POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

SMART TRAFFIC IMAGE PROCESSING SYSTEM (Smart TIPS): ADDRESSING THE TRAFFIC CONGESTIONS IN PARAÑAQUE CITY, METRO MANILA

A Thesis
Presented to the Faculty of the Parañaque City Campus
Polytechnic University of the Philippines
Parañaque City, Metro Manila

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

By

Ilagan, Sebastian Kyle L.
Morales, Vinci Louise
Ramos, Allen Miguel

September 2022

CERTIFICATION

This thesis, ***SMART TRAFFIC IMAGE PROCESSING SYSTEM (Smart TIPS): ADDRESSING THE TRAFFIC CONGESTIONS IN PARAÑAQUE CITY, METRO MANILA*,** prepared and submitted by SEBASTIAN KYLE L. ILAGAN, VINCI LOUISE MORALES, and ALLEN MIGUEL RAMOS in partial fulfillment of the requirements for the degree BACHELOR OF SCIENCE IN COMPUTER ENGINEERING has been examined and recommended for Oral Examination.

Evaluation Committee

ALDRIN P. OBSANGA
Adviser

MARVIN C. DE PEDRO
Member

CATHERINE R. LLAVE, MLing
Member

APPROVAL

Approved by the Panel on Oral Examination on July 16, 2022, with the grade of ____.

RIBERT R. ENIERGA, MIT
Chair

MARVIN C. DE PEDRO
Member

CATHERINE R. LLAVE, MLing
Member

Accepted in partial fulfilment of the requirements for the degree Bachelor of Science in Computer Engineering.

CLEOTILDE L. CRESCINI, DBA
Campus Director

ACKNOWLEDGMENTS

Serving as pioneers of the K-12 Program and who have dealt with various academic adjustments up to the occurrence of the COVID-19 pandemic which cut down more than half of the college experience, the researchers are very grateful that despite the challenges, it has been a long and great journey for them on this chapter of their lives.

From the bottom of their hearts, the researchers would like to extend their heartfelt gratitude to those people who helped throughout our research study:

First of all, to God who keeps on giving us positivity that we will finish our study. For enlightening us on the weakest days. The study was able to be done because of Him.

Second, is to their families who gave their never-ending support. For their consideration, for their time, for their advice, for financial assistance, and for their moral support in order to finish the research study.

Third, it is to the research teachers, especially the researchers' teachers who kept on being considerate every time. For giving some courage and guidance for the study, to Doc. Beata Maria F. De Ocampo, for guiding us at the first part of this journey. We would also like to thank Engr. King Ivan Earl Mauhay, for lending us the Raspberry Pi 3B which was primarily used in the development of the prototype.

Fourth, is to the classmates of the researchers, who helped them understand things. For their unending consideration and virtual answers. For the help, despite also doing a research paper to finish.

Fifth, is to the respondents who helped us to answer our surveys and who gave their time answering our surveys. And apologies for the inconvenience knowing that they have their class but still, they managed to contribute to the study.

And lastly, to the group who worked on this research study, who keeps on working hard, who keeps on being patient and enduring despite non-stop work. They keep on aiding our studies, despite our hectic schedules. For being optimistic and at their best, even if there were troubles ahead that would likely affect them.

CERTIFICATION OF ORIGINALITY

This is to certify that the research work presented in this thesis entitled *SMART TRAFFIC IMAGE PROCESSING SYSTEM (Smart TIPS): ADDRESSING THE TRAFFIC CONGESTIONS IN PARAÑAQUE CITY, METRO MANILA* for the degree Bachelor of Science in Computer Engineering at the Polytechnic University of the Philippines embodies the result of original and scholarly work carried by the undersigned. This thesis does not contain words or ideas taken from published sources or written works that have been accepted as basis for the award of a degree from any higher education institution, except where proper referencing and acknowledgement were made.

SEBASTIAN KYLE L. ILAGAN

VINCI LOUISE MORALES

ALLEN MIGUEL RAMOS
Researchers
July 16, 2022

ABSTRACT

Title : Smart Traffic Image Processing System (Smart TIPS): Addressing the Traffic Congestions in Parañaque City, Metro Manila

Researchers : Sebastian Kyle L. Ilagan
Vinci Louise Morales
Allen Miguel Ramos

Degree : Bachelor of Science in Computer Engineering

Institution : Polytechnic University of the Philippines

Year : 2022

Adviser : Engr. Marvin C. De Pedro

Traffic congestion has substantially impeded the nation's progress. We conducted this research in order to fully implement a Smart Traffic Image Processing System assisted by artificial intelligence that will lessen traffic congestion and benefit the following entities from the study's findings. The inconsistent standard road widths in densely populated areas, like Parañaque City, are a major contributing factor to this problem. This will show that even the Raspberry Pi can perform its function with a little help, and you may not spend a lot of money just to develop an image processing camera. The two pieces of hardware we can find on the market, such as the cheap but weak Raspberry Pi 3b and the powerful Laptop, are used in this experiment. It will open up new avenues for the advancement of microcomputers and the creation of smart systems. Despite the fact that this study's research largely focuses on traffic-related issues, the researchers believe that this methodology can be easily applied to other situations that would be advantageous to the local governing body. Other examples besides traffic management include intelligent parking, security, and manufacturing quality control. The general public would gain if the local government conducted feasibility studies on the aforementioned potential using its own IT and engineering departments.

Keywords: Polytechnic University of the Philippines, Bachelor of Science in Computer Engineering, Raspberry Pi, image processing, smart traffic, traffic management, Parañaque City.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

THE PROBLEM AND ITS SETTING

This chapter conveys the background of the study where it states the problem that the researchers would like to address in this study. It also articulates the frameworks, hypothesis, and other preliminary information relevant to the establishment of the research prior to its application in the actual world.

Introduction

Traffic congestions have had a significant negative impact on the country's development. One focal reason in this dilemma is the inconsistency of the standard road widths in high density areas, such as Metro Manila. It is believed that the extemporaneous construction of roadways are prominent, whereas driving cultures are not being examined, and maintenance of road networks are poor (Memon & Kumar Khiani, 2020). Moreover, the existence of vehicle accumulation throughout the years imputes traffic jams. In Metro Manila, there is a tremendous rise of motorcycle users with the rate of 320 percent in 10 years (Regidor, 2013). This is due to the mobility of the two-wheeled vehicle, that the motorists are preferring to utilize these more than the public transport. Another facet is being considered and is becoming the most affective – urbanization. In the Philippines, the urban population grew by 45 percent in 2014, and is expected to grow by 60 percent in 2050, which is causing the slow progress in public transportation and higher cases of traffic (Sheilah et al., 2017). All over the years, these factors have affected the country's economy. The traffic congestion is believed to consume 3.5 billion Philippine pesos per year (Sidel, 2020).

It does not only affect the country's development but is also a major influence on the negative impacts experienced by each traveler, and the environment. Commuters, mainly, have an exposure towards negative health and efficiency results. These include increase in blood pressure, poor sleep quality, fatigue, obesity, and increased chances of acquiring heart attack (Tenorio et al., 2019). Moreover, it can also cause emotional agitations, including stress, anxiety, frustration, and aggression. Not only that, but they are also vulnerable in acute exposure to traffic-related pollution and other nuisances, like noise and psychosocial stress (Sarnat et al., 2014). On the other hand, these traffic congestions are also one of the main factors in air pollution which has caused 4.2 million deaths in 2016. In fact, it was recorded in Europe that approximately 25 percent of greenhouse gases were produced by public transport in 2017 (Rossi et al., 2020). With the implication of rising concerns dealt by the traffic issues worldwide, various solutions have been implemented with the use of technology and innovation. Meng et al. (2021) have developed an image processing system to detect the density of the traffic. They have used 1080p camera in recording the four-lane road. Prior to starting their research, the researchers considered employing PIC microcontrollers, but after learning that the sensors can only detect at a short distance, the researchers have reconsidered their decisions. In addition, Choukekar & Bhosale (2018) have implemented the same device. Both researchers have utilized edge detection technique, such as zero crossing, Prewitt, LOG, Robert, Canny, and Sobel.

These projects were only implemented in the area of the image that the camera could reach from its point of view, resulting in erroneous and falsified conclusions for the traffic management decision-making process. Traffic does not only occur in the front of each intersection. Some of the causes of the congestion include: (1) an encounter between two vehicles; multiple encounters of vehicles leading to a higher traffic volume; (3) bottleneck congestion; (4) trigger-neck congestion, a situation where other sections of the intersection becomes affected due to bottleneck congestion; and (4) the inevitable density of vehicles during peak-hours (Koźlak & Wach, 2018). In addition, these projects are not dynamically adaptive to different critical situations, such as emergencies. It is one of the most severe ramifications of traffic gridlock. Delays in police action, as well as fire and rescue efforts, are among the others.

This project intends to address the traffic congestions frequently occurring in the major roads of Parañaque City, being a part of National Capital Region (NCR), a highly urbanized and populated region, where traffic mostly occurs. As a gateway city for tourism, it is critical to establish a strong first impression so that visitors can go to their intended destination without becoming stuck in traffic. This initiative aims to advance the research of Choukekar & Bhosale (2018) and Meng et al. (2021) by using numerous cameras that will communicate with one another in real time, allowing for a more adaptable and accurate interpretation of the processed images. This will apply to a variety of situations, including emergency vehicles, pedestrians walking, closed roads, and motorcades. Reducing traffic congestion through the use of technology might be a big step forward for the city in terms of living a modern lifestyle.

Theoretical Framework

This study is founded on the study of Density Based Smart Traffic Light Control System (Choukekar & Bhosale, 2018) and as can be widely seen, smart traffic systems are nothing new nor out of the ordinary. In fact, concepts like these have been long deployed for years now. For instance, Sydney Coordinated Adaptive Traffic System, or

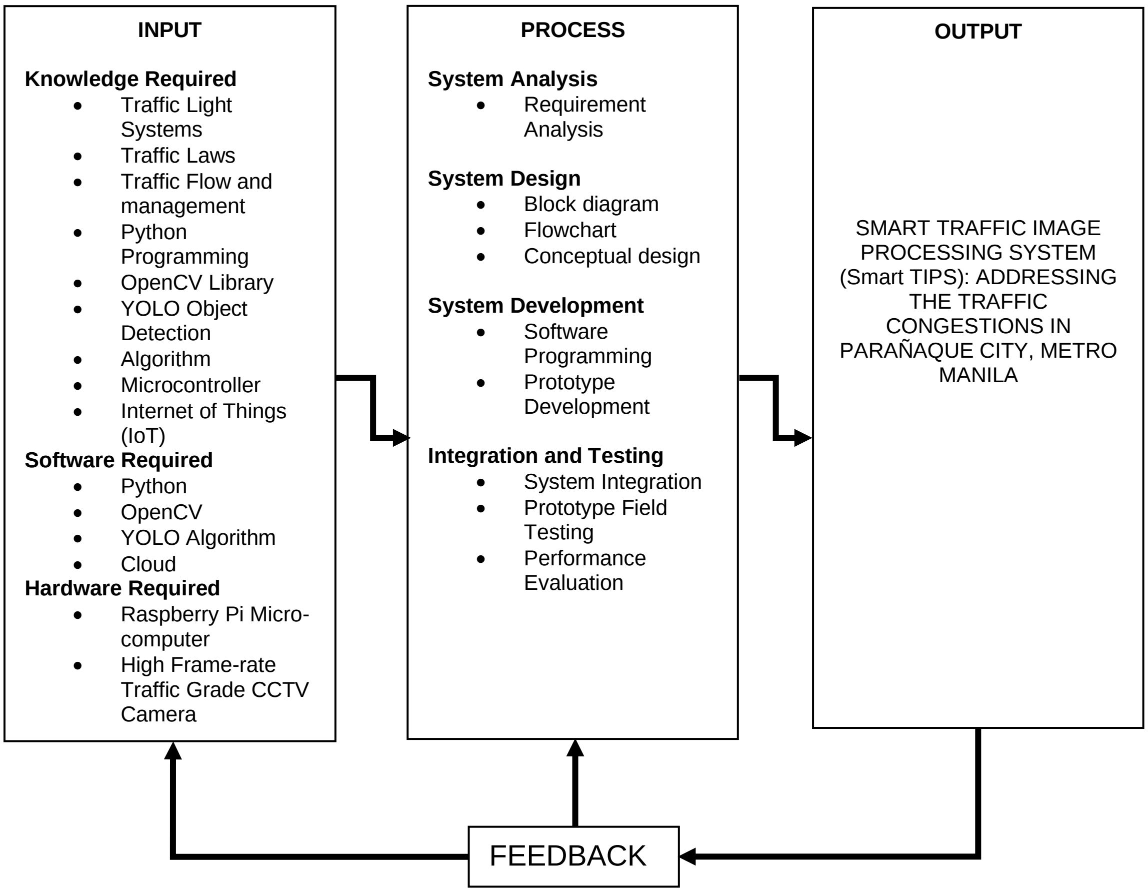
SCATS for short, is an intelligent traffic system established in Sydney, Australia. It has been multiple times over the years to keep up with the times, but its early versions have been deployed as early as the 1980s. Hence, it can be assumed that computer-based traffic system has already been prominent, especially in today's world. But just like with any piece of technology, this too must keep up with the innovations. Most smart traffic systems of the past decade relied on a single formula: sensors, such as induction coils and lidar for input and algorithm of some sort to process the data gathered from said sensors. Unfortunately, some of these sensors rely on certain conditions to function on their peak performance. Installation and repair are also a daunting task as some of these sensors need to be buried into the asphalt. Lastly, these are also subject to noise which can affect accuracy and performance. But thanks for advancement in sensor technology over the past decade most of the concerns with accuracy and reliability have been mostly addressed. Moreover, the use of multiple cameras are very important in processing the analysis in the traffic light system (Atta et al., 2020).

Over the past decade, big advancements came in Computer Vision technology which led to many studies regarding its prospective effectiveness in Smart Traffic Systems. Though many deployments have been done all around the world with each having their own reiterations on the technology, the technology is still in its infancy. People are still experimenting on how to properly use the technology to its fullest potential and solve its glaring issues. Atta et al. (2020), in Pakistan, has implemented a technology using radio frequency identifications (RFIDs) to track the road blockages and implemented multiple cameras, along with other sensors.

Conceptual Framework

In this research, the conceptual framework was built on the theory and concepts provided. The conceptual model in the figure below depicts how the system functions and how the technology for the research is related to one another.

Figure 1. Conceptual Framework Paradigm



Contained in the Input section is a list of the various fields and technologies required to conceptualize a prototype for this system: knowledge, software, and hardware. Knowledge requirements shall cover the provisions of proper traffic control, including the laws, systems, and management. Not only that, but also the other requisites needed for developing the device, including the platforms that will be used like Python, OpenCV, and YOLO object detection. Software requirements, include all the software that will be utilized in the development of the system, and as well for all the processes that will be conducted when the system has been deployed. You Only Look Once (YOLO) algorithm will help the image processing in providing a more accurate and comprehensive vehicle detection. Lastly, the Hardware requirement, High Frame-Rate Traffic Grade CCTV Cameras will be needed to capture real-time scenarios of the traffic. This particular type of camera was chosen as it is proven that having more frames will result into a more accurate object detection. Also, Raspberry Pi is needed to manage and send the data from the camera to the cloud.

Moreover, the Process section shows the outline of the development cycle of the prototype. The System Analysis will aid the researchers in studying the procedure on how the system will work and determine its goals moving forward. This also includes requirement analysis in which the researchers will focus to determine the needs and conditions required in order to build our system. Furthermore, System Design takes part in laying the groundwork for the actual device itself, creating Block Diagrams, Flowcharts on how the system will perform and drafting the conceptual design of the device. Moving on, the project has the System Development where development progress is made from both hardware and software with the end goal of producing a working prototype of the system. Lastly, Integration and testing involves the incorporation of the materials, such as camera, micro controller, modules, and lastly, the software. Thereafter, the researchers will conduct field testing of the finished prototype to evaluate its performance.

Finally for the Output section will show the end-product of the research which will be resulted from the work done as show from the first two sections. As can be seen in Figure 1, the researchers have arrows going from Output to Feedback, and Feedback to both Input and Process. This signifies that after the finished development of the system, the researchers will use the feedback received to refine both the project's development process and also revise the systems to resolve issues that may arise during operation.

Statement of the Problem

This research generally seeks to determine if the Raspberry Pi prototype can be used for the Smart Traffic Image Processing System (Smart TIPS) through the reference prototype that is a desktop-based computer system. Specifically, the researchers are interested in answering the following questions:

1. What is the current status of the image processing used in detecting traffic vehicles in Parañaque City?
2. What are the stages undertaken in the development of Smart Traffic Image Processing System (Smart TIPS) using Agile model?
3. What is the effectiveness of the system prototype in terms of:
    a. Accuracy;
    b. Recognition speed; and,
    c. Latency?
4. Is there a significant difference between the effectiveness of Raspberry Pi-based and desktop computer-based system prototype?
5. What are the issues and challenges encountered by the Raspberry Pi-based and desktop computer-based system prototype in using the proposed Smart Traffic Image Processing System (Smart TIPS)?
6. What is the training program that can be implemented to inform the general public regarding the proposed Smart Traffic Image Processing System (Smart TIPS)?

Hypothesis

The study's hypothesis is articulated, as indicated, to give a general notion of the expected relationship between the study's primary factors. Moreover, the study aims to prove that there is no significant difference between the prototype using the Raspberry Pi to the reference prototype using the desktop-based computer system.

Scope and Limitations of the Study

Smart Traffic Image Processing System (Smart TIPS) employs image processing to process video or image data fed to it. Providing information such as vehicle count and position in a specific frame, which can be utilized in a variety of applications, the most obvious of which is smart traffic. A possible application for it would be cameras installed at crossroads feeding video data to this image processing system to aid in the automatic estimation of traffic congestion on the road. As a result, the lengths of the green and red signals could be adjusted accordingly (Kavehvash, 2016). Understanding the flow and congestion of vehicular traffic is essential for an efficient road system. This research intends to utilize experimental research, while also implementing mixed-method research, particularly explanatory and exploratory research to provide a resolution to the research questions. This image processing system can be acquired with the help of Machine Learning via OpenCV, which is a library primarily aimed at real-time computer vision, and Python, a suitable programming language for this type of system. This image processing system will also be powered by the You Only Look Once (YOLO) algorithm, which is open source and looks at the entire image and uses a single convolutional layer to predict the bounding boxes and their confidence levels. A big focus of this project is also to build the final prototype for the least cost possible within reason. Therefore, the researchers' decisions in the hardware choices for prototype building will be heavily influenced by cost.

To start with, the Raspberry Pi 3b and computer-based devices like laptops are the two systems that the researchers have built. Due to its low price and 64-bit CPU, which is needed in image processing, the Raspberry Pi has become increasingly popular in recent years. Then comes a computer called Base Line, a laptop with a GPU and 16GB of RAM that is more powerful than the Raspberry Pi 3b. Our goal of using the Raspberry Pi 3b and laptop to detect an image for the Smart Traffic System was successfully accomplished. Both devices are capable of doing so. Since the Raspberry Pi 3b is the least capable of completing the task, it struggles, lowering its confidence level to 50% and recognizing fewer vehicles inside the bounding box per frame. However, a powerful laptop's ability to complete the work well and detect several cars can boost confidence levels to 90%.

Algorithms interpreting the processed data spat out by this image processing system to be used by applications such as smart traffic lights are beyond the scope of this research. Also, the data collected could be valuable for future transportation planning projects. This project is being pursued in order to reduce traffic congestion alongside other researchers focused on algorithms that will process the data spit out by this system in the Polytechnic University of the Philippines (PUP) - Parañaque Campus, specifically in Paranaque City. The prototype will be completed in 4 months, beginning in February 2022, and will end in May 2022, with the implementation stage lasting only 2 months, from June 2022 to July 2022.

Significance of the Study

This research will primarily benefit the common drivers within the premises of Parañaque City once the fully implemented smart traffic system has been produced in the future research. In the meantime, the following entities will benefit substantially from the outcomes of this project:

Government. The findings could be beneficial as a reference point and for future deployment of the prototype as a better complement to citizens' concerns regarding traffic management in Parañaque City. Parañaque City's Traffic and Parking Management Office the findings of the study will aid the city's traffic management in implementing this project to reduce traffic congestion and improve road safety.

Driving Schools. This research may assist them in becoming more informed about Smart Traffic innovation and incorporating it into their teachings.

Licensed Drivers. The study's findings will benefit Parañaque drivers by allowing them to use traffic management to travel more safely and conveniently.

Future Researchers. The findings of this study could serve as the foundation for future initiatives such as Smart Traffic and Road Congestion.

Definition of Terms

This section discusses the technical terms used in this research. The following terminologies are conceptually and operationally defined for better understanding of the readers.

Digital image. Digital image has been a prominent contribution in various applications such as medical, education, security, industrial, and domestic sectors (Wong, 2012). It is an end-product of devices that capture photographs like cameras. In this research, digital image pertains to the images being captured by the cameras attached to the traffic lights. These images will be processed and analyzed for the utilization of smart traffic light management.

Functionality suitability. Function suitability is a quality factor describes the extent to which a software product or system provides functions that satisfy the stated and implied needs of stakeholders when used under specified conditions. This quality factor has been divided into three (3) lower factors including functional completeness, functional correctness and functional appropriateness (Peters & Aggrey, 2020). In this research, since image processing has a wide range of applications, there must be a criterion for quality, such as functional appropriateness, to help evaluate this function.

Image processing. Image processing is a technique for using images collected from a device to create a specific process depending on the data input. It undergoes different methods and techniques, such as image enhancements, to improve the quality of each image to gain better and more accurate results (Mohan & Poobal, 2018). In this research, the same processes will be applied to the digital images and will be utilized to create an adaptive and preemptive traffic light system with prioritizations, such as the occurrence of emergency vehicles.

Maintainability. Maintainability is the ability of software products or systems to be modified, corrected, or adapted to current changes in the environment describes its maintainability feature. Five (5) lower factors including modularity, reusability, analyzability, modifiability, and testability were associated to maintainability (Peters & Aggrey, 2020). In this research, this quality factor makes it easier for future researchers to modify the image processing system.

Performance efficiency. Performance efficiency is a factor describes the ability of software product or system in managing the given number of resources to provide and maximize performance. This quality factor has also been decomposed into three (3) lower factors including time behaviour, resource utilization and capacity (Peters & Aggrey, 2020). In this research, to analyze resource allocation when providing essential services and functions in smart traffic, performance efficiency has been adopted into the image processing system quality model.

Reliability. Reliability is a factor expresses the capability of a system or software product to maintain its level of performance or specified functions under specified conditions for a specified time period. Four (4) lower factors are associated to reliability factor namely maturity, availability, fault tolerance and recoverability (Peters & Aggrey, 2020). In this research, this quality factor was incorporated into our image processing system's quality model to evaluate the dependability of various functions and services.

Security. Security is a factor is about how the software products or systems protect its information and data (information resources) from unauthorized persons or from other software products or systems. The security factor comes with set of lower factors which include confidentiality, integrity, non-repudiation, accountability, and authenticity (Peters & Aggrey, 2020). In this research, this is used to evaluate the operations of image processing systems in smart traffic.

Smart traffic. Smart traffic, conceptually, pertains to the application of a specific mechanism towards the traffic lights for it to become intelligent and adaptive to the traffic conditions (Yusuf et al., 2021). In this research, smart traffic is being referred to addressing the concerns in traffic congestion. However, the specific mechanism that will be utilized in this research is computer vision.

Usability. Usability is a factor describes the extent to which software or system product can be used to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use. The usability factor has set of lower factors which include appropriateness recognizability, learnability, operability, user error protection, user interface aesthetics and accessibility (Peters & Aggrey, 2020). In this research, to aid in the evaluation of image processing systems in smart traffic, supportability and detectability were introduced as lower factors under usability. the operations of image processing systems in smart traffic.

Chapter 2

REVIEW OF RELATED LITERATURE AND STUDIES

This chapter will cover five (5) significant topics: the currently existing smart traffic technologies from local to global areas, development of various smart traffic systems, the use of Raspberry Pi microcomputer as a tool for the development of a prototype in smart traffic, challenges that arise in using bare minimum hardware, and its implementation after its development. These subjects are critical for comprehending the fundamentals and usability of each component that the researchers will use to create a feasible and distinctive smart traffic light system.

Status of smart traffic technologies from a global and local standpoint

Different projects and technologies have been implemented to address traffic management. In India, Chavhan & Venkataram (2020) have introduced their approach using prediction analysis. It highlights the use of emergent intelligence (EI) and collected historical and present data, predicted traffic density, and travel time as input variables. Then, it was compared with existing technologies, such as k-nearest neighbor (KNN), artificial neural network (ANN), time series analysis (TSA), and support vector regression (SVR). It was discovered that using EI met the considerable level requirement while also reducing the time spent making accurate decisions and anticipating outputs.

On the other hand, an area traffic control (ATC) system was developed to manage the roadway issues in Phnom Penh. Matsuoka (2018) used a centralized system to manage traffic lights, as well as video vehicle detector cameras and improved traditional materials such as pavement markings, signal lanterns, and traffic signs. However, due to the ongoing growth in the number of vehicles on the road, the project was unable to maintain and manage traffic congestion in the city.

More technologies are being added to address the recurring issues with bottlenecks on the roads. A traffic management project in Spain adopted the use of vehicle-to-infrastructure (V2I) communications to resolve traffic circumstances and accidents with the help of wireless access in vehicular environments (WAVE) under the IEEE 802.11p communication standards. When the situation necessitates it, the traffic control station must manage all of the information flowing from the vehicles and return a warning signal Driving State and a recommended action to the driver (Milanés et al., 2012). Moreover, the use of fuzzy logic was furnished in determining the Driving State behavior of the driver.

In Pakistan, Atta et al. (2020) established a system that distinguishes road blockage using RFIDs. The initiative was developed to transform the static traffic signal behavior to become dynamic. Moreover, the researchers have used Internet-of-Things (IoT) technology, infrared (IR) sensors, and fuzzy device to overhaul traffic signal delays and timings. The project was able to deplete the traffic congestion to its minimal output.

On the local perspective, a similar project was conducted by the students at National University in Metro Manila. In this research, Carpio et al. (2018) has implemented an image processing technology on a closed-circuit television (CCTV) cameras. This was installed on a higher altitude in Earnshaw near their campus. As stated by Carpio et al. (2018), the image processing works by detecting the vehicles through a microprocessor. Only if the object surpasses the vehicle area will it be considered a vehicle. Anything with a smaller area will be assumed to be another object, such as a human or an animal. At the end of their research, they discovered that any camera, regardless of image quality, may be converted into a smart camera without incurring any additional costs. Furthermore, the team observed that the image processing technology can be used to a variety of different sectors, including medicine and artificial intelligence. The only limitation that the team has identified is that it must be maintained over the internet. Without it, real-time image upload and analysis may be jeopardized.

Lastly, as mentioned that image processing may be used not only in detecting traffic vehicles, Manlises et al. (2016) has implemented another image processing technology through a CCTV camera in Manila. This time, the project's goal was to detect pedestrians and advise them to be cautious when crossing the street. For the processes, the team used a webcam and a Raspberry Pi. The study was successfully implemented to detect pedestrians while also adjusting the number of timers based on the presence of any people crossing the street. The main limits observed by the researchers were proper lighting of the area in order to have a more precise detection, and pedestrian accessories such as umbrellas, eyewear, and headwear, which may also impair detection. Finally, it was discovered that the experiment was carried out during the day and without the presence of rain or fog, which may have resulted in a more appropriate outcome than what the researchers expected.

Development of different smart traffic systems

This section exhibits the usage of surveillance cameras and image processing system as an essential tool in traffic management. First, Kanungo et al. (2014) presented a traffic management scheme that uses video cameras installed at the four connected roads of an intersection since this requires little maintenance and is less likely to fail. The researchers have incorporated the camera data into a server to calculate vehicle density and apply an algorithm to manage traffic signals automatically. There are 30 frames being captured to monitor the density of the vehicles in real-time. However, the cameras become slightly ineffective during low light conditions, thus, two different modes were developed: hard-coded and dynamic coded. Hard-coded will only function during night-times; otherwise, dynamic coded will be implemented. The device was effective, reducing traffic congestion by four to five times.

Furthermore, it was discovered that a variety of systems, including motion detection, laser installations, and others, can be utilized to detect automobiles on the road. These solutions, on the other hand, were time-consuming, costly, and necessitated a large amount of hardware compatibility (Akoum, 2017). The use of image processing and video were used instead. In this research, Akoum (2017) utilized filter technique and acquired 90 percent of accuracy. To convert the image to binary, the researcher employed a foreground detector with Gaussian Mixture Models (GM-M) and applied filtering with image enhancements.

The use of fuzzy logic speed controller (FLSC) was also effective with image processing. In the project of Chabchoub et al. (2021), the FLSC was used to estimate traffic time based on the amount of automobiles on each road, regardless of the predetermined time. As a matter of fact, the FLSC was also integrated with multiple sensors and cameras to detect the vehicles within the “target zone,” which then, triggers the photo capture to process and decision-making will take place. To compute the number of vehicles, the photo was converted to black and white and white outlines were added to the objects. This device was able to adapt to the number of automobiles on each route in order to create a smart traffic light that prevented cars from waiting longer for a green light.

The use of multiple cameras in the aspect of minimizing the traffic congestion was highly supported by Kavehvash (2016). According to the study, using a single camera to make a more precise judgement at a traffic light is becoming limited and insufficient. As a result, the researcher used many cameras and applied the line-path triangulation approach to examine the outputs. However, due to the increasing difficulties of multiple image analysis, the researcher used an image fusion approach to assess a large number of vehicles from various perspectives using three-dimensional (3D) imaging techniques. Finally, it was discovered that these image fusion approaches improve the accuracy of a multi-camera traffic control system.

Another development of the smart traffic primarily utilizes the YOLO technology. It is an advancement that specializes image processing. Kuna (2020), in his research, has found that YOLOv3 algorithms is the most applicable and effective in deep learning of image processing and performing real-time object detection despite the obstructions in the scene. Along with this are tiny-YOLOv3 and faster Regional Convolutional Neural Networks (R-CNN). This is also supported by Khazukov et al. (2020) who shared that among the several neural networks (R-CNN, Faster R-CNN, RetinaNet, etc.), YOLOv3 achieved the best results by processing a single image in 51 milliseconds (ms) with a resolution of 608x608, resulting in 19 frames per second; as a result, it is capable of analyzing the highest quality of photographs without sacrificing much accuracy.

On the other hand, the YOLOv3 neural network has efficiently functioning in the ITS. In another study, the YOLOv3 algorithm was trained on a single-staged convolutional neural network (CNN) to recognize trucks in each camera image, and it worked well, with an average precision of 95 percent for an intersection over union (IOU) level of 0.75 (Hou et al., 2020). YOLOv3 is the most ideal algorithm in terms of computer vision. When it was attached to a camera, it retrieves data in real-time and interactive outputs (Redmon et al., 2016).

Raspberry Pi as a tool for building an efficient prototype in smart traffic

Proper specifications of technology play a vital role towards the development of a new device or system. With the growth of innovation, people's needs towards an efficient technology to help them alleviate their day-to-day experiences are also growing. The generation rapidly shifts its dependencies to the use of devices and giving green lights to digitalization.

One popular technology that has been currently highlighted in the market is the Raspberry Pi. It is a compact, powerful, inexpensive, programmable, and educational computer board that functions the same as a personal computer. Maksimović et al., (2014) shared some of the features of the Raspberry Pi which may be beneficial for a specific type of system or not. He expressed that the most primary impediment in its power is that no external device should consume more than 100mA from any of its USB ports. Not only that, but the Raspberry Pi has also limitations in its storage that only relies in secured digital (SD) cards and has only 256-512 megabytes (MB) of random access memory (RAM). Vujović & Maksimović (2014) agreed to the visible deterrent of the Raspberry Pi and added that its power consumption depends on the number of peripherals connected to it and particularly on how occupied the technology is. On a brighter note, Alex David et al. (2018) indicated the benefits or advantages of using the Raspberry Pi: it is cheaper than most high-end systems and it produces lesser heat than other systems in the market. Moreover, it can be a significant tool or technology for real-time image processing, which is a primary component of this research (Marot & Bourennane, 2017).

The use of Raspberry Pi has been prominent already in the implementation of various traffic management technologies. In Iran, Razavi et al. (2019) used image processing technology also to determine their smart traffic light scheduling. They made use of Raspberry Pi for a cloud-based storage for the video uploading. It was only used as a supporting attachment for the whole project where they can be able to make decision makings based on the density of traffic detected by their system. The same system was implemented by Jacob et al. (2018) where they utilized Raspberry Pi as a central unit for the controller which detects three (3) layers of density: low, medium, and high. Similarly, they also used this microcontroller as a medium to simultaneously upload footages through the cloud. Along with the Raspberry Pi, the researchers utilized some common materials in detecting traffic: ultrasonic sensors, camera modules, and simple light-emitting diodes (LED) to display the colors of traffic lights. Among all these studies, they would like to prioritize the detection of emergency vehicles and as well on its prioritization when it comes to traffic management.

Raspberry Pi is widely used in a variety of projects requiring a wide range of skills. This is one of the advantages of this microcontroller, and its capabilities are anticipated to equal those of a personal computer. It is also generally accessible to the majority of people due to its portability, low cost, and flexibility. However, it has significant drawbacks, which largely revolve on its power consumption, processing unit, and ability to process large amounts of data when compared to a high-end system. Nonetheless, different studies have been devised and executed, particularly in smart traffic, which uses Raspberry Pi as its central controller and offers various capabilities such as uploading footage to the cloud and serves as the primary controller based on traffic density.

Obstacles with using bare minimum hardware

Although image processing is a decade old technology, a lot of problems in the technology need to be ironed out when developing a system employing it. Especially when an individual is working with heavy budget constraints as are the researchers working on this study. Aside from the usual issues concerning visibility and weather conditions the plaque systems such as these (Kadam et al., 2019), a big source of challenges is the actual low-end hardware itself. In the prototype building process, a Raspberry Pi 3b was chosen as it wasn't too expensive and it fit the software requirement of having a 64-bit processor, and the researchers aimed to push the limits of how low-end of computers can be employed for a computational heavy system such as this. Before developing a prototype, a barebones image processing "software" was built in a computer as a proof of concept. It was originally thought by the researchers that said software can just be easily transferred to the Pi 3b. Unfortunately, due to the hardware limitations in the microprocessor in the Pi 3b this was not possible. As a result, a new image processing software was developed specifically for the Pi 3b using tools that supports its low-end processor and measly one gigabyte of LPDDR2 memory. Now that the researchers have the image processing software running in the Pi 3b, they encounter more challenges. Since the researchers have used heavily cut down versions of the original tools used in the computer version of the software, a huge drop in Latency, Accuracy, Max objects detected in one frame and Average FPS can be seen when comparing to the first image processing software developed. For example, a from 10fps average framerate in the computer version dropping to 2fps in the Pi 3b. Fortunately, the reasons for these issues can easily be traced. The accuracy and detection rate can be increased by some optimizations in the model and the program itself, but these fixes are miniscule and can be barely felt. Especially when compared the second fix which is a Tensor Processing Unit (TPU), a USB device which speeds up your machine learning algorithms on any Linux device, this includes the Raspberry Pi's. Regrettably, finding one in stock is a challenge in itself and it can cost up to three times higher than the actual Raspberry Pi 3b we are using. To add to that, a heavy focus of this research is to develop a system using low-end hardware and to be able to build the device using the least amount of money as possible. Thus, we will put the TPU aside for now while we explore for more solutions to the challenges we are facing.

A key feature of a road picture is that a major portion of it corresponds to the road, which can be assumed to be flat. One major issue is caused by natural occurrences such as fog and severe rain. It has an impact on the image processing quality and algorithm. These weather conditions are classified into two (2) types based on its visibility in the camera: steady weather and dynamic weather (Kadam et al., 2019). Fog, mist, and haze are examples of steady weather, and their particles are very small that they are difficult to discern once compressed in the same pixel. Dynamic weather consists of rain and snow and are a thousand times bigger than the particles of steady weather. (Kadam et al., 2019) also added that rainfalls are causing a major factor to decrease in the image quality, visibility, and causing a disturbance to the image processing algorithms.

Moreover, contrast enhancement is an indistinct problem when working with a fixed hazy image. Because of the physics of fog, visibility restoration necessitates estimating both the scene brightness and the scene complexity (Tarel et al., 2012). Another image processing approach includes the lane line detection. This technique focuses mostly on structured and non-structured highways, emphasizing reliability and real-time. Challenges arise, such as the influence of the surrounding environment and shadows cast by nearby trees or enterprises, causing poor image processing (Wang & Wang, 2018). It is also supported by Samuel et al. (2018) adding that impediments in road lines, surface changes, and differences in lane markings make the lane line detection technique difficult in image processing.

Effective techniques on the implementation of modern technologies

Technologies are incredibly helpful innovations to people around the world. It helps them in various categories: industrial, commercial, business, personal, and even miscellaneous. However, various steps must be taken for the general public to fully understand how the newly implemented device or technology works. This section discusses some of the existing projects and their means of implementing their outputs.

In New York City, Mydlarz et al. (2017) has implemented a low-cost acoustic monitoring devices. In their research, the team has set up a device where they can analyze and absorb the noise coming from the public and the vehicles within the New York City. They have found out that the foundation of any efficient cyber-physical system for noise monitoring is an accurate source of data from a dependable and low-cost sensor network. To put it simply, this project was implemented on their city government to distinguish noise patterns all throughout the urban areas. Decision-makers at local authorities might then strategically use the human resources at their disposal, for example, by effectively dispatching costly noise inspectors to troublesome places automatically detected by the proposed network. The researchers also found that their system may discover how sound affects the health of a city's inhabitants, how it correlates with urban problems ranging from crime to impaired educational settings, and how it affects real estate.

Similar to this project, a smart traffic system was also implemented in India. In this project, Rizwan et al. (2017) has attached a low-cost vehicle detecting sensor for every 500 to 1000 meters. This was made possible by the Internet-of-Things (IoT) in collecting real-time feeds of the cars' video footages. At the end of their study, they created a smartphone application to determine the density of traffic in various parts of their city. With the use of more modern technologies and sensors, the group hopes to increase its output to identify not only automobiles in general, but also specific vehicle types.

When a fundamental aspect of a publicly utilized system, such as a traffic system, is modified, the government or whoever is in charge must perform training and awareness programs for both workers and the general public so that they are not caught off guard when such a system is deployed. Aside from integrating them in the driver's license exams, they should also hold seminars to inform people about the nature of the system and how if functions. Furthermore, informative advertisement campaigns detailing the system functionality should be made to inform the last percentage of people who lack the time to attend said seminars. But thanks to the nature of automated systems such as these, public trainings would not be need, depending on the intrusiveness of the system to be deployed. For example, a smart traffic system would not require any public training as it is basically the same thing, it just has a function of dynamically changing the timer depending on the traffic congestion. Some personnel training may be required but little to nothing for the public as the traffic lights would remain the same in their eyes, just with some enhanced functionality. The reduction in training programs, seminars and ad campaigns would greatly reduce the deployment cost of the system, saving valuable taxpayer money.

Synthesis of the Reviewed Literature and Studies

The above-mentioned studies will aid in the development of a Smart Traffic Image Processing System (Smart TIPS) that employs many cameras and sensors.

The use of ATC system with improvement in the pavements, road signages, and traffic lights does not suffice the traffic congestion issue. In fact, as stated in their study, the ATC system was unable to improve the traffic jams in Phnom Penh due to the upsurge in road vehicles. On the other hand, being in the third-world country, the technologies provided in India, such as KNN, TSA, ANN, and SVR require high level technology which may not be able to sustain in the area field. Moreover, the technology shared by using V2I communications and fuzzy logic, may be too expensive and sophisticated for the project. Hence, this project will utilize image processing in determining the traffic congestions and analyze to realize the sustainable and reliable smart traffic light system. In fact, the use of surveillance cameras in providing a more dynamic traffic light system along with the use of RFID sensors to project the road blockages, IR sensors and the IoT technology. What is more preferable and realistic to adopt is vehicle detection by transforming conventional CCTV cameras positioned in roadways into smart cameras. These cameras require no additional fees and are simple to install. In particular, the function of image processing in several industries is becoming more important and effective, not only in transportation.

However, there may be some difficulties in the implementation of image processing. Challenges, like the presence of foggy and rainy weather, and the natural occurrences captured by the camera, such as the presence of multiple shadows of establishments and obstructions, may hinder the image processing technique. Generally, these factors affect the quality of image and influence its analysis: different weather conditions may cause presence of particles in the scene, while the shadows may change the quality in lighting; thus, contrast enhancement may develop an indistinct analysis of the image.

To address these image processing issues, several algorithms and techniques were developed. The Koschmieders model and histogram equalization and single image defogging method through DCP and FGS filtering were furnished to address the foggy particles in the photo. Among all the technologies, the YOLOv3 neural network and algorithms fits best in the Smart TIPS. Its rate in analysis (150 frames per second), and accuracy and image processing of a single photo in 51ms makes the analysis of traffic congestion most possible, accurate, and feasible. Moreover, the YOLOv3 algorithm was the best providing accurate analysis of image in object-detection regardless of the presence of any weather conditions.

The researchers wanted to examine the YOLO concept using low-cost hardware while maintaining picture processing quality. The implementation of this project shall contribute to the development of a more comprehensive and sustainable smart traffic management system in Parañaque City. To sum it up, image processing will be the best solution for ITS and smart traffic light systems in resolving the traffic congestion issues in the city. Along with the other valuable sensors, it may produce the most precise analysis and efficient decision-making in the traffic lights. Because a single camera cannot offer enough data to manage roadway gridlocks, the usage of several cameras and sensors shall be advocated. While, the use of motion detectors and lasers are discouraged, for these devices are time-consuming, expensive and requires a lot of maintenance.

When the system is released, the best method to notify the public is to collaborate with the government to undertake training and awareness programs for both workers and the general public, so that they are not taken by surprise when this system is introduced. Aside from incorporating them into driver's license assessments, they should also host seminars to educate people on the nature of the system and how it works. Furthermore, educational advertisement campaigns outlining system operation should be created to inform the remaining percentage of persons who do not have the time to attend those seminars.

Chapter 3

METHODOLOGY

This chapter probes the different methods that will be employed to establish the Smart Traffic Image Processing System (Smart TIPS). This will include various discussions on research design, research instrument that will be used in data gathering, and statistical treatment that will be used in the development of the project. Moreover, this will present various diagrams for further understanding of the study. These include block and schematic diagrams, and architectural design and prototype design.

Research Design

This research will employ applied research as the general research design. Applied research is the genuine investigation carried out in order to obtain new understanding (OECD, 2015). It entails applying current stock information and proper technique to a specific goal, which is primarily associated to the solution of a practical problem. This research design may be suitable for this research, which mainly tackles the effectivity of the use of modern technology towards addressing the pragmatic challenges in the traffic management of the roads in the city.

Moreover, this research will apply mixed method research to gather, analyze, and interpret the data provided by the evaluators. In the first half, the researchers will use exploratory research and may yield substantial information towards the development of a more comprehensive smart traffic management system. Exploratory research is qualitative in nature and is used to get basic insights into decision-making issues and opportunities (Shukla, 2011). This data collection technique focuses on gathering information from a small group of respondents by posing questions and watching behavior. Using exploratory research, the researchers will be able to acquire statuses of existing traffic management technology in the city through narrative analysis, which will be obtained through interviews and observation. Similarly, steps in the prototype's development will be established by observation and document analysis. It will be implemented to ensure the prototype's usability and efficacy based on user needs and the objectives of creating a long-term smart traffic management system. Finally, the researchers will be able to use and disseminate quantitative survey questionnaires, as well as conduct evaluative research to gain feedback from motorists and Parañaque traffic management personnel.

In the second half of this study, the researchers will use an explanatory research design, interpreting the acquired quantitative data to qualitative data. Exploratory research typically collects large amounts of unstructured data in order to investigate a new topic or respond to new concerns by breaking new ground by delving into new problem areas, working on topics about which little information is available, and gaining a broad understanding of a situation, community, or person (Strydom, 2013). After collecting the evaluations of the two primary variables, the differences in their evaluations will be studied using comparative research, which is a research approach that compares two (2) separate characteristics in order to construct or uncover a unique attribute or development on one or both of the materials (Esser & Vliegenthart, 2017). This will evaluate the variables listed on their evaluation to determine the efficacy of Smart TIPS, as well as the prototype's potential to contribute to traffic congestion alleviation. The researchers will be able to identify the prototype's issues and difficulties based on the analyzed data from their evaluations. This will be accomplished through the use of narrative analysis and evaluative research. Through the use of a literature review, the conclusions from this question will also be evaluated in light of similar concerns encountered in other smart traffic management systems in other countries. Finally, following the introduction of this system, training programs will be able to be developed to better aid and inform drivers and traffic management officials in order for this initiative to be used more extensively.

Process Flowchart

The researchers will explain the process flowchart that will be used in the preparation of the software development in this section. The researchers will adhere to the agile development methodology's framework. In addition, the group will use a Gantt chart to build a methodical procedure and timeframe for the construction of the prototype.

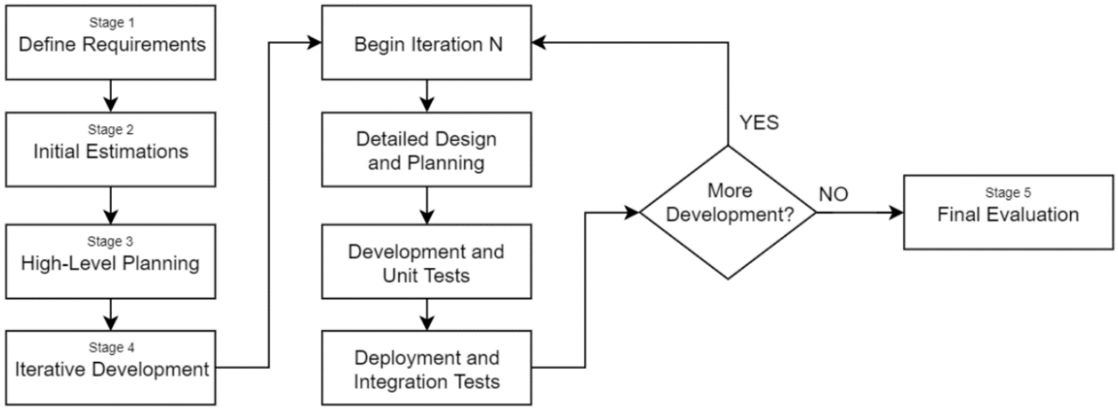
Figure 2. Gantt Chart of the Establishment of the Prototype

SYSTEM DEVELOPMENT	OCT 2021	NOV 2021	DEC 2021	JAN 2022	FEB 2022	MAR 2022	APR 2022	MAY 2022	JUN 2022	JUL 2022
Define Requirements	■									
Initial Estimations		■								
High-Level Planning			■	■						
Iterative Development					■	■	■	■		
Final Evaluation									■	■

Presented in the Figure 3 below is the flowchart for the Agile development model used in the production of the prototype. Particularly, this project utilizes Agile methodology of software development. It lightens the burden in working with the conventional methods, where some changes that are requested by the clients are difficult to implement since the project has already been established (Al-Saqqa et al., 2020). The major purpose of these methods is to speed up development and effectively respond to desired changes and to reduce the overhead in the software development process by allowing changes to be implemented without jeopardizing the process or requiring unnecessary rework. This strategy may be effective for the Smart TIPS because it will be in constant connection with the clients while contributing to the smart traffic management system in a quality and efficient manner.

This research will undergo into five (5) primary stages of agile methodology: (1) Define Requirements, (2) Initial Estimations, (3) High-Level Planning, (4) Iterative Development, and (5) Final Evaluation. During the requirements definition process, the researchers will collect data for the user, business, functional, and non-functional requirements. Following that, during the initial estimation stage, the researchers will carry out the process and data modeling to completely arrange the data flow. The next stage to be completed is high-level planning, which includes the identification of potential project limitations. This will entail managing the scope, scheduling, and costs. After all these requirements, the iterative development stage will begin. Before deploying to the client, a set of extensive tests must be performed to establish the integration procedures, development tests to see if there are any potential enhancements, and a comprehensive plan of execution. Finally, all preceding requirements should be met at the final stage. In the final evaluation, the users and client will assess the system and adopt it if no more changes are required.

Figure 3. Agile Development Model



Description of Research Instrument

The utilization of the research instruments in this study will provide a reference on how each problem is solved or fulfilled by the researchers. The first part of this research utilized descriptive and qualitative approach in determining the current status of the traffic management system used in the city of Parañaque, and as well as the system documentations vital for the assurance of the quality of the system. Moreover, the second part includes mainly on quantitative approaches through experimentation and testing.

1. **Current Status of Traffic Management System in Parañaque City**. To determine the current status of the traffic management system in Parañaque City, the researchers will set an interview meeting with the Traffic and Parking Management Office (TPMO) of the said city. A letter of intent is prepared declaring the objective of the researchers. It also includes a draft list of interview questions so that the authorities from the TPMO will have a prior orientation of the context. The list of questions is composed parallel to the objectives of this study: the current existing traffic management system used in the city, their development procedures on establishing the said system, the default settings applied to the system, potential issues and challenges encountered during the actual use of the system, and the training programs for the orientation of the new traffic enforcers.

2. **Development of the System Prototypes**. The system prototype is developed using the agile methodology. In preparation for this, the researchers use the construction of various requirements and documentations prior to the actual development of the system. Particularly, the researchers will establish four (4) requirements: Business

Requirements, User Requirements, Functional, and Non-Functional Requirements. During the initial estimations stage, the researchers will develop the Context Data Flow Diagram and YOLOv3 Architecture. Moreover, documentations such as actual screenshots of footages during the iterative development stage.

3. **Evaluation of Prototypes based on Various Parameters.** Both the prototypes – the desktop-based computer system and the Raspberry Pi, will undergo experimentation and testing to discover the different capabilities of each prototype on a certain parameter. In this research, these prototypes will be tested for 30 trials. The researchers will amass the data based on the (1) accuracy, (2) recognition speed, and (3) latency. The data will be examined and interpreted using the z-test through the use of Microsoft Excel Data Analysis tool. Afterwards, the data will be presented as follows in the Chapter 4:

Table 1

Sample Presentation of Results for Prototype Parameters

Trial No.	Parameter	Measurement	Remarks
1			
2			
. . .			
N			
Average			

4. **Difference between the Two Parameters.** Through the data gathered from the testing of efficiency of the prototype based on the parameters, the hypothesis will also be analyzed and interpreted using the Data Analysis tool of Microsoft Excel through the formula of z-test. The data on the testing of the hypothesis shall be presented as follows:

Table 2

Sample Table for Results of Hypothesis Testing

Parameters	z-score	Decision
Parameter 1		
Parameter 2		
Parameter 3		
Overall	**-**	**-**

5. **Issues and Challenges on the Prototypes.** To address this problem, the researchers will employ observation method during the multiple trials conducted to meet the previous objectives. The issues will be categorized into three (3) primary themes which will generally address the prototypes.

6. **Preparation of the Training Programs.** In preparation for the training programs that will be implemented once the system is deployed in the actual application, the researchers are completing the requirements accomplished from the earlier stages of the agile methodology. The presentation and interpretation of the training programs shall be indicated into three (3) primary clusters: System Documentation, Project Management Documentation, and User Documentation. The System Documentation shall include the documents from Functional and Non-Functional Requirements, and footages and materials from the iterative development. On the other hand, the Project Management Documentation includes the Business Requirements as well as the diagrams vital for the development of the prototype. Finally, the User Documentation presents the result of the User Acceptance Test (UAT) from the User Requirements document.

Statistical Treatment

Following the gathering of all data, the researchers will use statistical tools to evaluate and interpret it. The researchers will use statistics such as the Sample Mean, and Z-Test to the desktop-based computer system (reference) and the Raspberry Pi prototype, in order to organize the data and answer the study's problem.

To compute the central tendency of a data set, the researcher will utilize the Sample Mean formula. In terms of the respondent's evaluation.

$$\overline{X} = \frac{\boldsymbol{\Sigma f x}}{\boldsymbol{n}}$$

Sample Mean Formula

Where: $\overline{X}$ = sample mean

$\boldsymbol{f}$ = individual frequency

$\boldsymbol{x}$ = individual score

n = total frequency

The Z-test will be used to examine the significance of the difference between the desktop-based computer system (reference) and the Raspberry Pi prototype.

$$z = \frac{(\overline{x_1} - \overline{x_2}) - \mu}{\sqrt{\frac{{s_1}^2}{n_1} + \frac{{s_2}^2}{n_2}}}$$

Z-Test Hypothesis Testing for Two Samples Formula

Where: $\overline{x_1}$, $\overline{x_2}$ = means of samples 1 and 2, respectively

μ = population mean

s_1, s_2 = standard deviations of samples 1 and 2, respectively

n_1, n_2 = size of samples 1 and 2, respectively

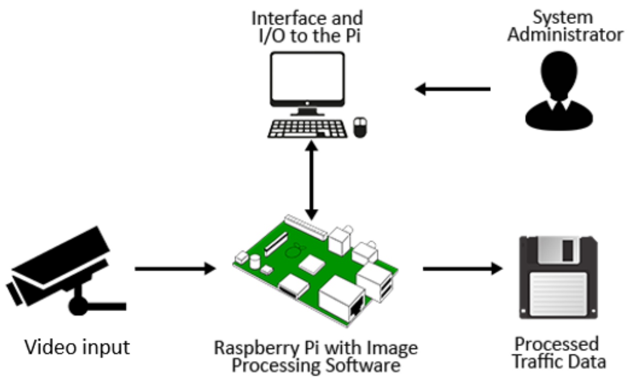
By considering the gathered values from the experimentation of the testing of the hypothesis and the parameters of the prototypes, the test will accumulate significant level of 0.5. It will also implement the two-tail critical value of 1.960. The sample mean shall be used to distinguish the mean of the trials done for the experimentation.

Design Process Flow

This section presents the different diagrams that will be essential in providing a substantial structure to help understand how the system will primarily work. These diagrams will be beneficial for the development of the hardware prototype and as well as it may serve as a primary reference for the components of the project.

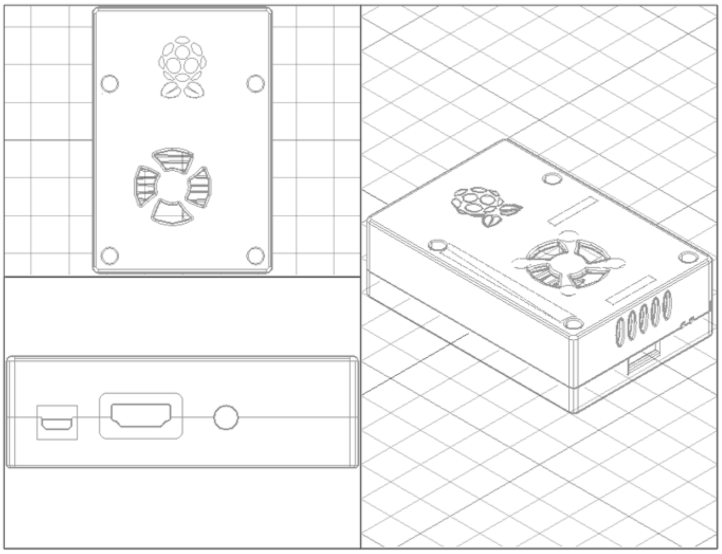
Architectural Design. The architectural design presents a simplified version of the process flow of the system. Architecture design is concerned with relationships and how aspects and components interact with one another. Presented in the Figure 4 below the flow on how the system administrator shall interact with the system, from its front processes, up to the systematic components of the system. This will be useful for the users to easily comprehend how will the prototype work.

Figure 4. Architectural Design of the Smart TIPS



Hardware Wireframe. The hardware wireframe displays the concept of the prototype once it is completely developed by the researchers. This helps the clients and the developers to have a prior visualization of the project while it is being processed. In this hardware wireframe, the prototype of Smart TIPS is presented in top, side, and isometric view, created through a software platform called AutoCAD.

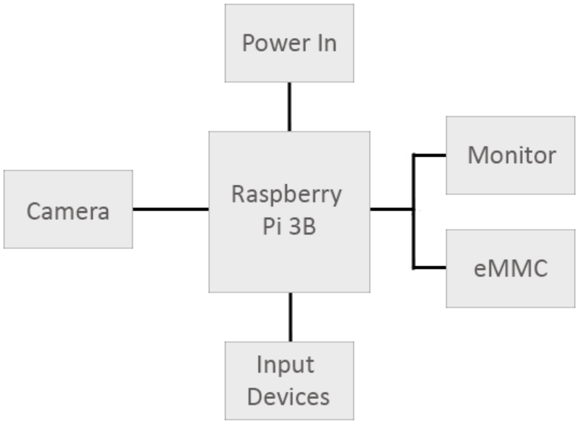
Figure 5. Hardware Wireframe of the Smart TIPS



Block Diagram

The system's block diagram depicts the system's main components as well as the processes that take place within it. Block diagrams enable us to comprehend the functioning of a system and aid in the creation of relationships within it. Primarily, it provides the functional visualization of the system.

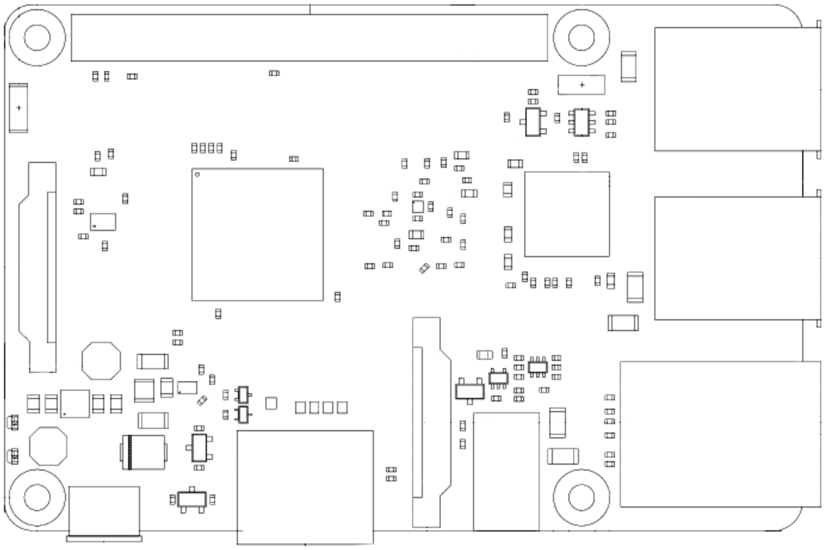
Figure 6. Block Diagram of the Smart TIPS



Schematic Diagram

In this research, the schematic diagram displays the two-dimensional circuit representation showing the functionality and connectivity between the electrical components of a Raspberry Pi 3B. Schematic diagrams are used to illustrate the high-level operation of a system or process. They simplify and promote communication by displaying and emphasizing the relations between system elements.

Figure 7. Schematic Diagram of the Smart TIPS



Chapter 4

RESULTS AND DISCUSSION

This chapter discusses the results gathered to meet the requirements in addressing the different problems and objectives of the Smart Traffic Image Processing System (Smart TIPS). The data were collected and presented by the researchers through various methods of qualitative and quantitative approach. This chapter contains an analysis, presentation, and interpretation of the study's findings.

1. Current status of traffic management in Parañaque City

The Traffic and Parking Management Office (TPMO) of Parañaque City has discussed that the city is using the conventional traffic light management. Each intersection is designated with a fixed timer with around 60 seconds for red light and 15 seconds for green light. Some of these traffic lights were replaced by an assigned traffic enforcer to assist in the flow of traffic or if the traffic light has any malfunctions. During heavy rainfall or foggy weather, these traffic lights were highly at risk to damages, hence, were chosen to be closed or turned off. The city is looking forward for new ways to address the traffic such as implementing plate number coding, truck bans, and other resolutions which will help ease the flow of traffic.

In terms of image processing, the city is implementing the No Contact Apprehension Policy as stated in Section 2 3 (a) of City Ordinance No. 17-06, Series of 2016, where traffic enforcement cameras will be put in various intersections and roadways throughout the City of Parañaque. These cameras are gadgets that automatically take high speed and high-resolution photos of vehicles that will beat a red light or reach a "yellow box" after a red light appears, be over the mandated speed limit, take wrong turns, counterflowing, enter or remain in a pedestrian lane near an intersection, or commit other violations of the Traffic Code. Speed cameras, on the other hand, are devices used to measure the speed of moving objects.

Parañaque City is not the only place where this kind of traffic management implementations exists. There is a similar traffic system implemented in the neighbor country. In Phnom Penh, Cambodia, Matsuoka (2018) also implemented a centralized control center to manage the traffic flow. It also used the conventional traffic management system with improvised traffic materials, such as pavement markings, signal lanterns, and traffic signs. This approach did not take a positive effect due to the ongoing growth in the number of vehicles. It was not able to maintain and manage the traffic congestion effectively. Whereas, in other countries, they are already trying to implement the use of modern technology to alleviate the traffic issues.

2. Development Stages of the Prototype for the Proposed Smart Traffic Image Processing System (Smart TIPS)

In the development of the prototype for the proposed Smart Traffic Image Processing System (Smart TIPS), the researchers followed the procedures and stages of agile methodology. During the start of the development, the researchers go through a cycle of planning, execution, and evaluation. The overall system development was distributed into five (5) different stages. These include definition of requirements, initial estimations, high-level planning, iterative development, and final evaluation.

2.1. Definition of Requirements. In this stage, the researchers employ four (4) system requirements documentation to ensure that the initial requirements of the system will be accomplished. It includes Business Requirements, Functional Requirements, User

Requirements, and Non-Functional Requirements. The Business Requirements are established by the researchers to determine the needs of the community that are needed to be addressed and resolved. In this study, the researchers include sustainable provisions including the acquisition of having a clear and understandable information, instruction, and notification regarding the system's usability. It also includes efficient operation with minor human interventions, and demonstrable aid to traffic operators rather than being an encumbrance to their operations. These requirements are all necessary for the system. The User Requirements, at the same time, was constructed to distinguish the users' needs. In this study, the researchers include three (3) primary users who will utilize the said system. These include the (1) traffic officers to use the system to observe the current traffic status through vehicle detection; (2) system administrators to monitor the system's functionality and status; and (3) future researchers to access real-time logs of the system and study how it works. The researchers applied these use cases to visualize how will the system be used once deployed. On the other hand, the researchers established the Functional Requirements to assist in the definition of a system's or one of its subsystems' performances. Functional requirements, in conjunction with requirement analysis, aid in the identification of missing necessities. They are constructed to identify the proposed system's service and behavior. This study features two (2) functional requirements: vehicle detection and tracking, and monitoring. To effectively implement these features, the system should have dependencies on camera, trained models, and vehicle detection model. Finally, Non-Functional Requirements are one of the most essential documentations needed prior to the development of the system. Non-functional requirements are important because they can seriously affect the success of a software system or product. Even if a system meets all its functional requirements but fails to produce the requisite quality results, users will not hesitate to reject it entirely. Hence, to attend to these regulations, the researchers initiated the following non-functional requirements: (1) the system should work with real-time feature having little to no delays on what the camera is detecting and the actual scenario; (2) it should also have a strong and reliable durability that can work for long periods of time; (3) it must be able to work by itself without requiring any internet or cellular connections other than its camera; and (4) it should be sustainable for future development and updates on the system.

2.2. Initial Estimations. In this stage, the researchers established diagrams to visually illustrate systems and processes that would be difficult to describe in text. These diagrams can be used to map out an existing system and improve it, or to plan out a new system for implementation. The system's Context Data Flow Diagram explains the process how the system will work properly. Any event that occurs within the camera's frame will be captured and sent to the Raspberry Pi as a live video feed. The vehicle detection software will then process this video input within the Pi. The video will be divided into frames and evaluated one by one. When the algorithm determines that an object fits the learned data, which would be vehicles, a bounding box will be put around it. The chopped images will then be patched together and returned to video form before being displayed on the monitor. The second diagram shows the breakdown of the YOLOv3 algorithm to present how it functions. The YOLOv3 architecture was divided into three (3) parts: the backbone, neck, and head. Deep architecture that was pre-trained on the ImageNet dataset sans top layers is typically used as the 'backbone'. These are typically standard CNN architectures like as Resnet, VGGNet, Mobilenet, and so on. YOLO, in particular, employs a modified version of backbone known as 'Darknet.' Meanwhile, the 'neck' is often made up of many layers with the objective of collecting feature maps from various stages. Finally, the 'head' is a component of the object detection model that is used for custom class prediction and drawing bounding boxes around objects.

2.3. High-Level Planning. The researchers, in this stage, has planned out the scheduling and essential distribution of task for the completion of the system. Using an Activity Chart with proper work-breakdown structure, the researchers are able to estimate the schedules and requirements in view of the agile methodology. In this table, all the stages of agile are included where their start and end dates are included. Moreover, the researchers developed a Gantt Chart to organize and track the progress of the system's development.

2.4. Iterative Development. This stage includes the actual execution of the development of the system. Presented here is the visualization of the components used in the system and how they are connected to each other. It also shows the actual testing of the system through a desktop-based computer system. In this iteration, a model car is used to represent as training model of actual vehicles in the roadways. This model car is trained to be detected in various angles including front, side, and top for it to become more efficient once it was deployed in actual traffic. It is also presented during this stage the Python codes used as training algorithms for the camera to recognize the vehicle detection. Finally, this stage features the trial-and-error activities made by the researchers to meet the necessary requirements and proper adjustments for the system.

2.5. Final Evaluation. In this stage, all the preparations and executions done by the researchers are evaluated and all the system experimentations are now done. This stage features results on the evaluation of two (2) different types of testing: System Testing and Evaluation Testing. In the System Testing, the researchers conducted experimentation on the parameters set for the completion of the system. In Raspberry Pi, the researchers tested its accuracy, recognition speed, and latency, which are also features of the Non-Functional Requirements. Through experimentation of 30 trials, the efficiencies of these parameters are gathered. The Raspberry Pi has an overall accuracy of 47.3%, which can recognize vehicles at 0.21 milliseconds and has a time delay of 4.97 milliseconds. On the other hand, the desktop-based computer system is tested in the same parameters as the Raspberry Pi. It acquires overall data of 90.75% accuracy rate, while having the capability to detect vehicles at 0.08 milliseconds, and it has also a shorter time of delay than the Raspberry Pi at 1.06 milliseconds. For the Evaluation Testing, the researchers have proposed for a User Acceptance Test (UAT) for its User Requirements. Upon the evaluation, the requirements of the traffic enforcer and the system administrator have passed the standards. On the other hand, the accessibility of real-time logs in the system is only approved for the PC version only as the Raspberry Pi cannot store operation logs that will be useful for the development of future technologies around this study.

The same software development model can be used to identify the possible requirements that will become more suitable for the needs of the community, and as well as the business for the system to be fully operational and sustainable once enforced in the actual application. It can be seen that the agile methodology is useful also in similar studies with multiple features, likewise in the study of Akoum (2017). In this study, the researchers applied a variety of modules such as motion detection, laser installations, and other modules applied to detect vehicles in the road. However, Akoum (2017) has realized that this is time-consuming, costly, and required a huge amount of hardware compatibility. Hence, the application of agile model is very efficient to avoid these kinds of challenges or alleviate those when encountered. Agile methodology becomes a flexible procedure to adapt to the needs of the stakeholders once an issue arises.

3. Effectiveness of the System Prototype in Terms of Accuracy, Recognition Speed, and Latency

The prototypes are tested on three (3) separate parameters that are critical for the use of image processing: accuracy, recognition speed, and latency. The data for these parameters were collected using 30 trials for each parameter during daytime.

3.1. Accuracy. This pertains to the program's confidence in determining whether or not the detected object is a vehicle. The accuracy of each trial was calculated by acquiring its average from its lowest and highest rate. Presented in the table, Raspberry Pi has its highest accuracy in its 26th trial with only 55.5%. On the other hand, the desktop-based computer system's minimum accuracy of 90.5% occurs frequently during the test. It can be observed that despite the frequent incidents of having its low accuracy, the desktop-based computer system was able to manage having a precise and consistent performance having a total average of 90.75% compared to the Raspberry Pi's average accuracy of 47.3%. In simple words, the desktop-based computer system tends to be more accurate than the Raspberry Pi.

3.2. Recognition speed. This refers to the system's capability to instantly detect the presence of a vehicle upon entering the system's designated field of view. During the test, it was shown that the Raspberry Pi's ($0.38\ \text{milliseconds}$) and desktop-based computer system's ($0.18\ \text{milliseconds}$) average recognition speed do not greatly differ from each other having only a difference of $0.20\ \text{milliseconds}$. Moreover, during the test, it is presented that the desktop-based computer system has a capability to detect vehicles within $0.08\ \text{milliseconds}$, a nearly instant vehicle detection. Whereas, in Raspberry Pi, the fastest time that the prototype can detect vehicles based on the trials is $0.21\ \text{milliseconds}$. Both prototypes have a close comparison, but the desktop-based computer system provides faster recognition speed.

3.3. Latency. The prototype's latency is articulated by calculating its real-time delay on the screen compared to the actual scenarios. In this research, latency is calculated through time measurement of seconds. Raspberry Pi can accumulate the longest time delay of $5.81 \text{ milliseconds}$ based on the trials. On the other hand, the desktop-based computer system has the longest delay of 2.01 seconds in its detection. Overall, Raspberry Pi has longer average time of delay (4.97 seconds) than the desktop-based computer system (1.06 seconds) which may cause difficulties and issues in providing a more reliable feature for the innovation.

The desktop-based computer system surmounts the capabilities of the Raspberry Pi in all three (3) parameters essential for image processing. It is also notable that in some areas, the Raspberry Pi experiences struggles, having a huge difference from the other prototype. One possible reason to this predicament can be understood in the specifications of the Raspberry Pi. Its power consumption depends on the number of components connected to it, but it only consumes around 100mA from any of its USB ports (Maksimović et al., 2014). This will also become a concern for the complex influence of the surroundings, such as cast shadows (Wang & Wang, 2018) and the obstructions in roadways and lane markings (Samuel et al., 2018), which may cause more difficulties in vehicle detection for the Raspberry Pi.

4. Difference Between the Effectiveness of Raspberry Pi and Desktop-Based Computer System

This study's null hypothesis states that there is no significant difference between the similarity in the effectiveness of Raspberry Pi and desktop-based computer system towards the application of the proposed system, however its alternative hypothesis states differently. The null hypothesis testing of the two prototypes was carried out with a significance level of 0.5 using the z-test formula.

This study also derived its mean ($\bar{x}$) and standard deviation ($\bar{s}$) to the averages of each parameter. Moreover, because the hypothesis evaluates the degree of correspondence or non-correspondence between the means of the two prototypes, the hypothesized mean difference is zero, and the two-tail test is used. The experimentation and hypothesis testing produced results indicating that in all of the parameters considered for the image processing, including accuracy, recognition speed, and latency, and the overall congruence in the null hypothesis should be rejected. Presenting that all z-values are produced and are in between the crucial limits of -1.96 and 1.96.

The correspondence between the Raspberry Pi and the desktop-based computer system in terms of accuracy ($z = -55.4355$), recognition speed ($z = 5.4693$), and latency ($z = 63.1929$) should be rejected. This indicates that the accuracy and latency of both prototypes differ from each other, but their recognition speed have slightly similar capabilities. Nevertheless, the overall congruity of both prototypes ($z = -51.5937$) presents their significant difference. Hence, the null hypothesis is rejected.

This should be noted that the Raspberry Pi prototype may acquire difficulties in running a substantial technology of YOLOv3, which will be primarily used in the image processing. YOLOv3 is an effective neural network other than R-CNN, Faster R-CNN, and RetinaNet, which has the capability to process a single image within 51 milliseconds having 19 frames per second (Khazukov et al., 2020).

5. Issues and Challenges Encountered in the Use of Prototype

When the researchers implemented the testing or experimentation to the two (2) prototypes, it has encountered some issues and challenges. Some of these may arise especially when the consistency of a parameter is being examined. In this research, there are three (3) challenges that researchers observed: hardware requirements, real-world performance, and efficiency of parameters.

4.1. Hardware Requirements. For this project, two prototypes were developed. A reference prototype running on laptop-grade hardware and a Raspberry Pi 3b+ prototype (Version 1) intended for real-world deployment. The goal is for the performance level of the Raspberry Pi Prototype to be as close as possible to the reference prototype. In pursuit of this goal, multiple tweaks were made to V1 to squeeze as much performance as possible from its processor, which are lowering the image resolution before it gets sent through the vehicle detection algorithm and lowering the confidence detection threshold to 35% in V1 compared to the 91% in the reference prototype. Unfortunately, even with these performance-enhancing modifications, the real-world performance of Version 1 is still too unsatisfactory for deployment testing. During the controlled tests for both the Reference and V1 prototype, a large discrepancy can be seen in the data gathered during the tests with V1 performing far worse than anticipated. This poor performance can be easily traced to the Raspberry Pi 3B+ itself, the Cortex-A53 in it, and its four (4) cores and threads with 1Gb or SDRAM are just not enough for the task at hand. This means we have to upgrade our hardware or use hardware accelerators to boost the performance of Pi.

4.2. Real-world Performance. As can be deduced from the data gathered during controlled testing, prototype Version 1 is far from primetime due to its multiple glaring performance issues. Firstly, the biggest problem the researchers have presently is the five-second delay between what is being seen on the monitor screen and what is in front of the camera. After a vehicle enters the camera's field of view, it will only reflect in the monitor after five (5) seconds have passed. And in terms of traffic 5 seconds can be a lot of time, which is why this long delay is completely unacceptable. This alone makes Version

1 completely unusable for real-world deployment as you would need little to no delay in the processing of the traffic data in order for it to be useful in applications such as smart traffic. The 5-second delay in Version 1 is no better than having no traffic enforcer at all and just letting the drivers plunge into chaos.

4.3. Efficiency in Parameters. Next up is the low detection accuracy and confidence due to its very little processing power. By no means is Version 1 bad when it comes to vehicle detection, but when put side by side with the reference prototype, it pales in comparison. On average, the reference prototype gets a 91% of confidence level when detecting vehicles and rarely loses its lock on cars. Effortlessly tracking ten cars at once. Meanwhile, Version 1 is averaging a 37% of confidence level and sometimes reaching its peak of 65%. Large inconsistencies in the confidence levels aside, the average and peaks are still a far cry from the reference prototype. Continuing the trend, our next issue is the limited concurrent detection of objects. While the reference can easily manage 10 at once and track them, Version 1 can only detect 3 vehicles per frame, and combining this with its low accuracy and confidence level results in bounding boxes jumping around when you have more than three (3) vehicles in the frame. This is obviously an extremely big flaw as there are more than 3 cars on any given road and having a limit defeats the purpose of why systems like this are developed in the first place. The last one is the low framerates, compared to the other issues this one is minor, but two (2) frames per second are too low, especially for smart traffic applications.

These issues are conventional and will mostly be equipped with Raspberry Pi. The device was very accessible due to its inexpensive factor and capabilities similar to a computer. However, when natural phenomena like heavy rainfall or foggy weather occur, the Raspberry Pi will easily experience a shortcoming. This is similar to the study of Kadam et al. (2019) when their device running through a Raspberry Pi has detected a potential hindrance, they have to develop a newer technology which will suit the microprocessor better. Moreover, Vujović & Maksimović (2014) has testified some notable Raspberry Pi issues and challenges: low capability due to issues in power consumption, has slow RAM with 256-512 MB only, and SD card dependency for storage and processing. These challenges will require a huge turnover to be highly reliable for the implementation of the project.

6. Training Program Developed for the Implementation of the Smart Traffic Image Processing System (Smart TIPS)

The development for its actual implementation should be supported by documentation focused on training and directing the system's intended users with all the knowledge they need regarding its functions when the proposed Smart TIPS has been approved and deployed to the potential clients. These training program documentations should be subjected to meticulous design, methodical execution, and stringent quality control. The Project Management Documentation, User Documentation, and System Documentation are its most critical components.

6.1. Project Management Documentation. This project intends to create smart traffic technology that will make use of processed images that can be more accurately and adaptably interpreted with the use of several cameras that will communicate with one another in real-time. The initial requirements, first estimates, high-level planning, iterative development, and final evaluation phases of this proposed system will all be looked at by the researchers. During the requirements formulation process, the researchers will collect data for the user, business, functional, and non-functional requirements. To properly manage the data flow, the researchers will then finish the process and data modeling at the initial estimating stage. The following step is to complete high-level planning, which entails determining probable project limits. This necessitates keeping an eye on the timeline, budget, and scope. Once all of these requirements have been satisfied, the iterative development cycle will begin. In order to design the integration processes, development tests must be done to look for potential improvements, and a comprehensive action plan must be created before deployment to the client. All preceding requirements must be met at the final stage. If no additional modifications are required, users and clients will examine the system in the final evaluation and accept it. Since it is an established system, there will be a recommendation for the following study that may be used to further this study and address any gaps that may exist.

6.2. User Documentation. Any behavior that is visible in the camera's field of vision will be recorded, and two developed systems—such as Raspberry Pi and computer-based systems like laptops—will use the camera as a source of input and output. They will receive a live video feed from the device. After that, the system will process this video feed before starting the program for car detection. When an object matches the learned data, which are images of automobiles, the algorithm breaks the movie into frames and analyses each one separately. A bounding box will then be drawn around the object. The films created from the images are then delivered to the monitor to be viewed. In the Object Tracker Logs, it also displays the Frame rate per second and the Frate number, which is the number of cars it detects in a single frame. Both of them are capable of performing their respective functions, but because of the low processor of the Raspberry Pi, Object Tracker Logs are not visible. The number and vehicle type are visible above the bounding box, allowing us to quickly ascertain the specifics of each vehicle. Our project can operate as a hardware-based system that requires hardware to detect the throughput as well as a video-aided system that can run inside the video feed that you input.

6.3. System Documentation. One of the main goals of the project, which is called "smart traffic system," is to develop a smart traffic system for the least powerful hardware possible. To that end, we have developed two systems: one that is based on the Computer Base Line, which is a powerful hardware, and another that is based on and runs on the Raspberry Pi 3, which is a very old and cheap Raspberry Pi that is now very popular. We were successful in achieving our objective of using the Raspberry Pi to detect traffic, but unfortunately, we ran into a few issues, including a slowness that causes a 5-second delay in real-time, making it necessary for us to wait until the feed appeared on the screen before moving the car. Another issue we ran into was a lower confidence level, as the Raspberry Pi only displays 30- to 50% confidence levels, which makes the system ineffective. We attempted to increase the system's detection level by 50%, but it only displayed the closest person on the camera. Another issue we ran into when using the Raspberry Pi was that the detective object has a limit of only being able to detect three objects in a frame, whereas in the real traffic world, a road can occupy more cars than just two. However, since we are still in the development phase, we should be able to scale up to slightly more powerful hardware to run the system. We currently can use a Tensor Processing Unit (TPU) in a USB device that, when connected to a Raspberry Pi, accelerates processing in the device. This improves performance by removing the 5-second delay and increasing frame rates to up to 20 frames per second. However, the TPU is expensive, so we cannot currently afford to purchase one. When it comes to the Computer Base Line, a laptop with a GPU and 16GB of RAM that is powered by an Intel i5 processor, the detection rate and confidence level are not a problem because it can be able to detect many cars in a frame that can increase the confidence level up to 90%. However, it is not recommended to use this because in the field we cannot regularly supervise the area of traffic since we are aiming for the Smart Traffic System, but it is a good comparison between the two developed systems, and as the population grows, the specifications become more important.

Similarly, the TPMO of the city of Parañaque has initiated training programs for their enforcers and employees on how they are going to utilize a certain device for their work. These include certifications, seminars, and actual hands-on implementation of the new regulation or device. They are also implementing an intensive training for newcomers or trainees, and as well, conducting regular courtesy calls to adapt to the dynamic situation of traffic in the metro. Moreover, with the proper training and familiarization with the technology, Mydlarz et al. (2017) development of acoustic monitoring devices to address the traffic. In their research, the team has created a device that can assess and absorb noise from the public and cars in New York City. They discovered that the foundation of any effective cyber-physical noise monitoring system is an accurate source of data from a dependable and low-cost sensor network. Simply expressed, this research was performed by their local government in order to discern noise patterns throughout urban areas. Technologies with complex modules or features require a long and comprehensive familiarization, not only to the personnel, but also to the general public for them to further understand its benefits and importance in the community.

Chapter 5

SUMMARY OF FINDINGS, CONCLUSIONS, AND RECOMMENDATIONS

This chapter discloses all the summary of the results gathered in the previous chapter, as well as the interpretations of the researchers. It also includes the conclusions based on the experimentations and tests conducted to determine the quality of the developed system. Finally, this chapter provides the outlet for the researchers to share the recommendations to further innovate the system for future uses.

Summary of Findings

This study was conducted to provide insightful and valuable information to contribute to determining the feasibility of the proposed Smart Traffic Image Processing System (Smart TIPS). Through the use of mixed method research having both the quantitative and qualitative procedures, the results are gathered and summarized as follows:

1. **Current Status of Traffic Management System in Parañaque City.** The current traffic management implementations used by the Traffic and Parking Management Office (TPMO) is the conventional traffic light management. Intersections are placed with traffic lights with fixed timers for red and green light. Traffic enforcers are also deployed to assist faulty traffic lights to direct the flow of traffic. During instances of rainy and foggy weather, these traffic lights are turned off. On the other hand, in terms of image processing, the city is utilizing the No Contact Apprehension Policy under the City Ordinance No. 17-06 which

include the use of cameras to detect violations of the motorists that capture their speed, placements on restricted areas, and other violations of the Traffic Code.

2. **Stages Undertaken for the Development of the Proposed System.** This study considers the development of the prototype design which will replicate and represent the actual system. The development undergoes through the agile methodology stages. These include the following: (1) definition of requirements, (2) initial estimations, (3) high-level planning, (4) iterative development, and (5) final evaluation. The completion of all the requirements is finished mainly at the first half of the research period. This is in accomplishment of the first three (3) stages of the agile methodology. On the other hand, the second half of the research period is devoted to experimentations of the system parameters, development of prototype, and as well, the evaluations to further assess its quality. All the gathered results and information will be used in constructing essential documentations for the training programs and as a reference for future research.

3. **Effectiveness of the Prototypes based on the System's Parameters.** The researchers conducted an experimentation to determine the effectiveness of the two (2) prototypes, using the Raspberry Pi and the desktop-based computer system. The prototypes are tested within 30 trials based on the system's accuracy, recognition speed, and latency. All the data are assessed by implementing the z-test formula through the Data Analysis tool of the Microsoft Excel. The results show that in terms of accuracy, the Raspberry Pi is only 47.5% accurate, compared to the desktop-based computer system with 90.75% accuracy. In terms of recognition speed, the Raspberry Pi can provide fast vehicle recognition with 0.21 milliseconds. Finally, the Raspberry Pi accumulates time delay of 4.97 milliseconds.

4. **Difference between the Effectiveness of the Two Prototypes.** The study's null hypothesis indicates that there is no significant difference between the effectiveness of the Raspberry Pi and the desktop-based computer system. The testing of hypothesis of the two (2) prototypes is undertaken using the z-test formula on a significant level of 0.5. The results show that the z-score of -51.5937 is distant from the acceptance region of -1.96 to 1.96. Hence, it shows that the effectiveness of both prototypes has a significant difference, and the null hypothesis is rejected.

5. **Issues and Challenges Encountered in the Use of the Prototype.** Despite the adjustments provided to the Raspberry Pi to match the quality of the desktop-based computer system, the real-world performance of this Raspberry Pi is still too unsatisfactory for deployment testing. The Raspberry Pi and the desktop-based computer system show a huge discrepancy having the Pi at its worse than anticipated performance. Moreover, due to the five-second delay of the Raspberry Pi to process and update detected vehicles in the monitor, the prototype becomes unusable for the real-world implementation. Aside from these issues, the more evident challenge that the researchers encountered is the Raspberry Pi prototype's parameters are distant from the reference prototype of the desktop-based computer system.

6. **Training Program for the Implementation of the Proposed System.** Through the scrutiny of the system requirements and the results of the experimentations, documentations for the system's training programs are established. The training program includes documentations from the Project Management, System, and Users. It will be intended to be used for possible operators or users of the system prototype and for future references. A copy of the prototype's training program is attached in the Appendices.

Conclusions

The researchers have gathered sufficient data for the system. With the presented results, the researchers have formulated the following conclusions:

1. **Adjustments on Raspberry Pi for the Image Processing.** The primary goal of this research is to distinguish the capabilities of the prototype, through the Raspberry Pi, to match the capacities of the desktop-based computer system in terms of the implementation of the image processing. Due to the limitations in budget, the researchers only maximized the features offered by the microcomputer. Since the image processing requires stronger processor, a 90% stringent confidence detection threshold on accuracy is not implemented for the Raspberry Pi to have a smoother flow of administration in its processor, thus, leading the Raspberry Pi to acquire approximately 47% of accuracy. The microcomputer is adjusted to have the threshold at 35% only, and the image quality resolution is downgraded. Otherwise, the Raspberry Pi will not be able to perform the supposed designated tasks applied to it.

2. **Dealing with Real-world Applications.** One of the major concerns in this research is the ability of the prototype using the Raspberry Pi to be applied in the real-world scenario. The researchers discovered through experimentation that the microcomputer acquires a five-second delay in the overall computer vision system. When a car reaches the camera's range of view, it will only be visible in the monitor after five (5) seconds. This is not applicable to the actual traffic situation as it may provide more accidents rather than contribute to alleviating the traffic. Partnering with hardware accelerators such as Tensor Processing Units (TPU) would definitely yield better results and eliminate the said problems.

3. **Potential in the New Version of Microcomputer.** Since the researchers are using the Raspberry Pi 3b+, it can be seen that the specification of this microcomputer is evidently low. This also proves the reason that the researchers have to create adjustments on the Raspberry Pi, and the parameters are unsatisfactory. Hence, it can be concluded that the use of the newer version of Raspberry Pi is projected to be more efficient when applied to this system. However, this new version of microcomputer is expensive but can be used by future researchers.

Recommendations

Based on this study's findings and conclusions, the proponent, therefore, has found the following recommendations considerable:

Academic Institutions, Faculty, and Students. As reflected by the gathered data from the testing of both prototypes, the resource-intensive task that is real-time image processing can be performed in a small Raspberry Pi albeit with some caveats. Again referring to the data gathered, the prototype intended for deployment (V1) did not perform as best as we could have expected. But it did prove that running image processing software on microcomputers such as the Pi for applications such as traffic management is feasible. With that in mind, the present researchers humbly ask and encourage students to carry on with this research and continue with iterations both with hardware and software, building upon what's already been laid out by our research. To immediately solve the current proposed systems glaring issues, the researchers advise moving up to a newer Raspberry Pi such as the Pi 4 and the addition of a Tensor Processing Unit (TPU). Minor adjustments to the software may also be required for it to work smoothly along the suggested upgrades.

As this proposed system is intended to serve as a more economical alternative to the more expensive traditional image processing systems, it would be in everyone's best interest for the academic institution to conduct an economic feasibility study as the result of this study would highlight the economic issues and benefits that this project will entail which would in turn help attract institutions to invest in this project.

Presently, the developed prototype is still in its early development stages and is quite far from what it will be when it has been deployed. Hence, the input of faculty members from engineering and other related fields is greatly appreciated. By virtue of their words backed by years of experience, flaws in the prototype's design can be caught and resolved in the early stages of the development. Further improvements to the system's architecture and design can also be made.

Local Government and Future Researchers. The prototype developed was designed specifically to only perform image processing as a standalone module, universal integration with other systems such as traffic lights systems was part of the plan but was not explored. Therefore, the present researchers recommend to future researchers for this area be given attention and resources.

Though the image processing performed in this research mainly concerns traffic-related means, the researchers believe that this can easily be applied to other fields that would be of use to the local governing body. Some examples aside from traffic management would be security, smart parking systems, and quality assurance for manufacturing. With these prospects, it would be advantageous for the masses for the local government to pour resources from its own IT and Engineering departments and perform feasibility studies regarding said prospects.

References

Akoum, A. H. (2017). Automatic Traffic Using Image Processing. *Journal of Software Engineering and Applications*, *10*(09), 765–776. https://doi.org/10.4236/jsea.2017.109042

Alex David, S., Ravikumar, S., & Rizwana Parveen, A. (2018). Raspberry Pi in computer science and engineering education. *Lecture Notes in Electrical Engineering*, *492*, 11–16. https://doi.org/10.1007/978-981-10-8575-8_2

Atta, A., Abbas, S., Khan, M. A., Ahmed, G., & Farooq, U. (2020). An adaptive approach: Smart traffic congestion control system. *Journal of King Saud University - Computer and Information Sciences*, *32*(9), 1012–1019. https://doi.org/10.1016/j.jksuci.2018.10.011

Carpio, J. N., Ranada, J. V., Bagara, M. G., Guadalupe, J. J. R., Landrito, C. L. O., Hilado, A. M. J., Ilano, K. M. L., Farala, J. D. R., & Doringo, M. R. C. (2018). Traffic congestion and speed assessment using image processing technology accessible via internet through smart devices. *International Journal of Simulation: Systems, Science and Technology*, *19*(3), 9.1-9.6. https://doi.org/10.5013/IJSSST.a.19.03.09

Chabchoub, A., Hamouda, A., Al-Ahmadi, S., & Cherif, A. (2021). Intelligent Traffic Light Controller using Fuzzy Logic and Image Processing. *International Journal of Advanced Computer Science and Applications*, *12*(4), 396–399. https://doi.org/10.14569/IJACSA.2021.0120450

Chavhan, S., & Venkataram, P. (2020). Prediction based traffic management in a metropolitan area. *Journal of Traffic and Transportation Engineering (English Edition)*, *7*(4), 447–466. https://doi.org/10.1016/j.jtte.2018.05.003

Choukekar, G. R., & Bhosale, A. G. (2018). Density Based Smart Traffic Light Control System and Emergency Vehicle Detection Based On Image Processing. *International Research Journal of Engineering and Technology (IRJET)*, *05*(04), 2441–2446.

Esser, F., & Vliegenthart, R. (2017). Comparative Research Methods. *The International Encyclopedia of Communication Research Methods*, 1–22. https://doi.org/10.1002/9781118901731.iecrm0035

Hou, R., Jeong, S., Lynch, J. P., & Law, K. H. (2020). Cyber-physical system architecture for automating the mapping of truck loads to bridge behavior using computer vision in connected highway corridors. *Transportation Research Part C: Emerging Technologies*, *111*(July 2019), 547–571. https://doi.org/10.1016/j.trc.2019.11.024

Jacob, S. M., Rekh, S., Manoj, G., & John Paul, J. (2018). Smart traffic management system with real time analysis. *International Journal of Engineering and Technology(UAE)*, *7*(3.29 Special Issue 29), 348–351.

Kadam, D., R. Madane, A., Kutty, K., & S.V, B. (2019). Rain Streaks Elimination Using Image Processing Algorithms. *Signal & Image Processing : An International Journal*, *10*(03), 21–32. https://doi.org/10.5121/sipij.2019.10303

Kanungo, A., Sharma, A., & Singla, C. (2014). Smart traffic lights switching and traffic density calculation using video processing. *2014 Recent Advances in Engineering and Computational Sciences, RAECS 2014*, 6–8. https://doi.org/10.1109/RAECS.2014.6799542

Kavehvash, Z. (2016). Improved multi-camera smart traffic surveillance system for resilient cities. *Scientia Iranica*, *23*(4), 1641–1647. https://doi.org/10.24200/sci.2016.2235

Khazukov, K., Shepelev, V., Karpeta, T., Shabiev, S., Slobodin, I., Charbadze, I., & Alferova, I. (2020). Real-time monitoring of traffic parameters. *Journal of Big Data*, *7*(1). https://doi.org/10.1186/s40537-020-00358-x

Koźlak, A., & Wach, D. (2018). Causes of traffic congestion in urban areas. Case of Poland. *SHS Web of Conferences*, *57*, 01019. https://doi.org/10.1051/shsconf/20185701019

Kuna, S. L. (2020). Real Time Object Detection and Tracking using Deep Learning. *International Journal for Research in Applied Science and Engineering Technology*, *8*(6), 58–64. https://doi.org/10.22214/ijraset.2020.6010

Maksimović, M., Vujović, V., Davidović, N., Milošević, V., & Perišić, B. (2014). Raspberry Pi as Internet of Things hardware : Performances and Constraints. *Design Issues*, *3*(JUNE), 8.

Manlises, C. O., Martinez, J. M., Belenzo, J. L., Perez, C. K., & Postrero, M. K. T. A. (2016). Real-time integrated CCTV using face and pedestrian detection image processing algorithm for automatic traffic light transitions. *8th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management, HNICEM 2015*, *December*, 8–11. https://doi.org/10.1109/HNICEM.2015.7393205

Marot, J., & Bourennane, S. (2017). Raspberry Pi for image processing education. *25th European Signal Processing Conference, EUSIPCO 2017*, *2017-Janua*, 2364–2368. https://doi.org/10.23919/EUSIPCO.2017.8081633

Matsuoka, S. (2018). Traffic management project in Phnom Penh. *IATSS Research*, *42*(4), 180–189. https://doi.org/10.1016/j.iatssr.2018.12.001

Memon, R. M., & Kumar Khiani, R. (2020). Traffic Congestion Issues, Perceptions, Experience and Satisfaction of Car Drivers/Owners on Urban Roads. *Mehran*

University Research Journal of Engineering and Technology, *39*(3), 489–505. https://doi.org/10.22581/muet1982.2003.04

Meng, B. C. C., Damanhuri, N. S., & Othman, N. A. (2021). Smart traffic light control system using image processing. *IOP Conference Series: Materials Science and Engineering*, *1088*(1), 012021. https://doi.org/10.1088/1757-899x/1088/1/012021

Milanés, V., Villagrá, J., Godoy, J., Simó, J., Pérez, J., & Onieva, E. (2012). An intelligent V2I-based traffic management system. *IEEE Transactions on Intelligent Transportation Systems*, *13*(1), 49–58. https://doi.org/10.1109/TITS.2011.2178839

Mohan, A., & Poobal, S. (2018). Crack detection using image processing: A critical review and analysis. *Alexandria Engineering Journal*, *57*(2), 787–798. https://doi.org/10.1016/j.aej.2017.01.020

Mydlarz, C., Salamon, J., & Bello, J. P. (2017). The implementation of low-cost urban acoustic monitoring devices. *Applied Acoustics*, *117*(August 2015), 207–218. https://doi.org/10.1016/j.apacoust.2016.06.010

OECD. (2015). *Frascati Manual 2015 The Measurement of Scientific, Technological and Innovation Activities*. https://www.oecd.org/sti/inno/frascati-manual.htm, visitado em 16/08/2021

Peters, E., & Aggrey, G. K. (2020). An ISO 25010 based quality model for ERP systems. *Advances in Science, Technology and Engineering Systems*, *5*(2), 578–583. https://doi.org/10.25046/aj050272

Razavi, M., Hamidkhani, M., & Sadeghi, R. (2019). Smart Traffic Light Scheduling in Smart City Using Image and Video Processing. *Proceedings of 3rd International Conference on Internet of Things and Applications, IoT 2019*, 1–4. https://doi.org/10.1109/IICITA.2019.8808836

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 779–788. https://doi.org/10.1109/CVPR.2016.91

Regidor, J. R. F. (2013). Traffic Congestion in Metro Manila : Is the Uvvrp Still Effective ? *Philippine Engineering Journal*, *34*(1), 66–75.

Rizwan, P., Suresh, K., & Rajasekhara Babu, M. (2017). Real-time smart traffic management system for smart cities by using Internet of Things and big data. *Proceedings of IEEE International Conference on Emerging Technological Trends in Computing, Communications and Electrical Engineering, ICETT 2016*. https://doi.org/10.1109/ICETT.2016.7873660

Rossi, R., Ceccato, R., & Gastaldi, M. (2020). Effect of road traffic on air pollution. Experimental evidence from covid-19 lockdown. *Sustainability (Switzerland)*, *12*(21), 1–17. https://doi.org/10.3390/su12218984

Samuel, M., Mohamad, M., Saad, S. M., & Hussein, M. (2018). Development of edge-based lane detection algorithm using image processing. *International Journal on Informatics Visualization*, *2*(1), 19–22. https://doi.org/10.30630/joiv.2.1.101

Sarnat, J. A., Golan, R., Greenwald, R., Raysoni, A. U., Kewada, P., Winquist, A., Sarnat, S. E., Dana Flanders, W., Mirabelli, M. C., Zora, J. E., Bergin, M. H., & Yip, F. (2014). Exposure to traffic pollution, acute inflammation and autonomic response in a panel of car commuters. *Environmental Research*, *133*, 66–76. https://doi.org/10.1016/j.envres.2014.05.004

Sheilah, M., Napalang, G., Regin, J., & Regidor, F. (2017). Innovation Versus Regulation: An Assessment of the Metro Manila Experience in Emerging Ridesourcing Transport Services. *Journal of the Eastern Asia Society for Transportation Studies*, *12*, 343–355. https://doi.org/10.11175/easts.12.343

Shukla, P. (2011). Exploratory research design. *Essesntials of Marketing Research : Part 1*, 29–37. https://nscpolteksby.ac.id/ebook/files/Ebook/Business Administration/Marketing Research an Introduction (2008)/3. Chapter 2 - Exploratory research design.pdf

Sidel, J. T. (2020). Averting “Carmageddon” through reform? An eco-systemic analysis of traffic congestion and transportation policy gridlock in Metro Manila. *Critical Asian Studies*, 378–402. https://doi.org/10.1080/14672715.2020.1793681

Strydom, H. (2013). An evaluation of the purposes of research in social work. *Social Work (South Africa)*, *49*(2), 149–164. https://doi.org/10.15270/49-2-58

Tarel, J., Hautière, N., Caraffa, L., Cord, A., Halmaoui, H., & Gruyer, D. (2012). Vision Enhancement in Homogenous and Heterogenous Fog. *IEEE Intelligent Transportation Systems Magazine*, *April*, 6–20.

Tenorio, B. G. V., Joe, J. A. S., Tamayo, R. C. M., & Canoy, N. A. (2019). The Daily Grind: A Rhizomatic Approach to Narratives of Managing Commuter Stress During Traffic Congestion in Metro Manila. *Journal of Constructivist Psychology*, *32*(1), 63–80. https://doi.org/10.1080/10720537.2017.1410458

Vujović, V., & Maksimović, M. (2014). Raspberry Pi as a Wireless Sensor node: Performances and constraints. *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2014 - Proceedings*, *May*, 1013–1018. https://doi.org/10.1109/MIPRO.2014.6859717

Wang, Z. xun, & Wang, W. (2018). The research on edge detection algorithm of lane. *Eurasip Journal on Image and Video Processing*, *2018*(1). https://doi.org/10.1186/s13640-018-0326-2

Wong, R. C. (2012). *Information Technology & Software Engineering*. *2*(5), 7866. https://doi.org/10.4172/2165-7866.S11-e001

Yusuf, A. N. A., Arifin, A. S., & Zulkifli, F. Y. (2021). Recent development of smart traffic lights. *IAES International Journal of Artificial Intelligence*, *10*(1), 224–233. https://doi.org/10.11591/ijai.v10.i1.pp224-233

APPENDICES

Appendix 1

Letter of Intent to Parañaque City TPMO

July 04, 2022

SPO1 REYNALDO P. MURILLO
Head, Traffic and Parking Management Office
City of Parañaque City, Metro Manila

Dear Mr. Murillo:

Warmest greetings!

Recently, the PUP Parañaque, headed by Dr. Beata Maria F. De Ocampo, has initiated a connection with the **Traffic and Parking Management Office (TPMO)** regarding the Smart Traffic Project. Our team is also a part of the same project and is currently focusing on the use of image processing to detect traffic vehicles.

Our thesis is in line with the development of new technologies to alleviate the transportation issues that the city is experiencing and may also be a gateway towards more production and innovation of the technologies.

In order for us to successfully accomplish the scope and goals of this research, we would like to humbly request the participation of the TPMO for an interview. We believe that this will provide more support and credibility on the technology that we intend to establish.

Your positive response will provide a huge contribution towards the completion of this thesis. We highly appreciate your acknowledgement to this request.

Please feel free to reach us out through this contact number (+63) 907 902-0882 or email address at ilagan.professionaluseonly@gmail.com.

Respectfully,

SEBASTIAN KYLE ILAGAN
Student, BS Computer Engineering 4-1

Appendix 2

Interview Guide for Interview with TPMO

Setting

Objective: To gather essential information from TPMO regarding the traffic management system of Parañaque City

Participants: TPMO official/s

Date, Time, and Location: (to be discussed with the interviewee)

Mode of Interview: Vis-à-vis (preferred by the interviewee)

Interview/Discussion

1. What is the current technology that the TPMO is using for the alleviation and management of traffic in Parañaque City?

____

____

____

2. What are the stages undertaken by the TPMO in the development of the existing technology?

____

____

____

3. How effective is the existing technology towards minimizing the traffic?

____

____

____

4. What are the issues and challenges encountered by the TPMO administrators when implementing this technology?

____

____

____

5. What are the training programs initiated to the staffs, enforcers, and authorities on the implementation of this technology?

____

____

____

Appendix 3

Results and Discussion Tables

Table 3

Effectiveness of the System Prototype in Terms of Accuracy

Raspberry Pi

Trial No.	Accuracy	Measurement	Remarks
1	42.5	Percent (%)	Not Accurate
2	51.0	Percent (%)	Not Accurate
3	52.0	Percent (%)	Not Accurate
4	46.5	Percent (%)	Not Accurate
5	43.5	Percent (%)	Not Accurate
6	39.5	Percent (%)	Not Accurate
7	42.5	Percent (%)	Not Accurate
8	45.5	Percent (%)	Not Accurate
9	40.5	Percent (%)	Not Accurate
10	47.0	Percent (%)	Not Accurate
11	52.5	Percent (%)	Not Accurate
12	50.0	Percent (%)	Not Accurate
13	49.0	Percent (%)	Not Accurate
14	46.0	Percent (%)	Not Accurate
15	42.5	Percent (%)	Not Accurate
16	45.0	Percent (%)	Not Accurate
17	46.0	Percent (%)	Not Accurate
18	48.0	Percent (%)	Not Accurate
19	45.0	Percent (%)	Not Accurate
20	41.5	Percent (%)	Not Accurate
21	45.0	Percent (%)	Not Accurate
22	50.0	Percent (%)	Not Accurate
23	45.0	Percent (%)	Not Accurate
24	49.0	Percent (%)	Not Accurate
25	50.0	Percent (%)	Not Accurate
26	55.5	Percent (%)	Not Accurate
27	51.0	Percent (%)	Not Accurate
28	53.0	Percent (%)	Not Accurate
29	55.5	Percent (%)	Not Accurate
30	49.0	Percent (%)	Not Accurate
Average	**47.3**	**Percent (%)**	**Not Accurate**

Desktop-based computer system

Trial No.	Accuracy (%)	Measurement	Remarks
1	90.5	Percent (%)	Accurate
2	90.5	Percent (%)	Accurate
3	90.5	Percent (%)	Accurate
4	91.5	Percent (%)	Accurate
5	90.5	Percent (%)	Accurate
6	90.5	Percent (%)	Accurate
7	90.5	Percent (%)	Accurate
8	90.5	Percent (%)	Accurate
9	92.5	Percent (%)	Accurate
10	90.5	Percent (%)	Accurate
11	90.5	Percent (%)	Accurate
12	91.0	Percent (%)	Accurate
13	90.5	Percent (%)	Accurate
14	90.5	Percent (%)	Accurate
15	90.5	Percent (%)	Accurate
16	90.5	Percent (%)	Accurate
17	91.0	Percent (%)	Accurate
18	90.5	Percent (%)	Accurate
19	90.5	Percent (%)	Accurate
20	90.5	Percent (%)	Accurate
21	91.0	Percent (%)	Accurate
22	90.5	Percent (%)	Accurate
23	90.5	Percent (%)	Accurate
24	91.5	Percent (%)	Accurate
25	90.5	Percent (%)	Accurate
26	92.0	Percent (%)	Accurate
27	90.5	Percent (%)	Accurate
28	91.0	Percent (%)	Accurate
29	90.5	Percent (%)	Accurate
30	90.5	Percent (%)	Accurate
Average	**90.75%**	**Percent (%)**	**Accurate**

Table 4

Effectiveness of the System Prototype in Terms of Recognition Speed

Raspberry Pi

Trial No.	Recognition Speed	Measurement	Remarks
1	0.89	Milliseconds (ms)	Slow
2	0.47	Milliseconds (ms)	Slow
3	0.34	Milliseconds (ms)	Slow
4	0.30	Milliseconds (ms)	Slow
5	0.41	Milliseconds (ms)	Slow
6	0.82	Milliseconds (ms)	Slow
7	0.78	Milliseconds (ms)	Slow
8	0.41	Milliseconds (ms)	Slow
9	0.30	Milliseconds (ms)	Slow
10	0.23	Milliseconds (ms)	Fast
11	0.27	Milliseconds (ms)	Slow
12	0.24	Milliseconds (ms)	Fast
13	0.36	Milliseconds (ms)	Slow
14	0.21	Milliseconds (ms)	Fast
15	0.31	Milliseconds (ms)	Slow
16	0.35	Milliseconds (ms)	Slow
17	0.35	Milliseconds (ms)	Slow
18	0.28	Milliseconds (ms)	Slow
19	0.38	Milliseconds (ms)	Slow
20	0.31	Milliseconds (ms)	Slow
21	0.34	Milliseconds (ms)	Slow
22	0.35	Milliseconds (ms)	Slow
23	0.31	Milliseconds (ms)	Slow
24	0.34	Milliseconds (ms)	Slow
25	0.29	Milliseconds (ms)	Slow
26	0.30	Milliseconds (ms)	Slow
27	0.34	Milliseconds (ms)	Slow
28	0.39	Milliseconds (ms)	Slow
29	0.33	Milliseconds (ms)	Slow
30	0.33	Milliseconds (ms)	Slow
Average	**0.38**	**Milliseconds (ms)**	**Slow**

Desktop-based computer system

Trial No.	Recognition Speed	Measurement	Remarks
1	0.09	Milliseconds (ms)	Fast
2	0.09	Milliseconds (ms)	Fast
3	0.22	Milliseconds (ms)	Fast
4	0.18	Milliseconds (ms)	Fast
5	0.11	Milliseconds (ms)	Fast

6	0.26	Milliseconds (ms)	Slow
7	0.08	Milliseconds (ms)	Fast
8	0.11	Milliseconds (ms)	Fast
9	0.15	Milliseconds (ms)	Fast
10	0.09	Milliseconds (ms)	Fast
11	0.17	Milliseconds (ms)	Fast
12	0.18	Milliseconds (ms)	Fast
13	0.21	Milliseconds (ms)	Fast
14	0.17	Milliseconds (ms)	Fast
15	0.13	Milliseconds (ms)	Fast
16	0.11	Milliseconds (ms)	Fast
17	0.21	Milliseconds (ms)	Fast
18	0.18	Milliseconds (ms)	Fast
19	0.35	Milliseconds (ms)	Slow
20	0.16	Milliseconds (ms)	Fast
21	0.22	Milliseconds (ms)	Fast
22	0.28	Milliseconds (ms)	Slow
23	0.21	Milliseconds (ms)	Fast
24	0.12	Milliseconds (ms)	Fast
25	0.22	Milliseconds (ms)	Fast
26	0.32	Milliseconds (ms)	Slow
27	0.15	Milliseconds (ms)	Fast
28	0.25	Milliseconds (ms)	Fast
29	0.17	Milliseconds (ms)	Fast
30	0.30	Milliseconds (ms)	Slow
Average	**0.18**	**Milliseconds (ms)**	**Fast**

Table 5

Effectiveness of the System Prototype in Terms of Latency

Raspberry Pi

Trial No.	Latency	Measurement	Remarks
1	4.73	Seconds (sec)	Delayed
2	7.89	Seconds (sec)	Delayed
3	5.39	Seconds (sec)	Delayed
4	4.77	Seconds (sec)	Delayed
5	5.24	Seconds (sec)	Delayed
6	5.06	Seconds (sec)	Delayed
7	5.02	Seconds (sec)	Delayed
8	5.33	Seconds (sec)	Delayed
9	4.82	Seconds (sec)	Delayed
10	5.02	Seconds (sec)	Delayed
11	5.11	Seconds (sec)	Delayed
12	5.21	Seconds (sec)	Delayed
13	5.81	Seconds (sec)	Delayed
14	4.78	Seconds (sec)	Delayed

15	5.21	Seconds (sec)	Delayed
16	5.40	Seconds (sec)	Delayed
17	5.13	Seconds (sec)	Delayed
18	5.06	Seconds (sec)	Delayed
19	5.47	Seconds (sec)	Delayed
20	4.96	Seconds (sec)	Delayed
21	4.99	Seconds (sec)	Delayed
22	4.73	Seconds (sec)	Delayed
23	4.69	Seconds (sec)	Delayed
24	5.30	Seconds (sec)	Delayed
25	4.58	Seconds (sec)	Delayed
26	4.56	Seconds (sec)	Delayed
27	5.03	Seconds (sec)	Delayed
28	5.11	Seconds (sec)	Delayed
29	5.09	Seconds (sec)	Delayed
30	5.06	Seconds (sec)	Delayed
Average	**4.97**	**Seconds (sec)**	**Delayed**

Desktop-based computer system

Trial No.	Latency	Measurement	Remarks
1	1.05	Seconds (sec)	Prompt
2	1.03	Seconds (sec)	Prompt
3	1.01	Seconds (sec)	Prompt
4	1.00	Seconds (sec)	Prompt
5	1.08	Seconds (sec)	Prompt
6	1.05	Seconds (sec)	Prompt
7	1.01	Seconds (sec)	Prompt
8	2.01	Seconds (sec)	Prompt
9	1.02	Seconds (sec)	Prompt
10	1.08	Seconds (sec)	Prompt
11	1.08	Seconds (sec)	Prompt
12	1.20	Seconds (sec)	Prompt
13	1.17	Seconds (sec)	Prompt
14	1.01	Seconds (sec)	Prompt
15	1.22	Seconds (sec)	Prompt
16	0.87	Seconds (sec)	Prompt
17	1.04	Seconds (sec)	Prompt
18	0.76	Seconds (sec)	Prompt
19	0.97	Seconds (sec)	Prompt
20	0.82	Seconds (sec)	Prompt
21	0.87	Seconds (sec)	Prompt
22	1.22	Seconds (sec)	Prompt
23	1.02	Seconds (sec)	Prompt
24	1.00	Seconds (sec)	Prompt
25	1.07	Seconds (sec)	Prompt
26	1.13	Seconds (sec)	Prompt
27	0.97	Seconds (sec)	Prompt
28	1.03	Seconds (sec)	Prompt

29	0.89	Seconds (sec)	Prompt
30	0.97	Seconds (sec)	Prompt
Average	**1.06**	**Seconds (sec)**	**Prompt**

Table 6

Z-Score of Experimentation Results in Terms of Evaluation Variables

Parameters	z	Decision
Accuracy	-55.4355	Reject
Recognition Speed	5.4693	Reject
Latency	63.1929	Reject
Overall	**-51.5937**	**Reject**

Appendix 4

System Documentation

SMART TRAFFIC IMAGE PROCESSING SYSTEM (Smart TIPS): ADDRESSING THE TRAFFIC CONGESTIONS IN PARAÑAQUE CITY, METRO MANILA

July 2022

Business Requirements

BR ID	Requirements	Required	Want
BR1-01	System should provide clear and understandable information, instruction, and notification regarding usability.	✓	
BR1-02	System should run efficiently with only minor human intervention.	✓	
BR1-03	System should not burden operators, but rather assist them in improving their services.	✓	

Functional Requirements

FR ID	Feature	Requirement	Dependencies
FR1-01	Vehicle Detection and Tracking	The system is able to process the video traffic feed from the camera to detect and track vehicles entering the frame of the camera.	- Camera - Trained Model - Vehicle Detection Model
FR1-02	Monitoring	The system is able to display the processed video data from the camera to visually show the vehicles being detected and tracked in a central monitor.	- Camera - Trained Model - Vehicle Detection Model

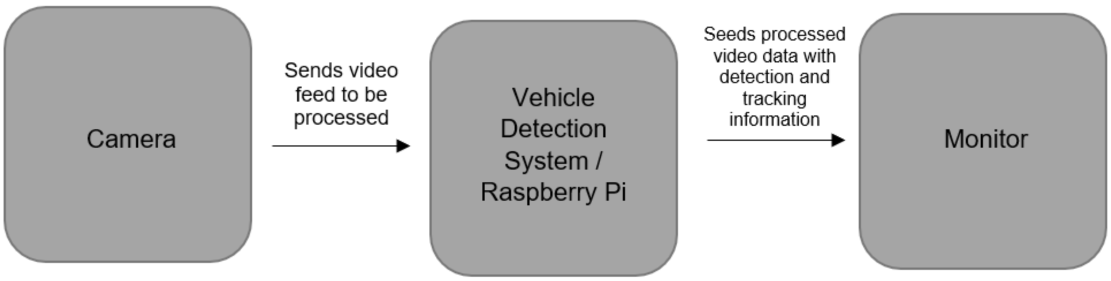
Non-Functional Requirements

NR ID	Feature	Requirement	Dependencies
NR1-01	Real-time	The system must have little to no delay from what is being seen by the camera to what being displayed in the monitor.	- Application - Hardware - Modules
NR1-02	Durability	The software and hardware side of the system must be capable of running for long periods of time.	- Application - Hardware - Modules
NR1-03	Standalone	The system must be able to work by itself without requiring any internet or cellular connection and other modules other than its main camera.	- Application - Hardware - Modules
NR1-04	Maintainability	The system must be maintainable though future updates.	- Application - Hardware - Modules

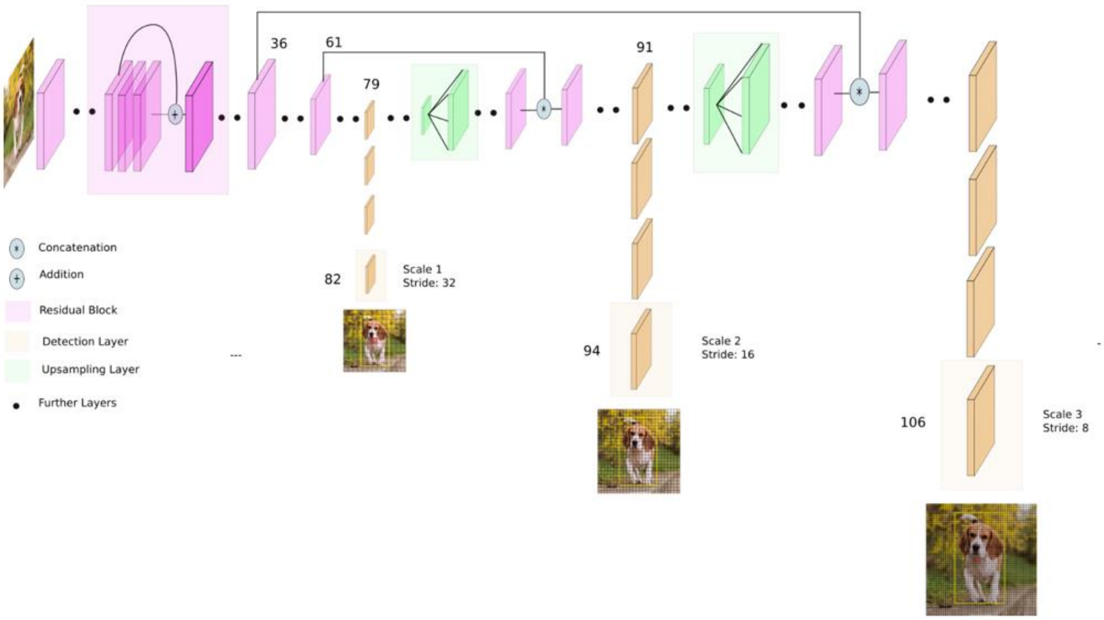
Project Evaluation (User Acceptance Test)

UR ID	As a…	I want…	So that…	Remarks
UR1-01	Traffic Officer	to use the system to check the current traffic status through detection of number of vehicles	I could be informed immediately which area or part of the road should be improved to assist the road management.	Passed
UR1-02	System Administrator	to monitor the system's functionality and status	I could manage the system and provide the necessary actions in case of any technical difficulties.	Passed
UR1-03	Future Researcher	to access the real-time logs of the system and observe how it works	I can use the system as a guide on working with object detection tools and gain reliable ideas in terms of creating valuable research regarding the current trends in smart traffic	Passed (for PC version only)

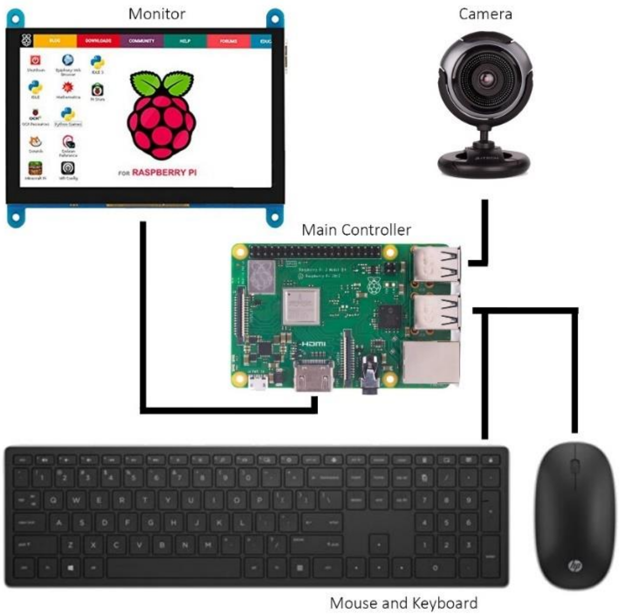
Context Data Flow Diagram

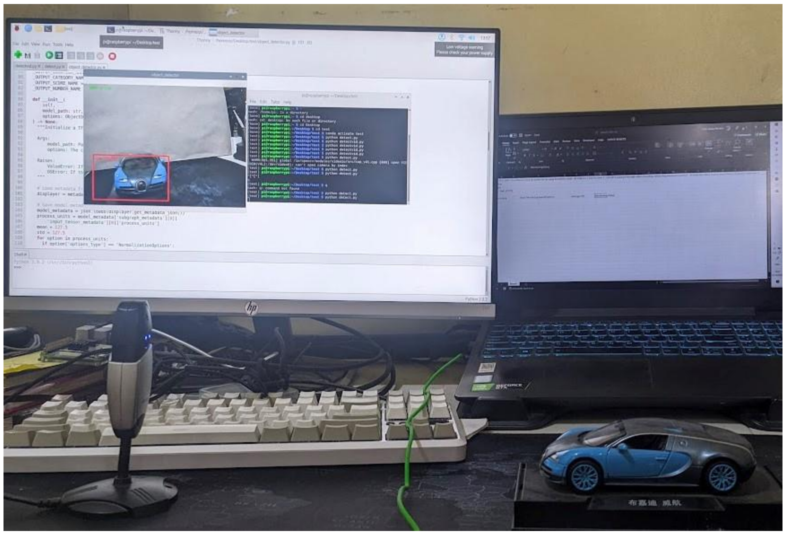


YOLOv3 Network Architecture



Iterative Development





```python
import os
# comment out below line to enable tensorflow logging outputs
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
import time
import tensorflow as tf
physical_devices = tf.config.experimental.list_physical_devices('GPU')
if len(physical_devices) > 0:
    tf.config.experimental.set_memory_growth(physical_devices[0], True)
from absl import app, flags, logging
from absl.flags import FLAGS
import core.utils as utils
from core.yolov4 import filter_boxes
from tensorflow.python.saved_model import tag_constants
from core.config import cfg
from PIL import Image
import cv2
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.compat.v1 import ConfigProto
from tensorflow.compat.v1 import InteractiveSession
# deep sort imports
from deep_sort import preprocessing, nn_matching
from deep_sort.detection import Detection
from deep_sort.tracker import Tracker
from tools import generate_detections as gdet
flags.DEFINE_string('framework', 'tf', '(tf, tflite, trt')
flags.DEFINE_string('weights', './checkpoints/yolov4-416',
                    'path to weights file')
flags.DEFINE_integer('size', 416, 'resize images to')
flags.DEFINE_boolean('tiny', False, 'yolo or yolo-tiny')
flags.DEFINE_string('model', 'yolov4', 'yolov3 or yolov4')
flags.DEFINE_string('video', './data/video/test.mp4', 'path to input video or set to 0 for webcam')
flags.DEFINE_string('output', None, 'path to output video')
flags.DEFINE_string('output_format', 'XVID', 'codec used in VideoWriter when saving video to file')
flags.DEFINE_float('iou', 0.45, 'iou threshold')
flags.DEFINE_float('score', 0.50, 'score threshold')
flags.DEFINE_boolean('dont_show', False, 'dont show video output')
flags.DEFINE_boolean('info', False, 'show detailed info of tracked objects')
flags.DEFINE_boolean('count', False, 'count objects being tracked on screen')

def main(_argv):
    # Definition of the parameters
    max_cosine_distance = 0.4
    nn_budget = None
    nms_max_overlap = 1.0

    # initialize deep sort
    model_filename = 'model_data/mars-small128.pb'
    encoder = gdet.create_box_encoder(model_filename, batch_size=1)
    # calculate cosine distance metric
    metric = nn_matching.NearestNeighborDistanceMetric("cosine", max_cosine_distance, nn_budget)
    # initialize tracker
    tracker = Tracker(metric)
```

```python

        # by default allow all classes in .names file
        #allowed_classes = list(class_names.values())

        # custom allowed classes (uncomment line below to customize tracker for only people)
        allowed_classes = ['car', 'bicycle', 'truck', 'motorbike', 'person']

        # loop through objects and use class index to get class name, allow only classes in allowed_classes list
        names = []
        deleted_indx = []
        for i in range(num_objects):
            class_indx = int(classes[i])
            class_name = class_names[class_indx]
            if class_name not in allowed_classes:
                deleted_indx.append(i)
            else:
                names.append(class_name)
        names = np.array(names)
        count = len(names)
        if FLAGS.count:
            cv2.putText(frame, "Objects being tracked: {}".format(count), (5, 35), cv2.FONT_HERSHEY_COMPLEX_SMALL, 2, (0, 255, 0), 2)
            print("Objects being tracked: {}".format(count))
        # delete detections that are not in allowed_classes
        bboxes = np.delete(bboxes, deleted_indx, axis=0)
        scores = np.delete(scores, deleted_indx, axis=0)

        # encode yolo detections and feed to tracker
        features = encoder(frame, bboxes)
        detections = [Detection(bbox, score, class_name, feature) for bbox, score, class_name, feature in zip(bboxes, scores, names, features)]

        #initialize color map
        cmap = plt.get_cmap('tab20b')
        colors = [cmap(i)[:3] for i in np.linspace(0, 1, 20)]

        # run non-maxima supression
        boxs = np.array([d.tlwh for d in detections])
        scores = np.array([d.confidence for d in detections])
        classes = np.array([d.class_name for d in detections])
        indices = preprocessing.non_max_suppression(boxs, classes, nms_max_overlap, scores)
        detections = [detections[i] for i in indices]

        # Call the tracker
        tracker.predict()
        tracker.update(detections)


        # update tracks
        for track in tracker.tracks:
            if not track.is_confirmed() or track.time_since_update > 1:
                continue
            bbox = track.to_tlbr()
            class_name = track.get_class()
            #class_name = track.get_class()
```

Appendix 5

Biographical Statement – Researcher 1

Sebastian Kyle L. Ilagan
B9 L1 Isabelle Garden Townhomes, Brgy. Moonwalk, Paranaque City
Contact No.: 0907-902-0882
Email: ilagan.professionaluseonly@gmail.com

Sebastian Kyle L. Ilagan is a fourth-year student taking up Bachelor of Science in Computer Engineering at the Polytechnic University of the Philippines. He received his Senior High School (SHS) Diploma under the Science, Technology, Engineering, and Mathematics (STEM) strand at the Saint Michael's College of Laguna in Biñan City. His current interests include a comprehensive plan to guide student leaders and he is currently the President of PUP Parañaque Central Student Council (PUPPQ CSC) for the Academic Year 2021-2022.

Appendix 6

Biographical Statement – Researcher 2

Vinci Louise Morales
6356A Saint Valentine St. SAV15, Brgy. San Isidro, Paranaque City
Contact No.: 0949-428-9164
Email: vincilouise.morales@gmail.com

Vinci Louise Morales is a 4th year student taking a bachelor's degree in Computer Engineering at the Polytechnic University of the Philippines. He is on track to graduate on AY 2022-2023 assuming no more hiccups occur with his subjects.

Appendix 7

Biographical Statement – Researcher 3

Allen Miguel Ramos
93 Deciembre St. Sunvalley Brgy. 195 Zone 20 NAIA, Pasay City
Contact No.: 0976-012-2816
Email: allenmiguelramos98@gmail.com

Allen Miguel Ramos is a 4th year student of Computer Engineering at the Polytechnic University of the Philippines. He received his Senior High School Diploma in Science, Technology, Engineering, Arts and Mathematics (STEAM) Strand at the Arellano University. He has faith in his Research teammates to get him through school in 2021–2022.