HA.ZEE: A Vehicular $PM_{2.5}$ Estimation Application Using Traffic Footage

A Special Problem
Presented to
the Faculty of the Division of Physical Sciences and Mathematics
College of Arts and Sciences
University of the Philippines Visayas
Miag-ao, Iloilo

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

CABATU-AN, John Gabriel
CUSTODIO, Adrian Miguel

Francis DIMZON
Adviser

June 8, 2023

Approval Sheet

The Division of Physical Sciences and Mathematics, College of Arts and Sciences, University of the Philippines Visayas

certifies that this is the approved version of the following special problem:

HA.ZEE: A Vehicular $PM_{2.5}$ Estimation Application Using Traffic Footage

Approved by:

Name	**Signature**	**Date**
Prof. Francis D. Dimzon (Adviser)	__________	__________
Dr. Arnel L. Tampos (Division Chair)	__________	__________

Division of Physical Sciences and Mathematics

College of Arts and Sciences

University of the Philippines Visayas

Declaration

We, JOHN GABRIEL CABATU-AN and ADRIAN MIGUEL CUSTODIO hereby certify that this Special Problem, including the pdf file, has been written by us and is the record of work carried out by us. Any significant borrowings have been properly acknowledged and referred.

Name	**Signature**	**Date**
Cabatu-an, John Gabriel (Student)	__________	__________
Custodio, Adrian Miguel (Student)	__________	__________

Dedication

Team Ha.Zee dedicates this Special Problem to the instructors, the teachers, and the professors who have helped us in our academic journey. They have provided us with the proper knowledge and skills to take on this project. We also dedicate this paper to the friends and families who have supported us throughout the project. Lastly, we dedicate this to the future researchers of the topic who plan to use this study. Whether it be the computer scientists who specialize in computer vision and machine learning; data scientists who share the goal of increasing the availability of image datasets of our local vehicles online; or environmental scientists who believe technology can be a great aid in helping solve environmental issues; we hope this paper will be useful in your endeavors.

Acknowledgment

Throughout the course of this study, we have had countless help from people that deserve to be acknowledged.

To Sir Francis D. Dimzon, we would like to extend our deepest gratitude to him for taking us under his wing and becoming our advisor in the study. This entire process has been a journey and we appreciate his guidance throughout it all.

To Ma'am Ara Abigail Ambita, thank you for all the lessons and skills she has taught us during our Machine Learning classes. Her guidance has allowed us to utilize those skills in this study. For that, we are truly grateful.

To the owners, commuters, and drivers of the different vehicles used in the training dataset of the study, we extend our deepest thanks to them all for this study would have not been possible without the vehicles. We owe this study to them and may it contribute to combating the air pollution in the country.

Lastly, we thank the GitHub, YOLOv5, and Roboflow community. Their bug fixes, documentation, and suggestions have been of great help in finishing this study. We extend our thanks to them and their willingness to share knowledge in pursuit of advancing our technologies.

"I would be remiss if I did not mention all the people that inspired us going forward in this journey and kept me sane:

To my parents who gave me the financial support to afford the equipment and sustenance I needed for this study, I am indebted to you, literally and figuratively.

To my best friends, even though we only see each other rarely, you never fail to check up on me. I appreciate all the support even after all these years.

To Kompol, Komsai, and my UPV friends, thank you for making this challenge a little bit easy to handle. I could not have done it without all the moral support you have given me.

To my groupmate, Yanni, my deepest gratitude for the help in this project. This project literally would not exist without your help. I am honored to have you as my groupmate.

To all the people that supported us in this endeavor, thank you very much.

Finally, to the stars from me, with love. "

- John Gabriel Cabatu-an

"I would like to thank the multitude of people who have supported me throughout my journey in UPV. They have seen me change throughout the years and have stuck with me through my highest highs and lowest lows.

To my family: Mama, Papa, and Yelli. You have been there for me in every step of my life. You have given me stability, whether it be emotionally or financially, and have seen my struggle to finish this degree. Thank you and I hope I made you all proud.

To Bs and Ns, I am so lucky I have found you all at an early point in my life. Different colleges did not stop us from constantly hyping each other up. Thank you for keeping me sane during the pandemic and I'm overjoyed to see everything you all achieved. I can't wait to meet you all again!

To the friends I've made in my stay in UPV, this academic journey was not an easy on and I certainly wouldn't have made it without you all by my side. The Dads, KomPol, and Beecardo, thank you for picking me up and making my stay here worthwhile.

To Komsai 2019, I have made it a goal to know each and every one of you at during our first year, and maybe make friends along the way. I did not expect to belong in such a batch of multitalented individuals, whom many I can truly consider my friends and confidants. I look forward to all your future endeavors and may we meet again when we've achieve our goals. Padayon!

To JG, my thesis groupmate, we've been groupmates in nearly every possible group project since our first semester and I am eternally grateful for that. When we found out we'd do this SP as a duo, we knew it would be challenging but we still stuck through with it. Thank you so much for everything, Gab! You deserve every fortune that comes your way. Padayon!"

- Adrian Miguel Custodio

Abstract

Air pollution is a global problem and the Philippines ranked third among countries with deaths relating to air pollution. Mobile sources are responsible for 65% of the pollutants in the atmosphere and for two decades, the country has tried to mitigate these atmospheric issues but shows no improvement. Air quality monitoring is important for mitigating air pollution in the Philippines. However,keeping these air quality monitoring devices operational needs high maintenance. Moreover, it is expensive to maintain these tools, and access to the data is limited. This project aimed to utilize new technologies to develop an alternative air quality monitoring system to help bring a solution to this problem. Ha.Zee mainly focused on the fine particulate matter ($PM_{2.5}$) emitted from traffic vehicles in the Philippines. This project utilized an object detection algorithm, YOLOv5 to be trained to identify and count the number of vehicles on the road to estimate the amount of ($PM_{2.5}$) emitted by vehicles. YOLOv5 proved to be a viable tool in detecting traffic vehicles for recording emissions and was fairly accurate in detecting the relevant objects in a scene while achieving F1-scores ranging from 0.68 to 0.91

Keywords: Machine Learning, Computer Vision, Object Detection, YOLOv5, traffic, vehicle-related emissions

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview of the Current State of Technology

The Department of Environment and Natural Resources (DENR) expressed on its website that monitoring air quality is essential in reducing air pollution and they plan to protect the environment and public health by strengthening their air quality monitoring systems (DENR, 2020).

An example of an air quality monitoring system that the DENR uses is the Differential Optical Absorption Spectroscopy (DOAS)(DENR, n.d.). DOAS captures light that passes through the atmosphere, to measure different wavelengths that were absorbed by different gasses. This method can accurately measure trace gasses absorption and it is simpler and less expensive to operate. DOAS, however, is greatly affected by turbulence in the atmosphere (Platt & Stutz, 2008). DENR also has particulate matter stations that record $PM_{2.5}$ and PM_{10} in the atmosphere (DENR, n.d.).

DOAS equipment needs frequent maintenance to be able to operate normally. In a news report by Enano & Subingsubing (2019) regarding air pollution at EDSA it was stated that the maintenance of this equipment requires “hundreds of thousands of pesos”.

Currently, the way to access the data from the AQMS stations is through the website (https://air.emb.gov.ph/ambient-air-quality-monitoring/) and the Google Play Store application. Figure 1.1 shows the contents of the application.

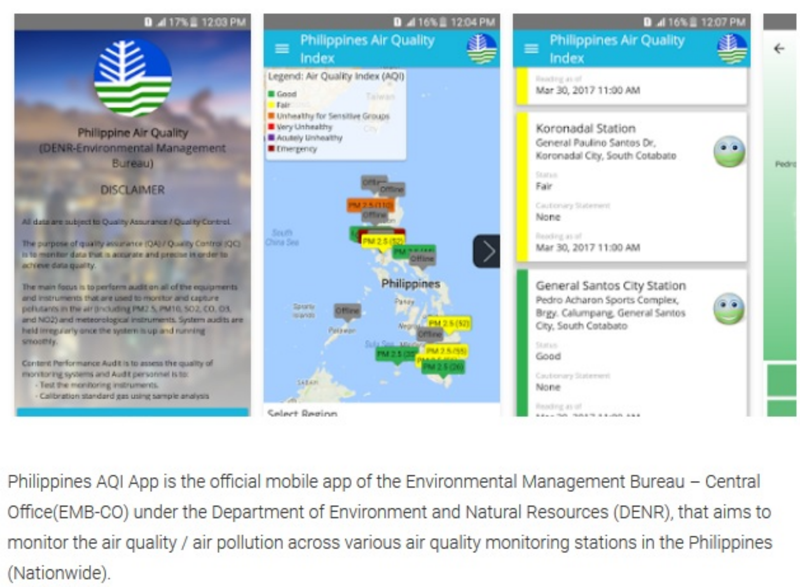


Figure 1.1: Screen capture of the mobile application of the Philippines Air Quality Index.

In the mobile application disclaimer, it is stated that the system audits are irregular and all data are subject to quality assurance and quality control. This means that the end user may not get the accurate data that they expect when using the application. Moreover, monitoring stations are not online 24/7 which makes the data less accessible.

1.2 Problem Statement

Air pollution has become a global problem over the years. As stated by Akimoto (2004), the availability of the CO_2 concentrations on the Measurement of Air Pollution from Satellite (MAPS) instrument in 1981 shows high concentrations of the greenhouse gas over tropical Asia, Africa, and South America. Not only does this data provide evidence that this has become an international issue, but it also shows how fossil fuel combustion can have an impact on air quality.

The Philippines, a country located in tropical Asia, is not devoid of these issues. An article by Abano (2019) states that a 2018 study by the World Health Organization reports the Philippines has ranked third among the countries with air pollution-related deaths. These deaths are tied to harmful particles entering a person's lungs, which can lead to multiple different ailments and diseases: heart disease, lung cancer, and respiratory infections, to name a few.

Air pollution can come from different sources, whether it be from stationary constructs like factories or mobile sources such as cars (Environmental Management Bureau, 2015). An air quality status report by the Department of Environment and Natural Resources (2015) shows that 65% of air pollutants come from these mobile sources. This worsened as the EMB's official site, Environmental Management Bureau (2018), states that based on the Emissions Inventory of 2018, the pollutant contribution of mobile sources has increased to 74%. In places where traffic is congested could be a huge contributing factor to vehicular emissions. Vergel and Yai (2000) state that the congestion in the roads of Metro Manila contributes to the worsening air quality, especially in the vicinity of the road environment.

A key component of these emissions is particulate matter (PM): $PM_{2.5}$ and PM_{10}. PM is a measure of solid or liquid particles in the air that are inhalable, this includes dust, smoke, and dirt (United Sates Environmental Protection Agency, 2022). Accordingly, PM_{10} is the measure of inhalable particulate matter in the air that is at most 10 micrometers in diameter. Similarly, $PM_{2.5}$ is the measure of inhalable particulate matter in the air that is at most 2.5 micrometers in diameter. US EPA (2022) additionally states that microscopic particulate matter particles can be inhaled and cause them to get to the lungs or the bloodstream which can lead to serious health problems, with $PM_{2.5}$ having the greatest risk.

In the country's attempt to mitigate the atmospheric issues, the Philippine Clean Air Act of 1999 (Republic Act No. 8749) was passed (Food and Agriculture Organization of the United Nations, n.d.). It entails the resolution of creating a national program of air pollution management, mainly focusing on pollution prevention. Two decades later and the country still sees increasing pollutants in the air and does not show signs of the improvement that was planned.

In addition to creating air pollution management programs, the Philippine Clean Air Act also aimed to utilize mass media communication to create awareness and active participation in air quality planning and monitoring. Considering that the available technology dedicated to monitoring the air quality in the country is sparsely spread throughout the country, being accessible to more citizens can help in creating awareness at a lower cost. With the aforementioned said, a system that could satisfy that goal can be created with newer technologies that were not present in the decades prior. Thus, Ha.Zee was constructed, a system that can gather $PM_{2.5}$ information from vehicles – one of the biggest contributors to air pollution and provide an estimate of emissions produced, as a solution to these problems.

1.3 Research Objectives

1.3.1 General Objective

The general objective of the study was to develop an application that calculates the location's average amount of fine particulate matter ($PM_{2.5}$) emission from vehicles through the use of a vehicle detection system. This system identified the vehicles on the street from a video recording. The system integrated the values of $PM_{2.5}$ of different vehicles, which was utilized to assign values to on-screen vehicles for their total average to be calculated. The average $PM_{2.5}$ value was then be displayed on the screen for the user to see.

1.3.2 Specific Objectives

This study specifically aims to:

1. Study object detection algorithms and find the appropriate model to use for the study

2. Gather a collection of images of vehicles to be used and train a system that detects vehicles in traffic footage.
3. Calculate the estimated $PM_{2.5}$ emission values based on the detected vehicles in the traffic footage
4. Test the trained model on the video traffic footage

1.4 Scope and Limitations of the Research

This application mainly focused on the $PM_{2.5}$ emitted by traffic in the Philippines, where the researchers reside as of writing the paper. Thus, it was only be set up and used on vehicles that travel within the country. For this reason, this project had some significant difficulties in using existing databases of vehicles that are mainly found locally (i.e. jeepneys and tricycles), with little to no readily available databases existing to be utilized in the study. To avoid said vehicles from not being detected, the researchers opted to take pictures and videos of the vehicles in traffic to be used as training data. These vehicles were recorded on a smartphone and were uploaded in Roboflow, a computer vision developer framework for preprocessing and model training techniques (Bhattacharyya, 2020). This was used to annotate them as their respective vehicle type.

$PM_{2.5}$ can come from many different sources. Air Quality Ontario (n.d.) states that the major sources of $PM_{2.5}$ are "motor vehicles, smelters, power plants, industrial facilities, residential fireplaces and wood stoves, agricultural burning, and forest fires." Since this project focuses on detecting vehicles, other objects were not accounted for when calculating the total $PM_{2.5}$ in the vicinity. Thus, this application was limited to finding an estimate of the average $PM_{2.5}$ emissions produced only by vehicles.

Furthermore, this study utilized the YOLOv5 object identification framework and was thus limited to the features of that version. Any other features and upgrades that are present in future versions of the framework were not be included in the study.

1.5 Significance of the Research

The main objective of this study was to create an application that helps its users identify the $PM_{2.5}$ levels of a traffic congested area through a video over the road. It served as an example of how computer vision can be utilized to get an estimate of the pollutants in an area via identifying the vehicles they come from and the amount of $PM_{2.5}$ they produce. This poses benefits to users that want to acquire information on the pollutant levels in traffic-congested areas. Civilians such as joggers are likely to plan their travel accordingly to avoid areas if $PM_{2.5}$ levels get too high.

For the environmental sector, this study can help contribute to air pollution awareness in the country, in which such data can be utilized when creating plans and protocols to combat the rising concern for the country's air quality. The system is also open source so it is a benefit for the general public to use without needing a full set of gear to check on pollution levels.

Lastly, as interest in the computer vision field of vehicle identification and recognition systems increases, this study can contribute to future research in said field. The study can be of help to future researchers on the topics of tracking vehicular greenhouse gas emissions. This may also provide data to vehicle image databases through the contribution of the local vehicles (Jeepneys and Tricycles) from the Philippines.

Chapter 2

Review of Related Literature

This chapter discusses the features, capabilities, and limitations of existing research, algorithms, or software that are related/similar to Ha.Zee. Ha.Zee, as an application, identfied the vehicles passing across the camera feed and calculates their $PM_{2.5}$ emission average

2.1 Air Quality Monitoring Systems

Air quality monitoring systems are systems that collect data to record and analyze atmospheric emission levels. There are various systems for air quality monitoring. Zoogman et al.(2017) showcased in a journal the use of satellite imagery for large-scale air quality monitoring. They call this instrument TEMPO (Tropospheric Emissions: Monitoring of Pollution). It is an instrument that collects data on tropospheric emissions such as NO_2, SO_2, H_2CO, Methane, etc. from a satellite in a geostationary orbit. This system is wide-range and precise, however, access to the equipment is limited. A more accessible air monitoring system was made by Zheng et al. (2016) using several sensors. This system makes use of low-power wide-area network (LPWAN) to give it a wider coverage compared to the IoT (Internet-of-Things) and the air quality data can be accessed through a mobile application. These systems make use of dedicated sensors to collect emission data whereas this project will make use of computer vision and machine learning.

2.2 Air Pollution from Vehicles

The Philippines currently has a problem with air pollution. According to Tantengco & Guinto (2022), the Philippines' $PM_{2.5}$ concentrations in urban areas exceed the WHO guideline value. They further state that the Philippines' $PM_{2.5}$ levels reach $58.4ug/m^3$ in traffic sites of Metro Manila during the dry season. Though there could be different sources of air pollution, 65 percent of the air pollutants come from mobile sources such as cars, motorcycles, trucks, and buses (Environmental Management Bureau, 2015).

Furthermore, CO_2, a component of greenhouse gasses, totaled "30 million tons and 56 thousand tons of particulate matter" (Fabian & Gota, 2009) in the Philippines and the transport sector contributed to 38 percent of fuel combustion back in 2000. The authors have noted that the motorized vehicle count would double by 2020. The increase in motorized vehicles also means an increase in their air pollution contribution.

A study by Lu (2022) analyzes the emissions of vehicles due to their impact on air pollution and road-environmental safety. The results show that from 2018 to

2019, two hundred eighty-two vehicle emission standard violations were recorded by the Land Transportation Organization (LTO) office. All of these violations were due to smoke-belching from vehicles. Another result to note was that all the violations were during work hours (6:00 AM to 5:00 PM). The vehicles caught for dangerous emissions were more than 10 years old, with one-third between 10 to 19 years old. The paper concluded that not only ensuring safe vehicle emissions can play an important role in reducing air pollution, there is a need for implementation and monitoring of said vehicle emissions to be within a safer threshold. The researcher notes that the Philippines still needs improvement in addressing the concerns of vehicles contributing to air pollution.

A recent paper by Rito, Lopez, and Biona (2021) raises the concern of quantifying traffic flow, which in this context, is also used for calculating the emission and energy consumption factors. The researchers state that calculating traffic flow has other researchers "deal with complex and arduous tasks, especially when conducting actual surveys". In this paper, the researchers instead utilized crowd-sourced data from Google Maps to estimate mobile emissions and energy use from the traffic flow of the road. The method was used on the EDSA highway in the Philippines and managed to garner an 8.63% error concerning the total vehicle count.

2.3 Vehicle Detection and Tracking

Vehicle detection is a method of identifying a vehicle via a camera. Research on this method started being conducted during the late 1970s (Nath & Deb, 2012) and as more vehicles enter our roads, there has also been more interest in the topic. Meng et al. (2020) defined vehicle detection-based computer vision as aiming at identifying and locating vehicles through digital images or videos. They further simplify the idea by stating that vehicle detection detects “blocks”, which reflect the vehicle’s position from the images and videos.

A paper by Yang et al. (2020) proposed an “object tracker–detector combined with an object tracking algorithm” for tracking vehicles in traffic. They created the tracker by combining strategies for the You Only Look Once (YOLO) model (which will be talked about in section 2.4) with a correlation filter (CF) tracker. To elaborate on object detection, a detection box merge strategy was used for YOLO. This is to prevent the algorithm from partially detecting an object or detecting it more than once. For the tracker, a “deep feature-based CF tracker” was designed. Lastly, to combine both into a tracker-detection program, a tracker was “first used to predict the location of an object in the subsequent frame.”

Another process to detect and track the vehicle would be through background subtraction. Background subtraction, according to Huang BJ. et al. (2017), is used to extract moving objects and then filter unwanted images through image processing tools.

A recent study by Li et al. (2022) studies another method of vehicle detection and recognition – via infrared image and feature extraction. The paper states that due to infrared images having shortcomings such as poor contrast or blurred edges, they mainly studied the color space preprocessing of the image with the use of the threshold segmentation method and infrared image enhancement to separate the vehicle and the background. Techniques such as the median filter and the improved histogram equalization are then used to remove the noise from the infrared image and to enhance the contrast of the image, respectively. The vertical Sobel operator is then selected to enhance the vertical edge of the image. The vertical Sobel operator is used to enhance the vertical edge of the image. Lastly, vertical edge symmetry, aspect ratio, and gray-scale symmetry are utilized for vehicle detection and recognition.

2.4 Object Detection Algorithms

Object detection in the context of this study, involves detecting an instance/instances of objects from one or several image classes (Amit, Felzenszwalb, & Girshick, 2020). The same researchers state that object detection systems construct a model for object class via a "training example set". The following are some algorithms utilized in constructing object detection models:

2.4.1 YOLOv5

Yolov5 is a pre-trained algorithm that uses a system of grids to detect objects from images or videos (https://docs.ultralytics.com/). One application of this algorithm was done by Yan et al. (2021) for an apple-picking robot. YOLOv5 was used to identify apples, however, the algorithm cannot detect apples that are safe to pick and those that are not. This may cause the picking arm of the robot to break if it tries to grasp an apple that is occluded by a solid object. They solved this problem by improving on the modules used for the algorithm. This is not a problem for this project as it only counts the number of vehicles without interacting with them.

In a study done by Zhou et al. (2021), they applied YOLOv5 algorithm to detect safety helmets on workers. The algorithm had an average detection speed of 110 fps in real time. The model, which was trained and tested using 6045 data sets, proved to be viable for real-time detection with a 94.7% effectiveness.

2.4.2 Region-based Convolutional Neural Networks

Region-based Convolutional Neural Networks or R-CNN, is a technique that uses Neural networks to detect objects; this algorithm requires a hefty processing time (Cuong, Trinh, Meesad, & Nguyen, 2022). The researchers of that study further mention the “area suggestions”, which is an image that is extracted into small dimensions and used as inputs to the R-CNN model. The R-CNN model then uses a selective search method to extract reference ranges, then the areas are divided into a set of objects, followed by selective searching to provide candidates suggestions. Once finished, these ‘proposals’ are sent to the cumulative Neural Network (CNN) and a Support Vector Machine (SVM) is used to classify the presence of an object.

In a study done by Rafique et al. (2017), they utilized the R-CNN, along with its successors: Fast-RCNN and Faster-RCNN, to provide solutions for detecting vehicle license plates. It was used with the goal of license plate detection in every frame of a video, detection of partial or obscured license plates, and the detection of license plates whilst using a moving camera on moving vehicles.

2.5 Vehicle Recognition/Identification Applications

In this study, Vehicle Recognition or Identification Applications would be considered as applications that either use any form of video-based software in locating the vehicle on the display feed; or software where static images can be used in identification. Chintalacheruvu and Muthukumar (2012) state that video-based vehicle detection technology has features such as: "non-intrusiveness and comprehensive vehicle behavior data collection capabilities", and that it has become an integral part of f Intelligent Transportation System (ITS)

V-App Vehicle Detection is a real-time vehicle detection system that utilizes the visual analytics provided by Meraki Smart Cameras and a License Plate Recognition function to 'overcome the limitations of common sensors'. It also has features such as vehicle distribution, which detects transit vehicles in an area by grouping them into categories; vehicle count and directions, which gets info on the total amount of vehicles transited and their direction details; and average busiest hours, which shows the higher transit and occupancy peaks in a graph (V-App - Vehicle Detection, n.d.).

BitRefine Heads is a computer vision platform that "utilizes deep learning algorithms to perform high-level visual analysis". It is a platform that detects everyday objects from any angle and also has a vehicle detection system. The recognition module is pre-trained and can detect vehicles as well as recognize the car's model based on the visual features. It gets its video source from an Real Time Streaming Protocol (RTSP) stream from an IP camera. The video then goes to a neural classifier that locates the vehicle in the frame and identifies its class using its own vehicle recognition module's database. The tracking module then takes the results and builds the vehicle's movement track. Then it passes additional images of said vehicle to the neural module to check whether the class is correct (BitRefine Heads, n.d.).

2.6 Vehicle Emission Calculator Applications

In this study, a vehicle emission calculator application would be regarded as an application that provides the total emission count or estimate of a vehicle after given inputs such as: vehicle type, vehicle make and model, fuel type, and the like.

The $PM_{2.5}$ Footprint Calculator v1.01 is an online web browser tool by the constituents of Mahidol University, Thailand. It calculates the primary and secondary $PM_{2.5}$ emissions ($PM_{2.5}$, NOx, NH_3, and SO_2) by asking for the distance traveled, age of the vehicle, fuel type, and city location. Due to the calculator being "developed as a tool for enhancing environmentally sustainable passenger transport in Thailand," it also displays information that assesses the health costs of health impacts of a vehicle. The effect of the emissions on humans' health is calculated using Disability-Adjusted Life Years (DALY) – which, according to the World Health Organization (n.d.), One DALY represents the loss of the equivalent of one year of full health. The calculator is divided into different vehicle types, each having its dedicated page for calculating the $PM_{2.5}$ levels (*PM2.5 footprint calculator-Overview*, 2021).

The Myclimate Car calculator is an online web browser application that determines the CO_2 emissions of a car during its travel. The application asks for the distance traveled, along with the fuel type and fuel consumption. Users also have the option to enter the cart type (compact, mid-range, luxury/SUV/Van) to add to the calculation of the CO_2 amount. The basis of this calculation is through the utilization of the ecoinvent database (Version 3.6), using the IPCC 2013 (Intergovernmental Panel on Climate Change) evaluation method. The emissions are calculated per vehicle kilometer (vkm). The application creators further note that there is an uncertainty margin of 5% added to the emissions due to statistical values used in the calculations (myclimate Foundation, n.d.).

The Next Greencar Make/Model Search Tool is an online car make and model search tool by Nextgreencar.com, a website established in 2007 to help car buyers transition from "fossil cars" to electric cars. This search tool takes the input of a car's manufacturer and/or a specific model to provide results of: tail-pipe CO_2, NOx, particulate emissions, and the NGC Rating. NGC Rating or Next Green Car Rating is a rating developed by the company to assess the environmental impact (Lilly, n.d.). The site then lists all the cars that satisfy the query, allowing users to compare them between their emissions.

The aforementioned applications use different techniques to calculate the harmful emissions from different vehicles but they commonly share the same process of asking for input: from the user via typing in the required information to output the estimated $PM_{2.5}$, CO_2, NOx, etc. emissions. Ha.Zee, while utilizing the same process of using predetermined pollutant levels being assigned to a vehicle, relied on computer vision training instead of user input to determine the vehicle and estimate the amount of the pollutant they would emit.

2.7 Summary

As the usage of vehicles in the Philippines rapidly increases through the years, it also starts becoming the main contributor to air pollution in the country – a problem that the Philippines is still trying to mitigate. The studies mentioned above mention that in an attempt to solve this concern, emissions such as fine particulate matter ($PM_{2.5}$) from mobile vehicles are collected and analyzed by making applications that can keep track of the emissions by identifying the type of vehicle on screen.

The type of vehicle can be identified through a collection of images of vehicles gathered by the researchers, being used for the model training of Ha.Zee. This can be utilized to create a system to identify vehicles via either still images and/or video. This chapter presented studies from different researchers that produced vehicle identification and recognition systems through machine learning and computer vision. Some of the applications used as an example can identify objects from a live video feed and produce results that list the vehicle type. Most of the related applications for vehicle tracking use an in-house system that is not publicly available to use without having to pay for them. The researchers instead utilized YOLOv5, an open-source pre-trained algorithm that uses a system of grids to detect objects from images or videos to be used in the study.

The recent studies mentioned make note of the interest in vehicle tracking systems and their benefits in managing traffic. This information, combined with the goal of reducing the emissions from vehicles that contribute to pollution, were used to support the researchers' purposes of the study.

Chapter 3

Research Methodology

This chapter lists and discusses the specific steps and activities performed by the researchers in developing Ha.Zee.

3.1 Technologies Used

The technologies that were utilized for this project are the following in the following subsections

3.1.1 Roboflow

According to Bhattacharyya (2020), Roboflow is a " computer vision developer framework for better data collection to preprocessing, and model training techniques". Roboflow also offers a suite of browser applications to preprocess and preparation of the data for computer vision and machine learning. Roboflow annotation will be used to manually set bounding boxes for model training and image augmentation for the manipulation of images (*Overview - Roboflow*, n.d.).

3.1.2 Jupyter Notebook using Google Colab and Python

Jupyter Notebook and Python will be used for training and fitting the data. Jupyter notebook, using google colab, offers free GPU with CUDA for processing, and Python, the programming language, offers essential libraries for machine learning (*Google Colaboratory*, n.d.).

3.1.3 A case for YOLOv5

Yolov5 is a pre-trained algorithm that uses a system of grids to detect objects from images or videos (https://docs.ultralytics.com/). This tool will be used for the vehicle detection in this project.

YOLOv5 is one of the commonly used algorithm for object detection. It is faster than other object detection algorithms like Region-based Convolutional Neural Networks (RCNN), Fast RCNN, and Faster RCNN. Gandhi (2018) wrote in an article the comparison between the RCNN algorithms and YOLOv5. He said that the major drawbacks of RCNN are that it classifies 2000 regions per image every time it runs, it cannot run in real time and it is a fixed algorithm. Fast RCNN employs a similar algorithm to RCNN but instead of classifying regions everytime, it uses CNN to generate a convolutional feature map where the bounding regions are derived. Faster RCNN improves upon this by using a different network for predicting the regions of the proposal. In his comparison, he found that Fast RCNN improves on the speed of RCNN significantly. He also mentioned that Faster RCNN, the fastest of the RCNN algorithms, is viable for real-time object detection.

Aside from the speed of the algorithm, Ultralytics (https://ultralytics.com/) provides extensive documentation of YOLOv5. This is one of the factors that affect the decision of using YOLOv5 for the study.

3.2 Research Activities

3.2.1 Data Gathering

This study planned to identify an area's pollution level through calculating average of the emissions coming from the cars on the road. In doing so, data of the vehicles, its identification, and its emission rates were needed for the study. Through the use of a camera, footage of the vehicles in traffic were recorded to gather the data of vehicles in traffic. This was utilized in training and testing for the software to recognize the vehicles on the video feed. The vehicle emissions were taken from a study by Rito et al. (2021).

Due to the study being conducted in the country of the Philippines, vehicles such as the local jeepney and tricycle do not have readily available image datasets. Videos and images of said vehicles in traffic were taken from different angles using a phone camera.

3.2.2 Preprocessing

Preprocessing the data includes defining the bounding box of the vehicles in the training data and augmenting the images to make the model perform better. Roboflow has an annotation tool that can be used for training the model to detect a vehicle in an image and its type. Augmentation of the images will be done by the YOLOv5 algorithm automatically given that the Albumentation library is installed. According to Dilmegani (2021), Augmentation can be used for transforming the images allowing the model to diversify its training data set making it perform better.

Annotation Method

Annotation of the vehicles consist of: full image annotation and vehicle cropping. In full image annotation, the entire image or video frame is used and every vehicle present is then selected and categorized for the Roboflow tool to save. Vehicle cropping is when a vehicle/small group of vehicles is/are cropped from the source image/video frame and is then annotated by the tool. This was done when a specific type of vehicle was needed for the database.

Figure 3.1 shows the full image annotation, wherein every bounding box is color coordinated to the type of vehicle used in the study.

Figure 3.2 uses a vehicle cropping method. In this specific example, the researchers needed more data of the tricycle. Hence, samples of tricycles over different images/frames from videos were cropped before being uploaded to the Roboflow annotation tool.

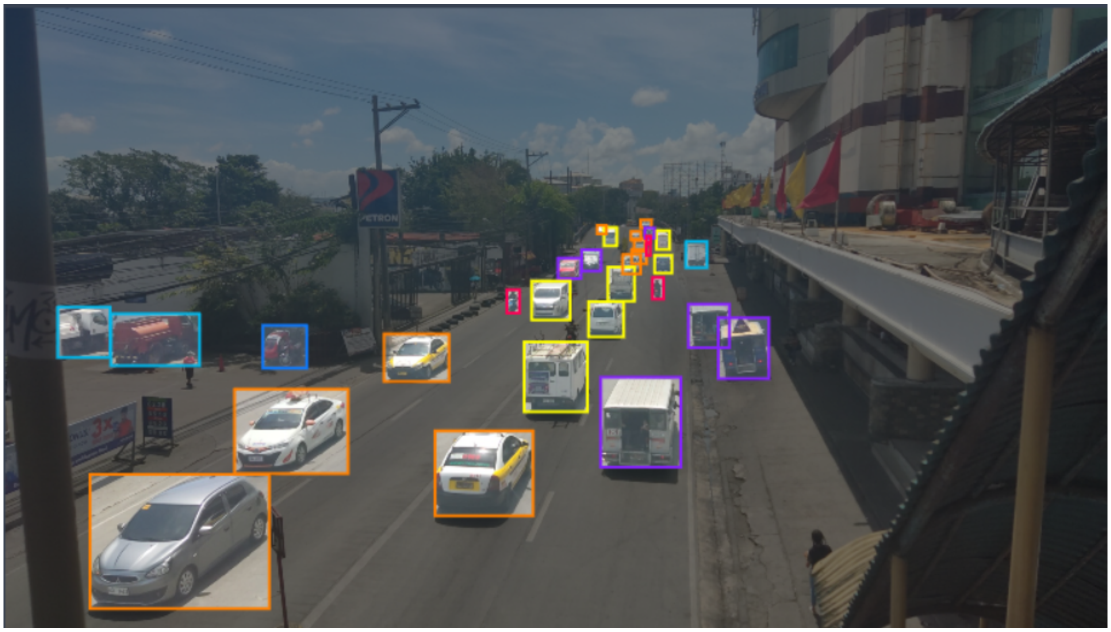


Figure 3.1: Full image annotation

Figure 3.2: Image cropped before annotation

3.2.3 Training and Performance Testing

The training data will be separated into different classes and is also the bases of classification for the training: cars, jeepneys, motorcycles, tricycles, trucks, and utility vehicles. The training will be done using the "train.py" in the YOLOv5 repository with the following line of code used in the Google Colab Notebook for training the data:

```
!git clone https://github.com/ultralytics/yolov5
%cd yolov5
%pip install -qr requirements.txt
import torch
import utils
display = utils.notebook_init()
```

The data was trained with 16 batches and 300 epochs. These are reflected by the following terminal command for training.

```
!python /content/yolov5/train.py --batch 16 --epochs 300 --data
    /content/drive/MyDrive/College/SP/data/data.yaml --weights
    yolov5m.pt --cache
```

The “/content/drive/MyDrive/College/SP/data/data.yaml” is the yaml file that contains the information about the different classes in the dataset. Below is an example of data.yaml.

```
train: ../train/images
val: ../valid/images
test: ../test/images

nc: 6
names: ['Car', 'Jeepney', 'Motorcycle', 'Tricycle', 'Truck',
    'Utility Vehicle']

roboflow:
workspace: special-project
project: sp-cmsc198.1
version: 3
license: CC BY 4.0
url:
    https://universe.roboflow.com/special-project/sp-cmsc198.1/dataset/3
```

The file contains the directory to the train, validation, and testing data as well as other important information such as the number of objects and their names that is needed by"train.py".

3.3 Model Applictation

The program detect.py was run to detect objects from an external device or video file using the trained model. When detect.py is run it will start to list the objects it detects. An average emission of each vehicle type will be used. The study from Rito et al. (2021) provides a table that quantifies the emission factors of energy consumption of multiple vehicle types, 6 of which are to be used in this study. The emission data from the study will be utilized for the application.The table below shows the grams of emissions per kilometer. The averages of the vehicles' $PM_{2.5}$ emissions at a given time will be displayed on the corner of the video recording and will be periodically updated as vehicles enter and leave the camera's or video's line of sight.

Table 3.1: Emission factors per vehicle type ($g_{emissions}/km$)

Vehicle Type	$PM_{2.5}$	CH_4	N_2O	CO_2
Tricycle	0.0562	4.0906	0.0021	66.8747
Motorcycle	0.0336	2.3022	0.0015	60.0983
Jeepney	0.8466	0.2357	0.0316	668.7415
Car	0.0221	0.7408	0.0099	109.8958
Utility	0.1430	0.3538	0.0063	92.4039
Light Truck	0.7519	0.3648	0.0226	842.0852

Calculating the $PM_{2.5}$ Emission Estimate

The system, after assigning values to each vehicle, calculated the vehicles' $PM_{2.5}$ emissions using the following equation:

$$\text{vehicle } PM_{2.5} \text{ estimation} = \frac{\sum(\text{total count of vehicle x * vehicle x's assigned } PM_{2.5} \text{ value})}{\text{Total vehicles detected}} \tag{3.1}$$

The total count of the vehicle x was multiplied to its assigned $PM_{2.5}$ value. The sum of the product of every vehicle was then divided by the total number of detected vehicles on the frame to get the average. This results in the estimated $PM_{2.5}$ value being produced from vehicles in a current frame.

3.4 System Architecture

A separate system for inference was made to allow the program to count the number of vehicles and approximate emissions. The available program for detection (detect.py) was not used because it was meant to run for general cases and not for videos and webcam footage of vehicles specifically as indicated in the github page for YOLOv5 (https://github.com/ultralytics/yolov5).

As shown in Figure 3.3, the system contains four classes. The VehicleDetection class is the class that makes objects for detecting vehicles and counting emissions, and this class utilizes three other classes: VehicleCounter, EmissionCounter, and DataLogger. This composition was done so that one class only has one role. The VehicleCounter class serves as a click counter for different vehicles and was used to count vehicles in a single frame. The EmissionCounter class stores the emission estimate in a double-ended queue so that the average emissions for a number of frames can be calculated. The DataLogger class logs the data into an external CSV file.

The VehicleDetection class is responsible for analyzing each frame of a video, detecting the objects from the frame, and incorporating the three classes to count vehicles from a video, averaging the $PM_{2.5}$, and exporting the data into a CSV

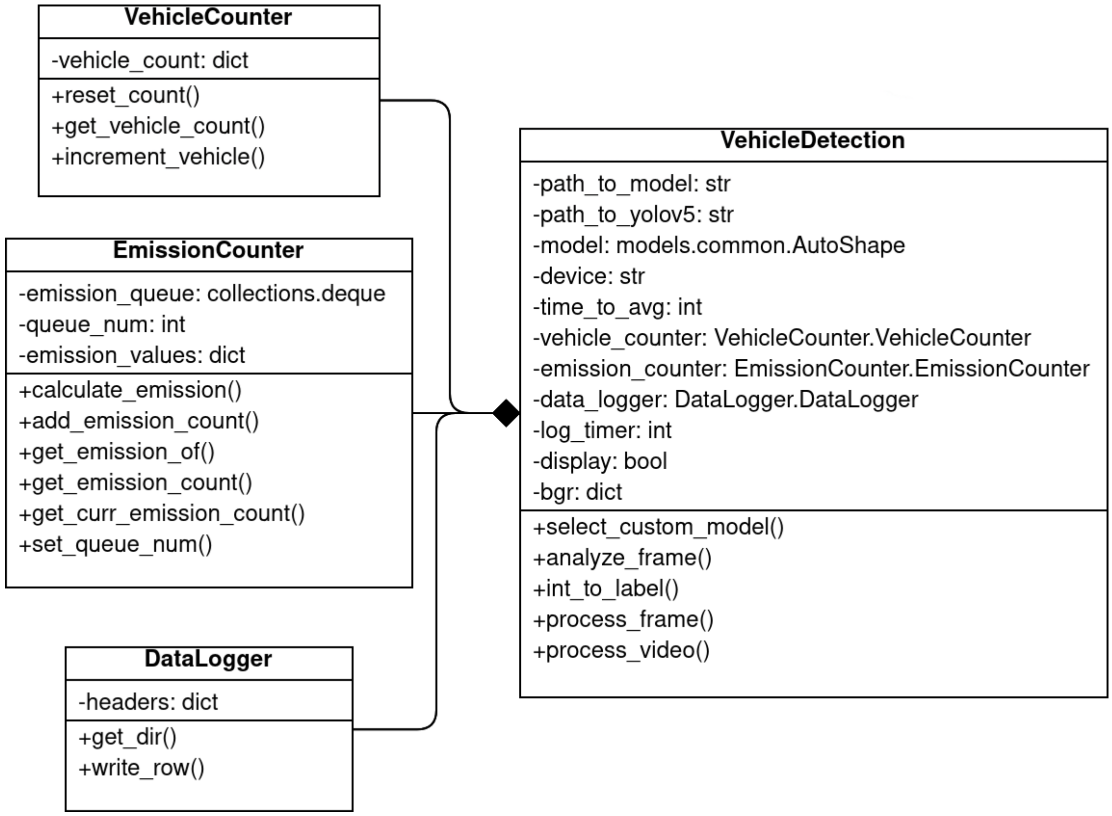


Figure 3.3: Object Detection Prototype used for Traffic recorded in street view file.

Chapter 4

Results and Discussions

4.1 Training Results

4.1.1 Loss Values and Metric Progression

After training the dataset, the YOLOv5 algorithm provided statistics to show how the model performs and progresses after a certain number of epochs, which in this study, 100 epochs were used.

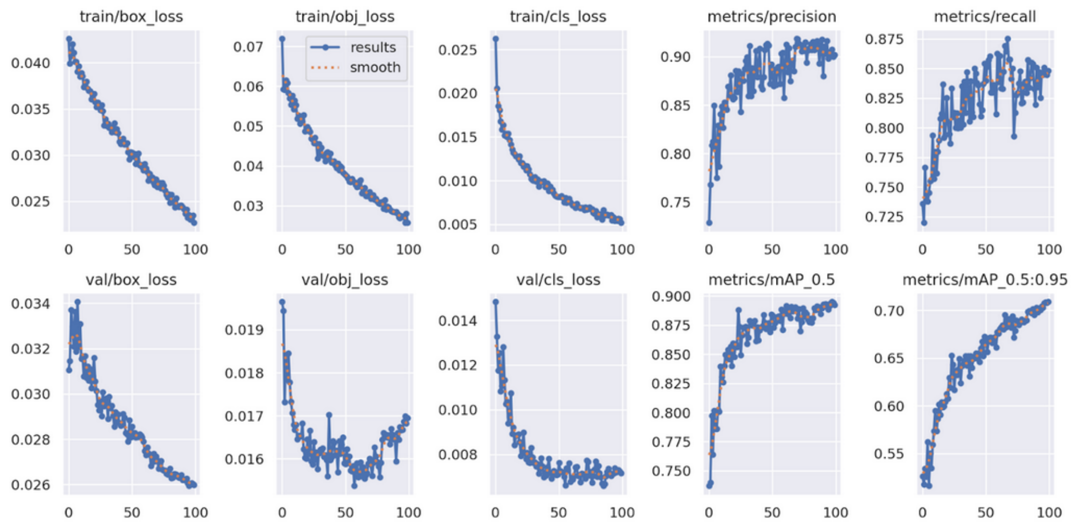


Figure 4.1: Graph depicting the loss value and metric progression during training with 100 epochs

Figure 4.1 shows the statistics of how the data set performed during training. The loss values used in the graph were box loss, objectness loss, and classification loss. These loss values represent the performance of the model during training concerning the ground truth (Hui, 2022). Box loss refers to the errors in the location and size of the predicted boundaries, objectness or confidence loss measures the probability that an object exists in the region of interest, and classification loss measures how effective the model is in predicting the correct class. The graph shows that as the training progresses the loss values followed a downward progression which meant that the model is making fewer errors as training continues for both the train and validation sets.

For the metric values there is precision, mean average precision, and recall. Recall is a metric that measures the portion of the true values that are predicted true while precision measures the portion of the predicted values that are true (Powers, 2008). Mean average precision was calculated by averaging the precision of each class and then averaging all the precision of every class (Shah, 2022). In the graph, the metrics show an upward, although not linear, progression along 100 epochs, which means that the model is improving continuously for training and validation.

4.1.2 Confusion Matrix/F-1 Score Calculation

After training the model, a confusion matrix was provided (figure 4.2) which depicts the normalized count of predicted versus true values of the six (6) classes and the background in the model, and is useful for getting the metrics necessary to measure the performance of the model. The normalization of the confusion matrix was obtained by dividing each of the row's elements to the sum of the entire row. The normalized elements all add up to 1.0. This allows the matrix to be read in percentages.

The diagonal boxes (which are dark blue in figure 4.2)indicate the true positives, wherein the predicted vehicle during training is actually the proper vehicle type. The true positives of every vehicle all have a good score, with tricycles having the highest value of 0.90 or 90% accurately detected as tricycles.

The columns, excluding the diagonal elements (dark blue) of true positives, indicate the false negatives. These are instances where the vehicle was detected as another type (i.e. 1% of cars were detected as either a jeepney, motorcycle, or utility vehicle).

The "background" class was included for the instance where the detected object is anything but vehicle. The background row signifies the percentages of the vehicles not being detected. Meanwhile, the background column signifies the instances where a vehicle was detected but in actuality, the detected object is part of the background or is not a vehicle. The background does not have true positive as it is not part of the training process.

The confusion matrix was made using the validation set of the training data with a confidence of 0.25 based on the source code of YOLOv5. This might cause discrepancies when running the model with real-world data as the confidence value might vary.

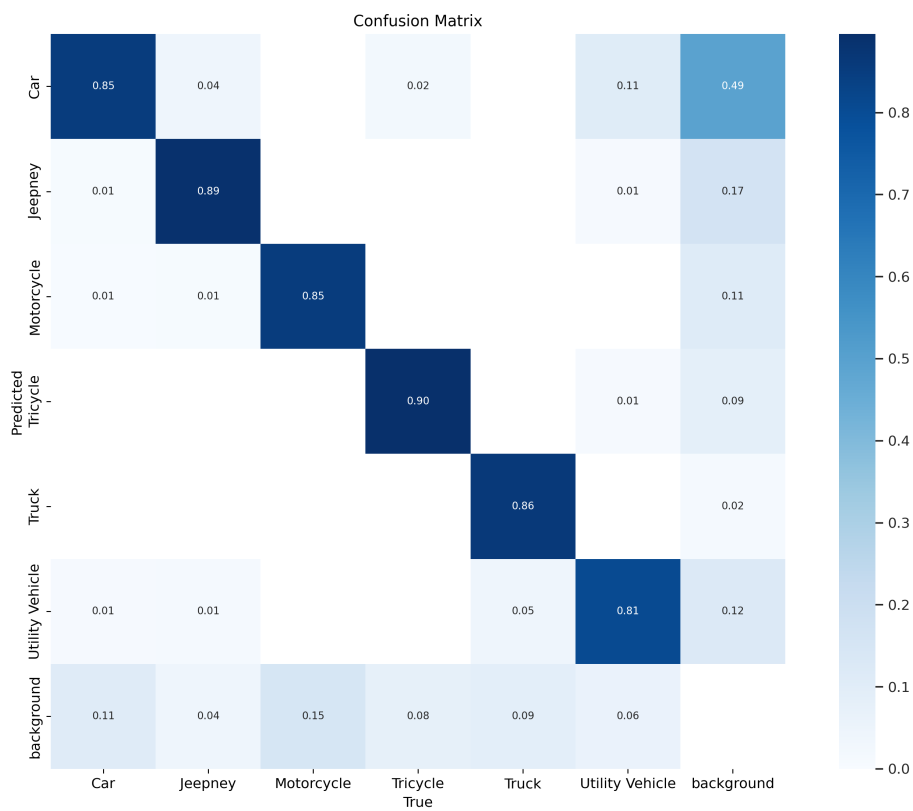


Figure 4.2: Confusion matrix of the training

For this system, the F1-score metric was used because there was a class imbalance due to the limitations of the locations where the data was taken as shown in

Figure 4.3. F1-score is a better metric to use compared to accuracy when there is a class imbalance because it measures using the number and type of errors, unlike accuracy which only calculated the number of correct predictions (Korstanje, 2021). Furthermore, the researchers decided that accuracy is not an ideal metric to represent what the model should be predicting, because as a multi-class object detection model that relies on the exact number of vehicles to calculate for emission, it is important to take into account not only the number of correct predictions but also the type of errors, which is what the F1-score metric provides. However, for comparison, the macro averages of the accuracy metric and F1-score metric will be calculated.

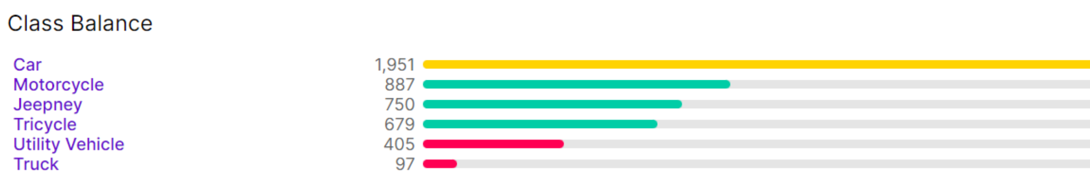


Figure 4.3: Graph of class balance

To get the F1-score for multi-class classification must be calculated first with the following formulas which were taken from the Towards Data Science article (Korstanje, 2021) and from Powers (2008):

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}} \tag{4.1}$$

$$\text{Precision} = \frac{\text{class TP}}{\text{class TP} + \text{class FP}} \tag{4.2}$$

$$\text{Recall} = \frac{\text{class TP}}{\text{class TP} + \text{class FN}} \tag{4.3}$$

$$\text{F1} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

For calculating the precision, we can get the values available in the confusion matrix in Figure 4.1. In the following example, the precision for the "Car" class is calculated using the formula from equation 4.1:

$$\text{Precision}_{\text{Car}} = \frac{0.85}{0.85 + (0.04 + 0 + 0.02 + 0 + 0.11 + 0.49)}$$

$$\text{Precision}_{\text{Car}} = \frac{0.85}{1.51}$$

$$\text{Precision}_{\text{Car}} = 0.5629139073$$

The value 0.85 was taken from the cell of the predicted and true value for the "Car" class which means that it is a true positive because the predicted objects were the true objects. The false positives from the matrix are the other classes that are predicted as the "Cars" class in the confusion matrix. Applying the method above but for the recall value of the "Car" class:

$$\text{Recall}_{\text{Car}} = \frac{0.85}{0.85 + (0.01 + 0.01 + 0 + 0 + 0.01 + 0.11)}$$

$$\text{Recall}_{\text{Car}} = \frac{0.85}{0.99}$$

$$\text{Recall}_{\text{Car}} = 0.8585858586$$

Similar for the calculation of precision, we get 0.85 as true positive from the confusion matrix. In this case, however, the formula uses false negatives which are classes from the confusion matrix that detected a car for objects that are not cars. Using both precision and recall the F1-score can be calculated by:

$$\text{F1}_{\text{Car}} = 2 * \frac{0.5629139073 * 0.8585858586}{0.5629139073 + 0.8585858586}$$

$$\text{F1}_{\text{Car}} = 2 * \frac{0.4833099204}{1.421499766}$$

$$\text{F1}_{\text{Car}} = 2 * 0.34$$

$$\text{F1}_{\text{Car}} = 0.68$$

Lastly, to calculate for accuracy the number of true predictions (true positive and true negatives) was divided by all the values in the confusion matrix. Below is the calculation of accuracy for the "Car" class:

$$\text{Accuracy}_{\text{Car}} = \frac{0.85 + 5.33}{6.98}$$

$$\text{Accuracy}_{\text{Car}} = \frac{6.18}{6.98}$$

$$\text{Accuracy}_{\text{Car}} = 0.8853868195$$

With this methods, the performance metrics of each class was calculated and the results are shown in Table 4.1.

Table 4.1: Table of performance metrics of each class

Class	**Precision**	**Recall**	**F1-Score**	**Accuracy**
Car	0.5629139073	0.8585858586	0.68	0.8853868195
Jeepney	0.8240740741	0.898989899	0.8599033816	0.9584527221
Motorcycle	0.8673469388	0.85	0.8585858586	0.9598853868
Tricycle	0.9	0.9	0.9	0.9713467049
Truck	0.9772727273	0.86	0.914893617	0.9770773639
Utility	0.81	0.81	0.81	0.9455587393

Table 4.1 showed the Precision value calculated from each class. In the table the class with the lowest precision is the "Car" class which is also the class with the most samples, this is due to the "Car" class being over-represented in the dataset therefore making the system overfit for that particular class, meaning it might detect a car even though it should be a different object (Raj, 2019). On the contrary, the "Truck" class had the highest precision value of approximately 0.9773, which meant that every time the model detects a truck there is a high chance that the model has predicted true (Powers, 2008), even though the "Truck" class was underrepresented in the dataset. This might be because the trucks are sparse in the testing dataset thus making errors rare. Meanwhile, for recall, the "Tricycle" class had the highest value, (0.9) which meant that the model has a

high chance of detecting true tricycles in a frame (Powers, 2008). Finally, all classes not including the “Car” class had an F1-score of at least 80%. The “Car” class has an F1-score of 68%, the lowest of all classes in this model, which was contributed by its low precision value.The accuracy of the model closely follows the ranking of the F1-scores, the only difference is that the ranks of the “Jeepney” and “Motorcycle” classes swapped places for the accuracy metric, also the accuracy of each class was at least 88%.

To find the overall F1-score of the model the macro average of the F1-scores was used. Macro averages were used instead of micro and weighted averages because macro averages put equal importance on all the classes in the model whereas weighted averages are ideal for datasets with classes having different degrees of importance and micro averages are ideal for datasets with a balanced distribution of classes (Leung, 2022). The Macro averages can be calculated by finding the average of the F1-score and accuracy metric from each class (Leung, 2022). To find the macro average of the F1-score and accuracy the average of their respective column were calculated from Figure 4.3.

Equation:

$$\text{metric} = \frac{\text{sum of all values of the metric}}{\text{number of classes}} \tag{4.5}$$

For F1-score:

$$\text{F1} = \frac{0.68 + 0.8599033816 + 0.8585858586 + 0.9 + 0.914893617 + 0.81}{6}$$

$$\mathrm{F1} = \frac{5.023382857}{6}$$

$$\mathrm{F1} = 0.8372304762$$

For Accuracy:

$$\text{Accuracy} = \frac{0.89 + 0.96 + 0.96 + 0.97 + 0.98 + 0.95}{6}$$

$$\text{Accuracy} = \frac{5.70}{6}$$

$$\text{Accuracy} = 0.95$$

The model had an accuracy of approximately 95% not taking into account the errors it made. However, if the errors were considered the model only has an accuracy of approximately 84% which is lower but is more ideal for the imbalance distribution of classes in the dataset. Nonetheless, the model was accurate enough to be used for object detection.

4.2 Object Detection

The weights obtained through training were used in pre-recorded videos to determine if the weights were trained successfully and to see how they perform in actual video footage. The following locations were chosen by the researchers to be used for the study: Valeria St. , De Leon St., Diversion Road, Lacson St., Roxas Ave. The locations were chosen for their variety in vehicle congestion and vehicle type (i.e. Diversion Road with a higher vehicle count; Roxas with more tricycles). These recordings were then processed using the custom vehicle detection Python program. The "processing" includes detecting the vehicles; drawing bounding boxes; and calculating and displaying the approximation of the emission of that area.

Using the trained model, the following are the results showing the Ha.Zee system detecting vehicles and their respective types, along with the vehicle $PM_{2.5}$ emission tracker displayed on the upper left portion of the frame. The $PM_{2.5}$ value here means the weight of particulate matter of at most 2.5 micrometers in diameter in kilograms for every kilometer. The higher the value for $PM_{2.5}$ the greater the health risk, but the interpretation of the value on the particulate matter and its consequences falls outside the scope of this study. To validate the detection model of the system, the researchers compared the ground truth of the total vehicles in the frame via manual counting to the detected vehicles. The information on the system's detected vehicles stored in the CSV file was utilized in this comparison. This section focused on comparing the number of vehicles the system could detect to the actual vehicle count and does not account for the accuracy of identifying the proper vehicle type.

Valeria St. (MaryMart, Iloilo City)

(a) Manual Count: 27 Vehicles

(b) Ha.Zee Count: 17 vehicles

Figure 4.4: Valeria St. Traffic Video Footage

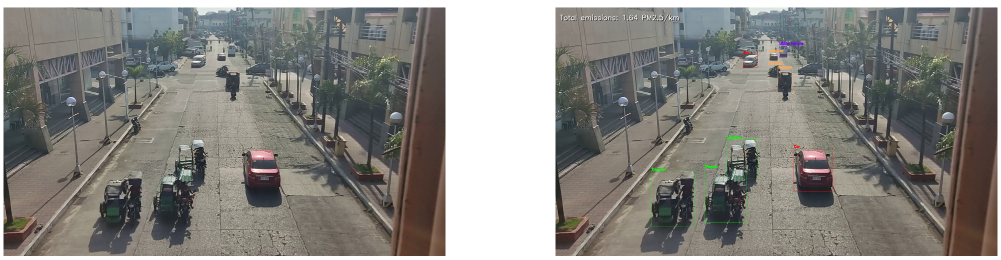
Using the frame above to represent the video footage from Valeria St., Iloilo City Proper; the researchers manually counted 27 total vehicles. The most common vehicle in this frame is the car, with a count of 12. After using the trained model, the system returned a count of 17 vehicles. In this instance, the most common vehicle type detected was the jeepney, with a count of 8. The road being a route for jeepneys could have influenced their frequency in appearance.

The table below shows the average of vehicles present over the course of the video duration. The average of vehicles appearing in the video is obtained through the sum of the average of a vehicle type's appearance per second. The same process was applied to the vehicles detected by the system. The ratio between the two counts is then calculated. Of the total average of manually-counted vehicles, only 64.98% were detected by the system.

Table 4.2: Ratio of Manual vs. Detected average vehicles counted (Valeria St.)

Vehicle type	**Manual Count Avg.**	**Ha.Zee Count Avg.**
Car	11.72	4.55
Jeepney	9.72	8.18
Motorcycle	4.55	3.55
Tricycle	0	0.09
UV	3	1.18
Truck	0	0
Total Average	27	9.145
Ratio/Percentage	0.6498	

De Leon St. (Robinsons Place, Iloilo City)



(a) Manual Count: 15 Vehicles

(b) Ha.Zee Count: 10 vehicles

Figure 4.5: De Leon St. Traffic Video Footage

Using the frame above to represent the video footage from De Leon St., Iloilo City; the researchers manually counted 15 total vehicles. The most common vehicles in this frame are motorcycles and tricycles, both with a count of 4. After using the trained model, the system returned a count of 10 vehicles. In this instance, the most common vehicle type detected was the jeepneys and tricycles, with a count of 3. It is noted that 2 of the detected jeepneys are false positives.

The table below applies the same calculation processes as the previous location. Of the total average of manually-counted vehicles, only 73.28% were detected by the system.

Table 4.3: Ratio of Manual vs. Detected average vehicles counted (De Leon St.)

Vehicle type	**Manual Count Avg.**	**Ha.Zee Count Avg.**
Car	2.63	2
Jeepney	2	1.72
Motorcycle	3.55	1.36
Tricycle	3.45	3.27
UV	1.63	1.36
Truck	0	0
Total Average	13.27	9.72
Ratio/Percentage	0.7328	

Diversion Road (Jaro, Iloilo City)

(a) Manual Count: 60 Vehicles

(b) Ha.Zee Count: 42 vehicles

Figure 4.6: Diversion Road Traffic Video Footage

Using the frame above to represent the video footage from Diversion Road, Iloilo City; the researchers manually counted 60 total vehicles. The most common vehicle in this frame is the car, with a count of 46. After using the trained model, the system returned a count of 42 vehicles. In this instance, the most common vehicle type detected was also the car, with a count of 27. Cars are a big majority of this video footage, yet have a big gap in the system's detection.

The table below applies the same calculation processes as the previous location. Of the total average of manually-counted vehicles, only 71.60% were detected by the system.

Table 4.4: Ratio of Manual vs. Detected average vehicles counted (Diversion Rd.)

Vehicle type	Manual Count Avg.	Ha.Zee Count Avg.
Car	47.09	29.36
Jeepney	7.90	8.09
Motorcycle	2.72	0.18
Tricycle	1.72	0.27
UV	1	5.82
Truck	1	0.27
Total Average	61.45	44
Ratio/Percentage	0.7160	

Lacson St. (Bacolod City, Negros Occidental)

(a) Manual Count: 15 Vehicles

(b) Ha.Zee Count: 12 vehicles

Figure 4.7: Lacson St. Traffic Video Footage

Using the frame above to represent the video footage from Lacson St., Bacolod City; the researchers manually counted 15 total vehicles. The most common vehicle in this frame is the car, with a count of 9. After using the trained model, the system returned a count of 12 vehicles. In this instance, the most common vehicle type detected was also the car, with a count of 5. This being a highway could be the reason why there is a lack of tricycles.

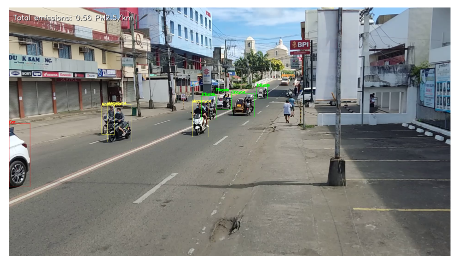
The table below applies the same calculation processes as the previous location. Of the total average of manually-counted vehicles, only 58.42% were detected by the system.

Table 4.5: Ratio of Manual vs. Detected average vehicles counted (Lacson St.)

Vehicle type	Manual Count Avg.	Ha.Zee Count Avg.
Car	10.45	3.18
Jeepney	2.45	1.91
Motorcycle	2.27	2.09
Tricycle	0	0.09
UV	1	2.18
Truck	0	0
Total Average	16.18	9.45
Ratio/Percentage	0.5842	

Roxas Ave. (Roxas City, Capiz)

(a) Manual Count: 10 Vehicles



(b) Ha.Zee Count: 9 vehicles

Figure 4.8: Lacson St. Traffic Video Footage

Using the frame above to represent the video footage from Roxas Ave., Roxas City; the researchers manually counted 10 total vehicles. The most common vehicle in this frame is the tricycle, with a count of 4. After using the trained model, the system returned a count of 9 vehicles. In this instance, the most common vehicle type detected was also the tricycles, with the same count of 4. The high count of Tricycles is due to it being one of the main modes of transportation.

The table below applies the same calculation processes as the previous location. Of the total average of manually-counted vehicles, only 77.89% were detected by the system.

Table 4.6: Ratio of Manual vs. Detected average vehicles counted (Roxas Ave.)

Vehicle type	**Manual Count Avg.**	**Ha.Zee Count Avg.**
Car	0.90	0.54
Jeepney	0.63	0.64
Motorcycle	2.64	1.63
Tricycle	4.45	3.72
UV	0	0.18
Truck	0	0
Total Average	8.63	6.73
Ratio/Percentage	0.7789	

The ratio between of the actual count of the vehicles compared to the amount detected by the system varies between the 5 locations. Roxas Ave., Roxas City's video footage had the highest percentage of 77.89%; while Lacson St. in Bacolod City garnered the lowest percentage of 58.42%. A factor on why Roxas Ave.'s average of vehicles detected is much closer to the average of the actual vehicles manually counted could be due to the fewer vehicles present and their distance from one another. Said distances could have also been the reason why Diversion Road's detected cars are significantly lesser than the counted vehicles, as some cars were obstructed by other vehicles and are sometimes counted as a singular vehicle. Another factor that affected the percentages could have been the quality of the video footage used in the study. All the footages shown in the figures were taken from the smartphones of the researchers and rendered some vehicles pixelated, which led to some vehicles being indistinguishable to the system.

Across the 5 locations, the system is averaged to detect 69.23% of the vehicles on a given video footage. This is a reasonably acceptable rate and could be improved upon if more data on different vehicles of varying image qualities were to be added to the training of the system.

Table 4.7: Average Percentage of Ha.Zee-detected Vehicles.

Location	**Ratio/Percentages**
Valeria St.	0.6498
De Leon St.	0.7328
Diversion Road	0.7160
Lacson St.	0.5842
Roxas Ave.	0.7789
Total Average	0.6923

Chapter 5

Conclusion and Recommendations

5.1 Conclusion

The air pollution problem in the Philippines continues to prevail. While there are machines that calculate air quality in different locations, they are limited to specific stations in the country and are not always available to be used at any given moment. In addition, these types of equipment are also expensive. As one of the latent goals of the Philippine Air Act to help combat the increasing pollution was to increase its awareness, a way to contribute to this was by making these air quality tools more accessible. With this, the researchers sought to address this issue using the currently available technologies.

Computer vision and machine learning are emerging technologies in today's digital world. As vehicles are contributors to rising air pollution (their $PM_{2.5}$ emissions being one of the components), the researchers sought to use the aforementioned technologies to create a system that detects vehicles and identifies their type and the emission they exhaust. Thus, Ha.Zee was developed.

Ha.Zee is a system that detects vehicles and records the average emission ($PM_{2.5}$) they would produce. YOLOv5 was used for the training of a dataset, which contains 1550 images of different kinds of vehicles, and its accuracy was evaluated using precision, recall, and F1-score, and detecting the vehicles in a video, although a separate inference system was developed to fit the purpose of the study.

YOLOv5 is a viable tool for doing object detection on traffic vehicles as it can detect objects almost in real time provided that the equipment used was sufficient. The system was fairly accurate in detecting the relevant objects in a scene with a macro average F1-score of approximately 0.84, even though it was trained with a limited and imbalanced dataset, the "Car" class being over-represented which, consequently, made the model be least accurate in that class.

With the development of Ha.Zee, this shows that it is feasible to use an object detection algorithm such as YOLOv5 to be trained to detect vehicles and calculate an estimate of the $PM_{2.5}$ emissions that they would produce via assigning values to a vehicle type. Though datasets of local vehicle images are sparsely available, the ones taken by the researchers and used in training the system were sufficient enough to create a working product that includes them in the equation.

5.2 Recommendations

The application and its features are not without its flaws. There are some limitations that are brought by lack of equipment and time constraints thereby for future improvements, the researchers suggest that the dataset be populated with higher quality images, variation in the images for each class, and longer training time. Resampling could also be used to balance the dataset to eliminate bias when detecting objects. The model also struggles in low-light conditions which is a consequence of the low sensitivity of the cameras used for taking video footage.

Furthermore, Ha.Zee directly benefits from moving vehicles and has difficulty distinguishing vehicles that are at rest compared to the ones that are parked so it is limited at a certain angle. Further improvements such as only focusing on vehicles that are identified to be on the road would help with this problem.

Lastly, the researchers hope that after this study, there is an increase in the availability of image datasets containing local vehicles in the Philippines. They recommend future researchers in the topic to continue contributing to these datasets for the benefit of future vehicle detection projects.

References

Abano, I. V. (2019, Jun). *In the news: Health experts in the philippines lead the fight against dirty air.* Retrieved from `https://noharm-global.org/articles/news/asia/news-health-experts-philippines-lead-fight-against-dirty-air` (Accessed: 2022-12-05)

Air quality ontario. (n.d.). Ministry of the Environment,Conservarion and Parks. Retrieved from `http://www.airqualityontario.com/science/pollutants/particulates.php`

Akimoto, H. (2004, 01). Global air quality and pollution. *Science (New York, N.Y.)*, *302*, 1716-9. doi: 10.1126/science.1092666

Amit, Y., Felzenszwalb, P., & Girshick, R. (2020). Object detection. *Computer Vision*, 1–9. doi: 10.1007/978-3-030-03243-2_660-1

Bhattacharyya, J. (2020, Dec). *Step by step guide to object detection using roboflow.* Retrieved from `https://analyticsindiamag.com/step-by-step-guide-to-object-detection-using-roboflow/`

BitRefine Heads. (n.d.). *Bitrefine heads vehicle recognition software.* Retrieved from `https://heads.bitrefine.group/use-cases/vehicle-recognition/115-vehicle-recognition` (Accessed: 2022-12-04)

Chintalacheruvu, N., & Muthukumar, V. (2012). Video based vehicle detection and its application in intelligent transportation systems. *Journal of Transportation Technologies*, *02*(04), 305–314. doi: 10.4236/jtts.2012.24033

Cuong, N. H., Trinh, T. H., Meesad, P., & Nguyen, T. T. (2022). Improved yolo object detection algorithm to detect ripe pineapple phase. *Journal of Intelligent and Fuzzy Systems*, *43*(1), 1365–1381. doi: 10.3233/jifs-213251

DENR. (n.d.). *Purchase of air monitoring equipment aboveboard- emb.* Retrieved from `https://ncr.denr.gov.ph/index.php/news-events/press-releases/purchase-of-air-monitoring-equipment-aboveboard-emb` (Accessed: 2022-12-07)

DENR. (2020, March 15). *Denr: Air quality monitoring is a top priority.* Retrieved from `https://www.denr.gov.ph/index.php/news-events/press-releases/1490-denr-air-quality-monitoring-is-a-top-priority` (Accessed: 2022-12-07)

Dilmegani, C. (2021). *What is data augmentation? techniques, benefit and examples.* Retrieved from `https://research.aimultiple.com/data-augmentation/`

Enano, J. O., & Subingsubing, K. (2019, Jun). Clean air act 20 years later: Edsa still 'worst place to be'. *Inquirer.Net*. Retrieved from `https://newsinfo.inquirer.net/1135618/clean-air-act-20-years-later-edsa-still-worst-place-to-be` (Accessed: 2022-12-06)

Environmental Management Bureau. (2015, Sep). *Environmental management bureau — initially established as a supporting ...* Retrieved from `https://emb.gov.ph/wp-content/uploads/2015/09/1-Air-Quality-1.8-National-Air-Quality-Status-Report-2008-2015.pdf` (Accessed: 2022-12-05)

Environmental Management Bureau. (2018). *Emissions inventory 2018.* Retrieved from https://air.emb.gov.ph/emission-inventory-2018/ (Accessed: 2022-12-05)

Fabian, H., & Gota, S. (2009, 01). Co2 emissions from the land transport sector in the philippines: Estimates and policy implications.

Food and Agriculture Organization of the United Nations. (n.d.). *Philippine clean air act of 1999, republic act no. 8749.* Retrieved from https://www.fao.org/faolex/results/details/en/c/LEX-FAOC045271/ (Accessed: 2022-12-05)

Gandhi, R. (2018, Jul). *R-cnn, fast r-cnn, faster r-cnn, yolo - object detection algorithms.* Towards Data Science. Retrieved from https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

Google colaboratory. (n.d.). Google. Retrieved from https://research.google.com/colaboratory/faq.html

Huang, B.-J., Hsieh, J.-W., & Tsai, C.-M. (2017). Vehicle detection in hsuehshan tunnel using background subtraction and deep belief network. In N. T. Nguyen, S. Tojo, L. M. Nguyen, & B. Trawiński (Eds.), *Intelligent information and database systems* (pp. 217–226). Cham: Springer International Publishing.

Hui, J. (2022, Sep). *Real-time object detection with yolo, yolov2 and now yolov3.* Medium. Retrieved from https://jonathan-hui.medium.com/real-time-object-detection-with-yolo-yolov2-28b1b93e2088

Korstanje, J. (2021, Aug). *The f1 score.* Towards Data Science. Retrieved from https://towardsdatascience.com/the-f1-score-bec2bbc38aa6

Leung, K. (2022, Sep). *Micro, macro and weighted averages of f1* *score, clearly explained.* Towards Data Science. Retrieved from `https://towardsdatascience.com/micro-macro-weighted-averages-of-f1-score-clearly-explained-b603420b292f`

Li, K., Deng, R., Cheng, Y., Hu, R., & Shen, K. (2022). Research on vehicle detection and recognition based on infrared image and feature extraction. *Mobile Information Systems*, *2022*, 1–10. doi: 10.1155/2022/6154614

Lilly, C. (n.d.). *Co2 emission search by manufacture, make and model 2023.* Retrieved from `https://www.nextgreencar.com/emissions/make-model/` (Accessed: 2022-12-04)

Lu, J. L. (2022). Environmental pollution towards the workplace in the philippines. *Acta Medica Philippina*, *56*(1). doi: 10.47895/amp.v56i1.3889

Meng, C., Bao, H., & Ma, Y. (2020, sep). Vehicle detection: A review. *Journal of Physics: Conference Series*, *1634*(1), 012107. Retrieved from `https://dx.doi.org/10.1088/1742-6596/1634/1/012107` doi: 10.1088/1742-6596/1634/1/012107

myclimate Foundation. (n.d.). *Car co2 emissions calculator – carbon offset car.* Retrieved from `https://co2.myclimate.org/en/car_calculators/new` (Accessed: 2022-12-04)

Nath, R. K., & Deb, D. (2012, 09). Vehicle detection based on video for traffic surveillance on road. , *3*.

Overview - roboflow. (n.d.). Retrieved from `https://docs.roboflow.com/`

Platt, U., & Stutz, J. (2008). *Differential optical absorption spectroscopy (doas)—principles and applications* (Vol. 15). doi: 10.1007/978-3-540-75776-4

Pm2.5 footprint calculator-overview. (2021). Retrieved from `https://www.eg.mahidol.ac.th/dept/egce/pmfootprint/overview.php`

Powers, D. (2008, 01). Evaluation: From precision, recall and f-factor to roc, informedness, markedness and correlation. *Mach. Learn. Technol.*, *2*.

Rafique, M. A., Pedrycz, W., & Jeon, M. (2017). Vehicle license plate detection using region-based convolutional neural networks. *Soft Computing*, *22*(19), 6429–6440. doi: 10.1007/s00500-017-2696-2

Raj, J. T. (2019, Sep). *What to do when your classification dataset is imbalanced.* Towards Data Science. Retrieved from `https://towardsdatascience.com/what-to-do-when-your-classification-dataset-is-imbalanced-6af031b12a36`

Rito, J., Lopez, N., & Biona, J. (2021). Modeling traffic flow, energy use, and emissions using google maps and google street view: The case of edsa, philippines. *Sustainability*, *13*(12), 6682. doi: 10.3390/su13126682

Shah, D. (2022, Mar). *Mean average precision (map) explained: Everything you need to know.* Retrieved from `https://www.v7labs.com/blog/mean-average-precision`

Tantengco, O. A. G., & Guinto, R. R. (2022). Tackling air pollution in the philippines. *The Lancet Planetary Health*, *6*(4), e300. Retrieved from `https://www.sciencedirect.com/science/article/pii/S2542519622000651` doi: https://doi.org/10.1016/S2542-5196(22)00065-1

United Sates Environmental Protection Agency. (2022). *Particulate matter (pm) pollution- epa.* Retrieved from `https://www.epa.gov/pm-pollution/particulate-matter-pm-basics` (Accessed: 2023-01-06)

V-App - Vehicle Detection. (n.d.). *Ppe detection - vehicle detection.* Retrieved from `https://www.v-app.io/vehicle-detection/` (Accessed: 2022-12-04)

Vergel, K. N., & Yai, T. (2000, July 21). Analysis of road traffic flow and traffic environment in metro manila. In *The 8th annual conference of transportation science society of the philippines.* Retrieved from https://ncts.upd.edu.ph/tssp/wp-content/uploads/2018/08/Vergel00.pdf

Who-disability-adjusted life years (dalys). (n.d.). World Health Organization. Retrieved from https://www.who.int/data/gho/indicator-metadata-registry/imr-details/158

Yan, B., Fan, P., Lei, X., Liu, Z., & Yang, F. (2021). A real-time apple targets detection method for picking robot based on improved yolov5. *Remote Sensing*, *13*(9), 1619. doi: 10.3390/rs13091619

Yang, B., Tang, M., Chen, S., Wang, G., Tan, Y., & Li, B. (2020). A vehicle tracking algorithm combining detector and tracker. *EURASIP Journal on Image and Video Processing*, *2020*(1). doi: 10.1186/s13640-020-00505-7

Zheng, K., Zhao, S., Yang, Z., Xiong, X., Xiang, W., & et al. (2016). Design and implementation of lpwa-based air quality monitoring system. *IEEE Access*, *4*, 3238–3245. doi: 10.1109/access.2016.2582153

Zhou, F., Zhao, H., & Nie, Z. (2021). Safety helmet detection based on yolov5. *2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA)*. doi: 10.1109/icpeca51329.2021.9362711

Zoogman, P., Liu, X., Suleiman, R., Pennington, W., Flittner, D., Al-Saadi, J., . . . et al. (2017). Tropospheric emissions: Monitoring of pollution (tempo). *Journal of Quantitative Spectroscopy and Radiative Transfer*, *186*, 17–39. doi: 10.1016/j.jqsrt.2016.05.008

Appendix A

Appendix

A.1 Ha.Zee Command Line Help

Figure A.1 shows the command line help for the “VehicleDetect” program. This helps the user to determine what the different arguments do and what is needed to run the program. This can be accessed via the terminal by inputting the command “python VehicleDetect.py -–help” or “python VehicleDetect.py -h”

A.2 Log File

To record the data, the results were saved in a CSV file as shown in Figure N.2. The recorded data was written using the current frame that it was recorded in, not the average, hence why the values on the recorded data vary a lot between time periods.

```
(specProb) [jg@Asparagus SP_Hazy]$ python VehicleDetect.py --help
usage: VehicleDetect.py [-h] [--weights weights] [--conf conf] [--iou iou]
                        [--device device] [--time-to-avg time_to_avg]
                        [--log-timer log_timer] [--no-display]
                        filepath

Hazy A software for approximating PM2.5 emission from traffic footage.

positional arguments:
  filepath              Location of the video file

optional arguments:
  -h, --help            show this help message and exit
  --weights weights     path location of the weights that will be used
  --conf conf           Set confidence threshold
  --iou iou             Set IOU
  --device device       Set Device to use to CUDA or CPU
  --time-to-avg time_to_avg
                        Set the amount of time the program will average the
                        values
  --log-timer log_timer
                        time in seconds for the logger to write to the file
  --no-display          option to disable display
```

Figure A.1: Ha.Zee command line help shown

datetime	PM2.5	Cars	Jeepney	Motorcycle	Tricycle	Truck	Utility Vehicle
2023-06-02 07:38:32.224018	0.48	2	0	2	4	0	1
2023-06-02 07:38:40.234070	0.32	3	0	1	4	0	0
2023-06-02 07:38:46.144584	0.25	2	0	1	3	0	0
2023-06-02 07:38:52.065172	0.42	2	0	2	3	0	1
2023-06-02 07:38:58.003533	0.3	3	0	2	3	0	0
2023-06-02 07:39:03.801215	0.49	1	0	3	4	0	1
2023-06-02 07:39:09.732694	0.4	1	0	3	5	0	0
2023-06-02 07:39:15.656918	0.4	1	0	3	5	0	0
2023-06-02 07:39:21.589908	0.52	1	0	4	4	0	1
2023-06-02 07:39:27.492558	1.19	1	1	3	4	0	0
2023-06-02 07:39:33.409357	1.16	1	1	2	4	0	0
2023-06-02 07:39:39.196901	1.07	0	1	0	4	0	0
2023-06-02 07:39:45.052615	1.07	0	1	0	4	0	0
2023-06-02 07:39:50.892566	1.07	0	1	0	4	0	0
2023-06-02 07:39:56.817816	0.96	0	1	0	2	0	0
2023-06-02 07:40:03.250800	0.9	0	1	0	1	0	0

Figure A.2: Example Logfile where each entry was generated for approximately 5 seconds