highly Acceptable

Capacity *

Maximum limitations of the software are enough to performs its functions.

	1	2	3	4	
Not Acceptable	◯	◯	◯	◯	Highly Acceptable

Usability

Degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use.

Learnability *

Degree to which the user can learn to operate the software with efficiency and effectiveness.

	1	2	3	4	
Not Acceptable	◯	◯	◯	◯	Highly Acceptable

Operability *

Software design that affects the easiness of operating the system.

	1	2	3	4	
Not Acceptable	◯	◯	◯	◯	Highly Acceptable

Reliability

Degree to which a system, product or component performs specified functions under specified conditions for a specific period of time.

Availability *

Software design that affects the easiness of operating the system.

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Fault Tolerance *

The system operates as intended despite the presence of hardware or software faults.

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Maintainability

This characteristic represents the degree of effectiveness and efficiency with which a product or system can be modified to improve it, correct it, or adapt it to changes in environment and in requirements.

Reusability *

The software can be used to generate results that can be reused for other projects.

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Modifiability *

The software can be modified efficiently without introducing defects or degrading existing product quality.

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Testability *

Software's effectiveness and efficiency can easily be tested through the different test criteria.

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Thank you for answering our evaluation form. All your insights and feedback are very valuable to us. We deeply appreciate your effort and time in filling out our evaluation form and providing your thoughts about it. If you are interested in contacting us, feel free to message us with the emails provided below:

joshuelernest.simbulan@tup.edu.ph
imee.compra@tup.edu.ph
simondaniel.delacruz@tup.edu.ph
ronan.esguerra@tup.edu.ph
andrewjames.tejerero@tup.edu.ph

Once again, we are thankful for your contribution to our academic journey. Your support means a lot to us, and we are deeply thankful for your involvement. God Bless!

Insights and Feedback:

Your answer

Back Submit Clear form

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-5301-3001 local 608| Fax No. +632-8521-4063 Email: cos@tup.edu.ph | Website: www.tup.edu.ph	Index No.	REF-COS-3.5-INT-TGC
		Revision No.	00
		Effectivity Date	06132022
VAA-COS	THESIS GRAMMARIAN CERTIFICATION	Page	1 / 1

THESIS GRAMMARIAN CERTIFICATION

This is to certify that the thesis entitled,

VEHICLE DETECTION AND TRAFFIC DATA GENERATION USING YOLOV8 IN METRO MANILA HIGHWAYS

authored by

Imee Q. Compra
Simon Daniel M. Dela Cruz
Ronan M. Esguerra
Joshuel Ernest Q. Simbulan
Andrew James S. Tejerero

has undergone editing and proofreading by the undersigned.

This Certification is being issued upon the request for whatever purposes it may serve them.

JENNIFER P. ALINSUNOD, CHRA, LPT
Grammarian

Technological University of the Philippines

June 13, 2024

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-301-3001 local 711 Email: urds@tup.edu.ph | Website: www.tup.edu.ph	Index No.	F-URD-4.1-CSI
		Issue No.	01
		Revision No.	00
		Date	07102023
VRE-URD	**CERTIFICATE OF SIMILARITY INDEX USING TURNITIN**	Page	1 / 1
		QAC No.	CC-07102023

This is to certify that the manuscript entitled

"Vehicle Detection and Traffic Data Generation Using YOLOv8 in Metro Manila Highways"

Authored by

Imee Q. Compra
Simon Daniel M. Dela Cruz
Ronan M. Esguerra
Joshuel Ernest Q. Simbulan
Andrew James S. Tejerero

College of Science

has been subjected to similarity check on June 13, 2024
using Turnitin with generated similarity index of **11%**.

Processed by:

Assoc. Prof. FRANCIS A. ALFARO, Ed.D., LPT
Faculty, University Research
and Development Services

Certified correct by:

Assoc. Prof. FRANCISCO D. ESPONILLA II, Ed.D.
Director, University Research
and Development Services

Transaction ID	URD-TCERT-ELGB-10032022
Signature	

Vehicle Detection and Traffic Data Generation using YOLOv8 in Metro Manila Highways

andrew tejerero TEJERERO_VEHICLE DETECTION AND T... Switch to the new view Submission Details Help

Top sources | All Sources

0 Flags

11% Overall Similarity

11% Overall Similarity

Document Details

-- AI

#	Source	Type	Similarity
1	tup on 2024-06-13	SUBMITTED WORKS	<1%
2	tup on 2022-06-25	SUBMITTED WORKS	<1%
3	www.v7labs.com	INTERNET	<1%
4	www.coursehero.com	INTERNET	<1%
5	edoc.pub	INTERNET	<1%
6	www.retailsensing.com	INTERNET	<1%
7	Coventry University on 2022-03-04	SUBMITTED WORKS	<1%
8	tup on 2024-06-11	SUBMITTED WORKS	<1%
9	repository.sustech.edu	INTERNET	<1%
10	University of East London on 202...	SUBMITTED WORKS	<1%
11	www.geospatialworld.net	INTERNET	<1%
12	apps.dtic.mil	INTERNET	<1%
13	www.portcalls.com	INTERNET	<1%
14	Fontaine, Michael. "Traffic Monit...	CROSSREF	<1%
15	medium.com	INTERNET	<1%
16	King's College on 2024-04-11	SUBMITTED WORKS	<1%
17	National School of Business Man...	SUBMITTED WORKS	<1%
18	Quezon City University on 2024-...	SUBMITTED WORKS	<1%
19	University of Birmingham on 20...	SUBMITTED WORKS	<1%
20	iso25000.com	INTERNET	<1%

Vehicle Detection and Traffic Data Generation

Using YOLOv8 in Metro Manila Highways

Imee Q. Compra
Simon Daniel M. Dela Cruz
Ronan M. Esguerra
Joshual Ernest Q. Simbulan
Andrew James S. Tejerero

A Thesis Presented to the
Faculty of the College of Science
Technological University of the Philippines
Ayala Blvd., Manila

In Partial Fulfillment of the
Requirements for the Degree
Bachelor of Science in Computer Science

June 2024

Vehicle Detection and Traffic Data Generation using YOLOv8 in Metro Manila Highways

ABSTRACT

The worsening traffic conditions in urban cities reflect neglect in proper preparati and a lack of studies about traffic management. This problem can be pointed out as one the effects of having insufficient comprehensive data, specifically in Metro Manila. T primary objective of this study is to develop a project that can replace the traditional wa of collecting and generating traffic data with a more robust, automated, and scalable w application. YOLOv8 – the main algorithm for vehicle detection – was implemented on diverse dataset of traffic images in highway environments. The model's accuracy, spe and robustness were assessed through precision, recall, and score metrics. Data generati techniques are also employed to export traffic count values from either live or record video inputs. The results demonstrate that the YOLOv8 achieved a high detection accura with a mean average precision (mAP) of 86.76% and a real-time processing speed of 2 30 frames per second (FPS), and successfully generating compact traffic data that can used for data-driven scientific studies. That concludes that this project is a potent stepping stone towards understanding the underlying crisis involving road a transportation systems as it takes advantage of modern technologies to resolve and previ

RESEARCHERS' PROFILE

Name: Imee Compra

Age: 22

Date of birth: August 18, 2001

Address: 3540 Sta. Monica St. Don Galo, Parañaque City

Parents: Isidro K. Compra, Myrna Q. Compra

EDUCATIONAL BACKGROUND

Tertiary	**Technological University of the Philippines – Manila** 2020 - Present
Secondary Education	**Olivarez College, Parañaque (Senior High School – STEM Track)** 2018 - 2020 **Parañaque Science High School** 2014 – 2018
Elementary Education	**Fourth Estate Elementary School** 2008 - 2014

Name: Simon Daniel M. Dela Cruz

Age: 22

Date of birth: April 7, 2002

Address: Ipo Road, Minuyan Proper, City of San Jose del Monte, Bulacan

Parents: Cromwel C. Dela Cruz, Sylvia M. Dela Cruz

EDUCATIONAL ATTAINMENT

Tertiary	**Technological University of the Philippines – Manila** 2020 - Present
Secondary Education	**Sapang Palay National High School (Senior High School – STEM Track)** 2018 - 2020
	Sapang Palay National High School 2016 – 2018
	City of San Jose del Monte Science High School 2014 - 2016
Elementary Education	**Minuyan Proper Elementary School** 2008 - 2014

Name: Ronan M. Esguerra

Age: 22

Date of birth: December 17, 2001

Address: Alessandro St., Vita Toscana, Brgy. Mambog IV, Bacoor City, Cavite

Parents: RonRon D. Esguerra, Ma. Dinah M. Esguerra

EDUCATIONAL ATTAINMENT

Tertiary	**Technological University of the Philippines – Manila** 2020 - Present
Secondary Education	**St. Dominic College of Asia (Senior High School – STEM Track)** 2018 - 2020 **St. Dominic College of Asia** 2014 – 2018
Elementary Education	**Little Jesus Learning Center** 2008 - 2014

Name: Joshuel Ernest Q. Simbulan

Age: 22

Date of birth: June 8, 2002

Address: Blk. 4 Lot 80 Purok 3 Brgy. San Pedro City of San Jose del Monte Bulacan

Parents: Ernesto E. Simbulan, Marilen Q. Simbulan

EDUCATIONAL ATTAINMENT

Tertiary	**Technological University of the Philippines – Manila** 2020 - Present
Secondary Education	**Sapang Palay National High School (Senior High School – STEM Track)** 2018 - 2020 **Sapang Palay National High School** 2014 – 2018
Elementary Education	**Bagong Buhay F Elementary School** 2008 - 2014

Name: Andrew James S. Tejerero

Age: 22

Date of birth: September 25, 2001

Address: Blk 12, Lot 10, Ph4-A, Towerville, Brgy. Minuyan Proper, City of San Jose del Monte, Bulacan

Parents: Justino M. Tejerero, Lydia S. Tejerero

EDUCATIONAL ATTAINMENT

Tertiary	**Technological University of the Philippines – Manila** 2020 - Present
Secondary Education	**Sapang Palay National High School (Senior High School – STEM Track)** 2018 - 2020 **Sapang Palay National High School** 2014 – 2018
Elementary Education	**Towerville Elementary School** 2008 - 2014