SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMAKURU-572103

(An Autonomous Institute under Visvesvaraya Technological University, Belagavi)

Project Report on

"Smart Traffic Light Management System"

submitted in partial fulfillment of the requirement for the completion of VI semester of

BACHELOR OF ENGINEERING

in

ELECTRONICS & COMMUNICATION ENGINEERING

Submitted by

Sunil Danappa Biradar	(1SI20EC097)
Thrinesh S	(1SI20EC102)
Varun M	(1SI20EC107)
Shilpashree S P	(1SI20EC112)

under the guidance of

Dr. Y Harshalatha

Assistant Professor

Department of E&CE

SIT, Tumakuru-03

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

2022-23

SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMAKURU-572103

(An Autonomous Institute under Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CERTIFICATE

Certified that the mini project work entitled "SMART TRAFFIC LIGHT MANAGEMENT SYSTEM" is a bonafide work carried out by Sunil Danappa Biradar (1SI20EC097), Thrinesh S (1SI20EC102), Varun M (1SI20EC107) and Shilpashree S P (1SI20EC112) in partial fulfillment for the completion of VI Semester of Bachelor of Engineering in Electronics & Communication Engineering from Siddaganga Institute of Technology, an autonomous institute under Visvesvaraya Technological University, Belagavi during the academic year 2022-23. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the department library. The Mini project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering degree.

Dr. Y Harshalatha	Head of the Department
Assistant Professor	Dept. of E&CE
Dept. of E&CE	SIT,Tumakuru-03
SIT,Tumakuru-03	

External viva:

Names of the Examiners	**Signature with date**
1.	
2.	

ACKNOWLEDGEMENT

We offer our humble pranams at the lotus feet of **His** Holiness, **Dr. Sree Sree Sivakumara Swamigalu**, Founder President and **His** Holiness, **Sree Sree Siddalinga Swamigalu**, President, Sree Siddaganga Education Society, Sree Siddaganga Mutt for bestowing upon their blessings.

We deem it as a privilege to thank **Dr. M N Channabasappa**, Director, SIT, Tumakuru, **Dr. Shivakumaraiah**, CEO, SIT, Tumakuru, and **Dr. S V Dinesh**, Principal, SIT, Tumakuru for fostering an excellent academic environment in this institution, which made this endeavor fruitful.

We would like to express our sincere gratitude to **Dr. K V Suresh**, Professor and Head, Department of E&CE, SIT, Tumakuru for his encouragement and valuable suggestions.

We thank our guide **Dr. Y Harshalatha**, Assistant Professor, Department of Electronics & Communincation Engineering, SIT, Tumakuru for the valuable guidance, advice and encouragement.

Sunil Danappa Biradar	(1SI20EC097)
Thrinesh S	(1SI20EC102)
Varun M	(1SI20EC107)
Shilpashree S P	(1SI20EC112)

Course Outcomes

CO 1 : Identify , formulate the problem and define the objectives

CO 2 : Review the literature and provide efficient design solution with appropriate consideration for societal, health and safety issues

CO 3 : Select the engineering tools/components and develop an experimental setup to validate the design

CO 4 : Test, analyse and interpret the results of the experiments in compliance with the defined objectives

CO 5 : Document as per the standard, present effectively the work following professional ethics and interact with target group

CO 6 : Contribute to the team, lead the diverse team, demonstrating engineering and management principles

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO-1	3												3	
CO-2		2				1							2	1
CO-3			2		2							2	2	2
CO-4				2										2
CO-5								2		2		2		2
CO-6									2		1			1
Average	3	2	2	2	2	1		2	2	2	1	2	2	2

Attainment level: - 1: Slight (low) 2: Moderate (medium) 3: Substantial (high)

POs: PO1: Engineering Knowledge, PO2: Problem analysis, PO3: Design/Development of solutions, PO4: Conduct investigations of complex problems, PO5: Modern tool usage, PO6: Engineer and society, PO7: Environment and sustainability, PO8: Ethics, PO9: Individual and team work, PO10: Communication, PO11: Project management and finance, PO12: Lifelong learning

Abstract

The problem of traffic congestion needs to be addressed in an efficient way, specially in the country like India. The present traffic control systems that are used are either manually controlled or automatically controlled with static time allocation. Both the manual and automatic control have drawbacks: one requires more man power to operate and is less efficient and other uses static time allocation which results in wastage of time. Therefore, a smart traffic management system is implemented in this project.

In order to improve the efficiency of traffic control, a Deep learning algorithm and an integrated system will be implemented. To optimize road capacity and traffic flow, this system will intelligently decide when to alter signals based on the number of vehicles on each road observed by cameras on each lane. Additionally, traffic density data is uploaded to the cloud for further analysis.

The system uses image processing to obtain binary-mask of the input image. A YOLOv8 model is used to count the number of vehicles. Raspberry-Pi handles real-time data and control the traffic light. The Raspbian OS desktop can be accessed using VNC viewer, and ThingSpeak is used to upload traffic density data to the cloud.

Contents

List of Figures

Chapter 1

Introduction

Nowadays, many countries suffer from the traffic congestion problems that affect the transportation system in cities and cause serious dilemma. In spite of replacing traffic officers and flagmen by automatic traffic systems, the optimization of the heavy traffic jam is still a major issue, especially with multiple junction nodes. The rapid increase in the number of automobiles and the constantly rising number of road users are not accompanied with promoted infrastructures with sufficient resources. Partial solutions were offered by constructing new roads, implementing flyovers and bypass roads, creating rings, and performing roads rehabilitation.

The traffic problem is very complicated due to the involvement of diverse parameters. The traffic flow depends on the time of the day where the traffic peak hours are generally in the morning and in the afternoon; on the days of the week where weekends reveal minimum load while Mondays and Fridays generally show dense traffic oriented from cities to their outskirts and in reverse direction respectively.

In this project, binary masked traffic images are given as input to the YOLO model. The YOLO model provides vehicle count of each road. Raspberry-Pi processes this real time data to give priority to the road which has the highest vehicle count. The traffic density data is uploaded to the cloud using ThingSpeak for further analysis.

1.1 Motivation

The number of vehicles is increasing rapidly in urban areas and metropolitan cities with an increase in population. The conventional traffic signaling system is not adaptable to changes in the amount of traffic. Traffic jam harms the economy, causes serious air pollution, and noise pollution, and thus worsens the overall environmental conditions.To overcome all these problems, a proper smart signaling system is critically needed to be developed and installed. This project aims to solve the problem by making traffic signals smart enough to distinguish between the roads which have different densities of traffic present in them. This project makes use of Deep learning model to detect the respective density on road and adjust the traffic signaling intelligently.

1.2 Objectives of the project

The objectives of the project are:

- To measure the vehicle count using YOLO model and to alter the traffic signaling accordingly.
- To upload traffic density information to the cloud for future analysis.

1.3 Organisation of the report

The report is divided into 7 chapters.

Chapter 1 contains the motivation, objective and organisation of the report. The literature survey of the project, which includes brief description of different techniques available for smart traffic light management system is presented in chapter 2. Chapter 3 deals with the system overview. The system description is explained in chapter 4. Chapter 5 describes overall software, algorithm and flowchart of the project. The results of the project and some snapshots are presented in chapter 6. The conclusion and scope of future work of the project is given in chapter 7.

Chapter 2

Literature Survey

The authors have proposed a system that can detect the amount of the traffic flow in each lane of the intersections and adjusting the current signal duration time smartly using Microcontroller AT89C51 and ultrasonic sensors. When there is no vehicle in the crossing roads, the testing device will work uninterestedly. Once the vehicles are detected in a lane, the vehicle in this lane will have the first priority to pass. The controlling port of the ultrasonic sensor sends out a high level signal which should be higher than 10 microseconds and meanwhile starts the timer. The port of the receiver receives the high level signal and then reads the result displayed by the timer. The result is considered as the available adjustment range for the current signal duration time. Further the actual distance can be calculated by the sound propagation speed. Considering the available signal duration time and the actual distance to the signal, the system will judge whether the vehicle is able to go through the intersection or not according to the distance between the vehicle and the intersection [1].

In [2], the system uses Internet of Things for IoT based computing, where different services such as server, storage and application are delivered for traffic management. The objective of this project is to design and develop IoT based traffic monitoring and control system. A network of sensors is used to track the number of vehicles and the traffic congestion at the intersection. The system controls the traffic light of each path in the city with the help of a central unit. The system uses Arduino Board for the application. Phone is communicated with system using IoT link. When the controller wants to connect to the system to control the traffic lights, he/she has to enter the password.

Results of a study states that, "Around 65-70 percent of deaths could have been prevented, if the emergency teams responded quickly". This paper proposes IoT enabled STMS which makes it possible to create Green Corridor for emergency vehicles when required. Emergency vehicles have RFID tag(s) installed on them. RFID scanners are placed at some distance from traffic lights. Traffic signals functionality is not altered when emergency vehicles are not in the vicinity of traffic lights. As soon as the RFID tag is scanned the traffic light switch to green and displays a message on the LCD screen to give way to the ambulance. The proposed system is capable to switch the traffic signal to green, when an emergency vehicle approaches the traffic signal, which the manual methods failed to do [3].

Conventional traffic system does not have proper monitoring system and often requires manual handing at traffic junction. This not only causes mental stress in passengers but also lot of fuel goes wasted due to delay at traffic junction. Therefore this paper proposes a system to handle traffic in a smart way by automatically adjusting its timing based on traffic density using Arduino Uno ATMega 328 microcontroller which consumes less power and has inbuilt ADC. In this, traffic is sensed using digital IR Sensors and IR Sensors detect vehicles further based on the signal reflected from them. Sensors are placed adjacent to the road to control the traffic density by changing traffic signal appropriately. All IR Sensors are interfaced with Arduino Uno and it reads data from IR Sensors [4].

Using the idea of RFID Technology, Microcontroller and card reader use SPI to communicate. The system adopted these modules in this project to measure the speed of a vehicle. The reader and tag interact with the 13.56 MHz electromagnetic field. The working principle of the RFID reader is the induction of electromagnetic waves. The RFID reader emits electromagnetic waves through the built-in antenna and reads the measured value of the RFID tag within a certain range of 0-12m. Whenever vehicle enters the range of an RFID reader it reads data stored on RFID tags. RFID tags placed on the vehicles are detected by the reader through the antennas. The EPCs retrieved by the reader are transferred to the microcontroller which calculates the speeds of the vehicles [5].

In this proposed system, data-set image will be sourced from the storage and converted to a greyscale image, later this image will be processed by the microprocessor (Raspberry Pi) with the help of OpenCV, a library of Python for Image Processing. The project will make use of binary images captured in real-time and a reference image which will be stored in the system. Thereafter, capturing the image the comparison will be made between the two binary images (i.e. captured image and reference image). A similar comparison will be performed on each lane and then the suitable output will be obtained from the process [6].

In [7], The system takes four captured images and four reference images of four lanes as input. Captured images are the images of roads with vehicles and reference images are the empty road images. The system converts RGB images into grayscale images. After conversion, Gaussian blur and edge detection operation is done for all the images and then it compares captured image of one lane with the reference image of the same lane. The comparison is done with all four images. Image matching is done and traffic density is calculated after comparison. After the calculation of traffic density, time is allocated. Among the four lanes the road with the highest density is allocated a green signal with more time and allowed to move vehicles and the rest of the lanes will be made to wait for some time.

The authors have proposed YOLOv3-live method based on the improvement of the YOLOv3-tiny network structure, which can perform real-time detection of vehicles in embedded devices. In the YOLOv3-live network structure, multi-scale receptive fields are used to extract features, making the extracted features more comprehensive and rich. At the same time, the downsampling operation in the original network structure is modified, and the feature map is delayed downsampled. Replace part of the pooled layers with convolutional layers of step size 2, reducing feature loss during downsampling. The experimental results show that the detection performance is improved compared with YOLOv3-tiny [8].

The authors have proposed YOLO v3-based nighttime vehicle detection method. All images were enhanced by an optimal MSR algorithm to alleviate the uneven brightness and to improve the sharpness and detail information. A nighttime vehicles dataset is established to train a pre-trained YOLO v3 network. Evaluating the testing dataset, the proposed method detects more vehicles, and achieves a higher AP and FPS values than the Faster R-CNN and SSD [9].

As vehicle detection is challenging for humans to calculate, this paper uses machine learning which has the benefits of automatically learning and improving over time tracking, classifying, and counting the number of vehicles passing by certain areas. This framework is used to train model weight files YOLO which is the deep learning algorithm for vehicle detection. This deep learning algorithm for vehicle detection is state-of-the-art object detection neural network that can be applied in small and large networks while maintaining speed and accuracy. The model then uses a different framework named TensorFlow. In this paper, the following steps are followed: Installation, collection of the images, labeling and classifying, training the YOLO model, running code in GitBash, python program development, and output video [10].

Chapter 3

System Overview

The Raspberry Pi is chosen as the primary processor in this project due to its capability to process photos and extract meaningful information, which is shown in Figure 3.1. As part of the project workflow, images are inputted into the Raspberry Pi for pre-processing tasks, including operations like masking and resizing. The pre-processed images are fed into a YOLOv8 model for accurate identifying and counting vehicles. Based on the vehicle count data received from the Raspberry Pi, the project incorporates a system to prioritize traffic signals accordingly. This prioritization ensures that traffic signals can adapt and optimize their timing based on the current traffic conditions. The collected vehicle count data is then uploaded to a cloud-based platform. This cloud integration allows for centralized storage and analysis of the traffic data.

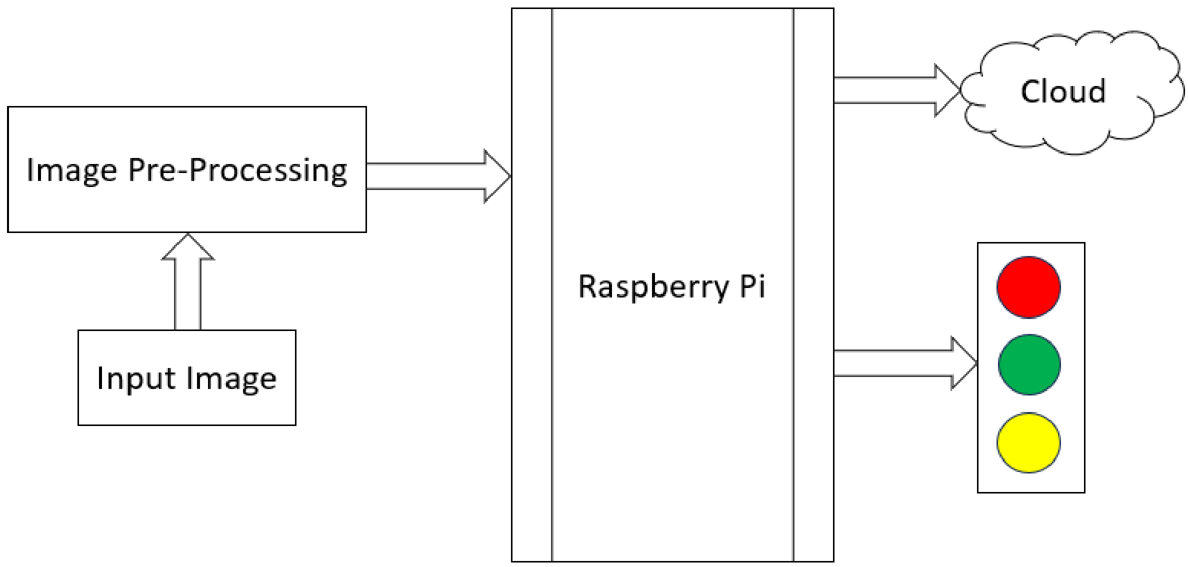

Figure 3.1: Block Diagram of Smart Traffic Light Management System

Input Image : The process of capturing or obtaining digital images involves using cameras, scanners, or other imaging devices to acquire images. This can also be done through the internet by downloading images from websites or extracting them from online sources. Some of the input images are shown in Figure 3.2. Blurred images can occur due to camera movement, subject movement or improper focus and low-light images where there is insufficient availability of sun-light are also considered for the analysis.

Figure 3.2: Input Images

Image Pre-Processing : It refers to a set of techniques applied to digital images before they undergo further analysis, interpretation, or visualization. The goal of image pre-processing is to enhance the quality of the images, remove noise or artifacts, adjust the image characteristics, and prepare them for specific applications or algorithms. Pre-processed images are shown in Figure 3.3.

Figure 3.3: Pre-Processed Images

Raspberry Pi : Raspberry Pi is the main processor which is used for image pre-processing and applying computer vision algorithms. The python programming environment is compatible with machine learning techniques. It comes with all the critical features of the motherboard in an average computer but without peripherals or internal storage. To set up the Raspberry computer, you will need an SD card inserted into the provided space. The SD card should have the operating system installed and is required for the computer to boot. Raspberry computers are compatible with Linux OS. It comes with the standard HDMI and USB ports, 1GB RAM, and Wi-Fi and Bluetooth connections in addition to the already functional Ethernet.

Cloud : Cloud Storage is a service which provides seamless scalability and it removes the necessity of operating databases which are distributed in nature. These days, having a storage device that is always connected to the network and accessible from anywhere in the world is a requirement. As a result, the data kept in the cloud can be utilised to study the situation and provide the user with complete information.

Traffic Light : Three signals typically make up traffic lights, and they use colour to communicate important information to drivers and riders. The standard traffic signal colours are vertically ordered red, yellow, and green.

Chapter 4

System Description

This chapter gives the detailed description of hardware components and model used in this project.

4.1 YOLO

A common task in computer vision is object detection. It focuses on locating a region of interest inside an image and categorizing this region in the same way that a standard image classifier would. YOLOv8 is the recent version of YOLO by Ultralytics that expands on the success of earlier editions as a cutting-edge, state-of-the-art (SOTA) model by adding new features and enhancements for improved performance, flexibility, and efficiency. The whole spectrum of visual AI tasks, including detection, segmentation, posture estimation, tracking, and classification, are supported by YOLOv8.

A fully convolutional neural network (CNN) is used by the single-shot detector YOLOv8 to predict bounding boxes and class probabilities simultaneously in an image. The architecture of YOLOv8 is shown in Figure 4.1. The architecture of YOLOv8 builds upon the previous versions of YOLO algorithms. This architecture consists of 53 convolutional layers and employs cross-stage partial connections to improve information flow between the different layers. The head of YOLOv8 consists of multiple convolutional layers followed by a series of fully connected layers. These layers are responsible for predicting bounding boxes and class probabilities for the objects detected in an image. YOLOv8 augments images during training online. At each epoch, the model observes a slightly different variation of the images it has been provided.

Anchor Free Detection: YOLOv8 is an anchor-free model. This means it predicts directly the center of an object instead of the offset from a known anchor box. Anchor boxes were a notoriously tricky part of earlier YOLO models, since they may represent the distribution of the target benchmark's boxes but not the distribution of the custom dataset. Anchor-free detection reduces the number of box predictions, which speeds up Non-Maximum Suppression (NMS), a complicated post-processing step that sifts through candidate detections after inference.

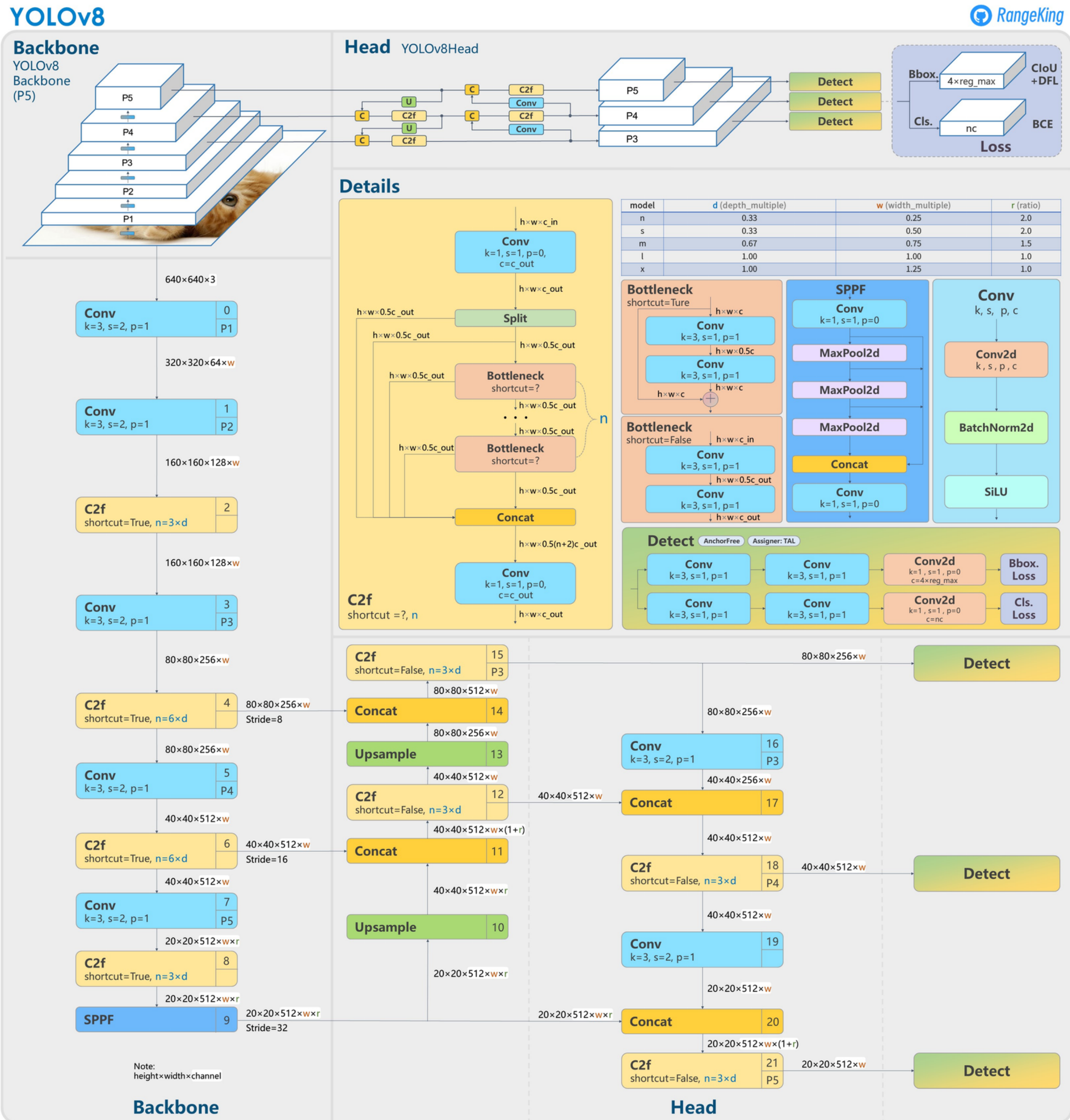

Figure 4.1: YOLOv8 Architecture

Advanced Backbone and Neck Architectures: YOLOv8 employs state-of-the-art backbone and neck architectures, resulting in improved feature extraction and object detection performance.

Optimized Accuracy-Speed Tradeoff: With a focus on maintaining an optimal balance between accuracy and speed, YOLOv8 is suitable for real-time object detection tasks in diverse application areas.

4.2 Raspberry-Pi

The **Raspberry Pi** is a small single-board computer is shown in Figure 4.1. With its GPIO pins and support for various programming languages, the Raspberry Pi allows for hardware interfacing and automation. The Raspberry Pi 3B is a versatile and capable single-board computer suitable for a wide range of projects and applications. CPU of 1.2GHz clocking frequency with 64-bit quad-core ARM Cortex-A53 processor. The GPIO pins can be used with a variety of alternative functions, some are available on all pins, others on specific pins.

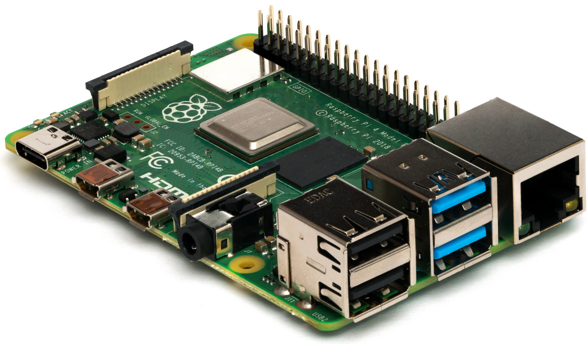

Figure 4.2: Raspberry-Pi

Specification:

- Clock frequency: 1.2 GHz
- Chipset (SoC): Broadcom BCM2837
- Processor: 64-bit quad-core ARM Cortex-A53
- Graphics processor: Dual Core VideoCore IV® Multimedia Co-Processor
- Memory (SDRAM): 1 GB LPDDR2
- Port extension: 40-pin GPIO
- Video outputs: HDMI
- Audio outputs: Audio Output 3.5mm jack
- Peripherals: 17 x GPIO
- Supply: 5V 2.5A via micro USB

Chapter 5

System Software

This chapter gives the detailed description of software applications, flowchart used in the proposed system. The algorithm used in this project as the following steps.

1) Import the required libraries and modules for object detection, image processing, and network requests.
2) Load the pre-trained YOLO model.
3) Set up the necessary credentials and configuration for the cloud server or API you want to send the count data.
4)Read the input images and their corresponding masks.
5) Pre-process the masks to extract the regions of interest using thresholding.
6) Define a function for performing object detection and counting:
   a) Accept the YOLO model, the pre-processed image, and any other required parameters as input.
   b) Use the YOLO model to detect objects in the image.
   c) Iterate over the detection results and extract the bounding box information.
   d) Identify and count the desired objects based on the detected class labels and confidence scores.
   e) Return the count.
7) Process each image and its corresponding mask using the YOLO model and the counting function. Store the vehicle counts for each image.
8) Send the count data to the cloud server or API using the appropriate URL parameters to include the count values.
9) Adjust the traffic signaling according to the count of vehicles on each lane prioritizing the highest count of vehicles.
10) Display the processed images with bounding boxes or any other desired visualizations.

5.1 Flowchart

The flowchart of the "**Smart Traffic Light Management System**" is shown in the Figure 5.1.

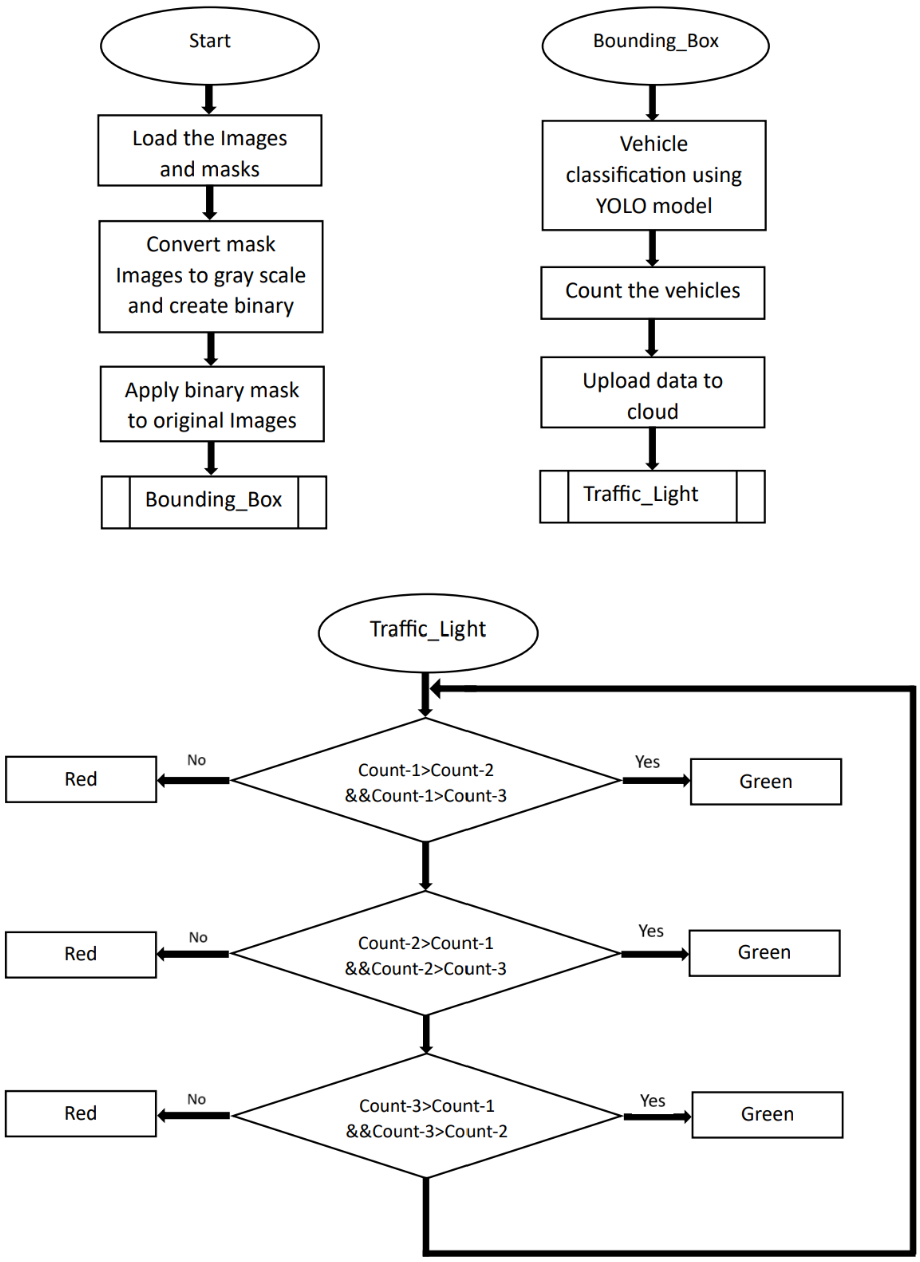

Figure 5.1: Flowchart

5.2 Software

The Software applications used in the project like Thonny, VNC viewer are described in detail.

5.2.1 Thonny

Thonny is a free Python Integrated Development Environment(IDE) for writing and testing Python code. Smoother code completion is possible with Thonny for both internal and external packages Thonny is well-suited for developing software components like data processing, image analysis, and machine learning, all of which are useful for implementing traffic management algorithms.

Figure 5.2: Thonny

5.2.2 VNC Viewer

A graphical desktop-sharing program called virtual network computing (VNC) uses the remote frame buffer protocol to operate another machine from a distance. For controlling local computers and mobile devices, we utilise VNC Viewer. VNC Viewer software allows access to and remote control of a computer from a device such a computer, tablet, or smart phone.

Figure 5.3: VNC Viewer

5.2.3 ThingSpeak

With the help of the IoT analytics platform service ThingSpeak, you can gather, visualize, and examine real-time data streams online. Clients can update and receive updates from channel feeds using the ThingSpeak MQTT broker. It is used to update traffic junction lane counts to the cloud.

Figure 5.4: ThingSpeak

5.2.4 Google Colab

Google Colab is a cloud-based Jupyter notebook environment. Colab provides a free computing resource (with limitations) and comes with pre-installed libraries and packages. Commonly used in data analysis, machine learning, and artificial intelligence.

Figure 5.5: Google colab

Chapter 6

Results

The hardware of the smart traffic light management system, which includes a Raspberry pi with a power supply connected to it, is shown in Figure 6.1. The system that monitors the number of vehicles approaching a signalized intersection. Consequently, control flow should be improved based on traffic conditions. This improves traffic flow management, reduces congestion, and minimizes the traffic delays. Figure 6.2 shows the density of vehicle at that instant of time using ThingSpeak cloud.

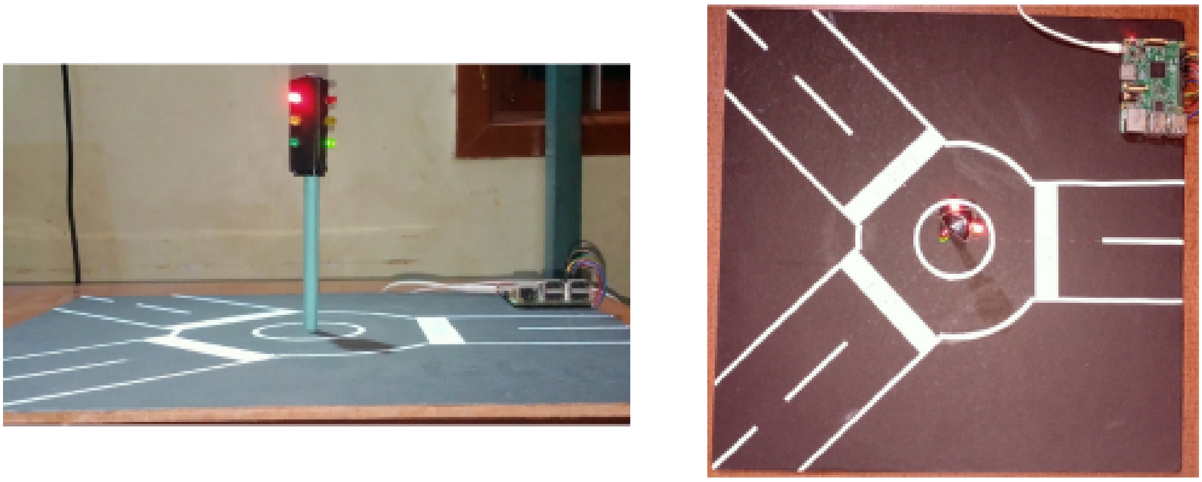

Figure 6.1: Prototype of Smart Traffic Light Management System

Sub-section 6.1 presents a comprehensive evaluation of the smart traffic light management system, assessing its performance and results under different time periods. Figure 6.3 showcases the results of the smart traffic light management system under normal conditions, demonstrating its optimal performance in terms of traffic flow optimization, reduced waiting times, and efficient signal control. In contrast, Figure 6.4 focuses on evaluating the system's performance under low light conditions. The results indicate that the system is capable of adapting to reduced visibility situations. Extends the evaluation to include scenarios with both low light conditions and blurred vision. The results highlight the system's ability to handle additional complexities, such as compromised visibility, and still provide reasonable traffic management outcome is shown in Figure 6.5.

However, Figure 6.6 reveals that the system's performance under night conditions is not as satisfactory compared to all other scenarios, including normal, low light, and low light with blurred vision conditions. The challenges associated with nighttime conditions, such as limited visibility and increased complexities, seem to affect the system's efficiency. Validation results of the YOLOv8 trained model are shown in Figure 6.7.

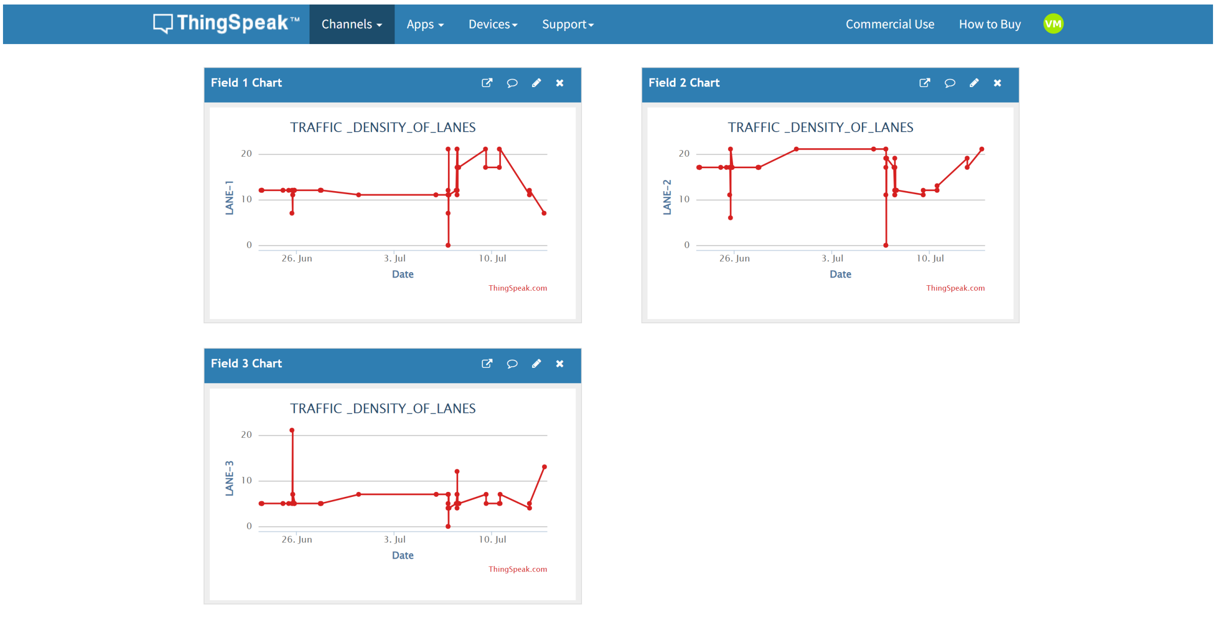

Figure 6.2: ThingSpeak

6.1 Analysis

Results under various circumstances are displayed in the figures below, which demonstrate the trained model's precision and accuracy of detecting vehicles under conditions including low light, blurred images, and night time.

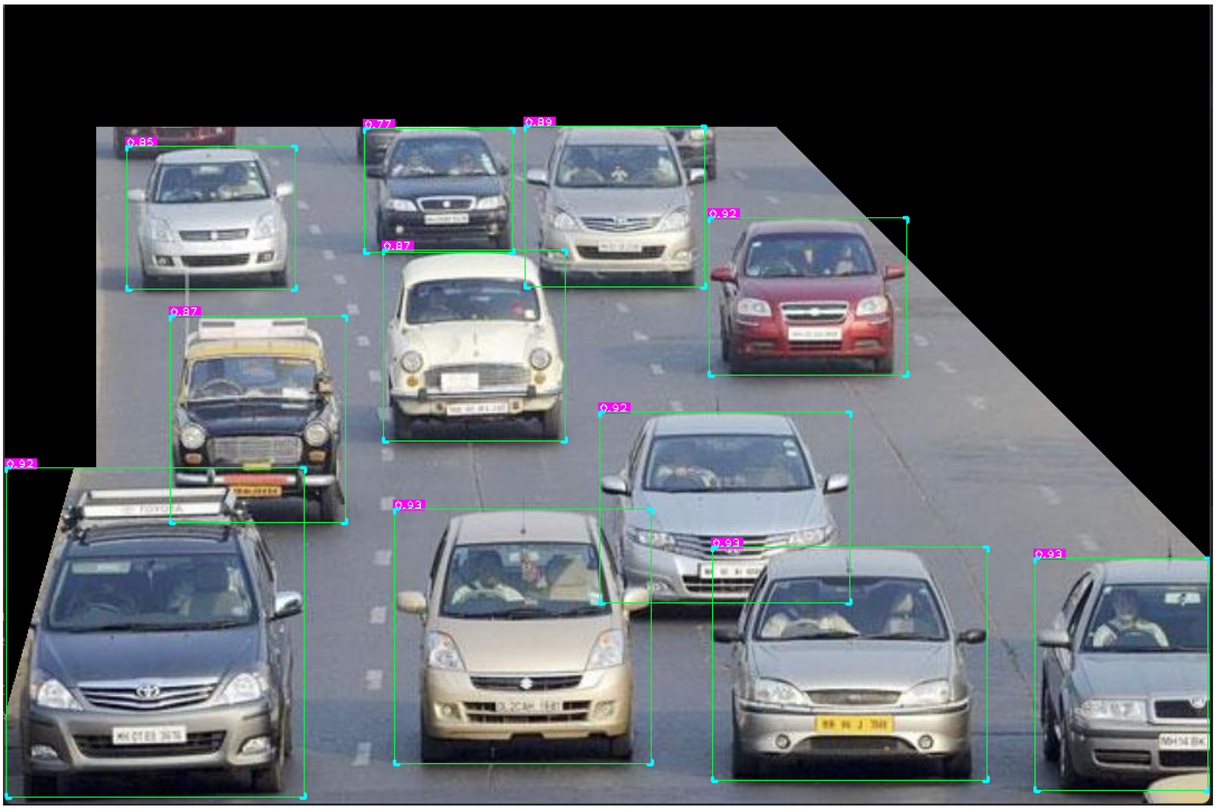

Figure 6.3: Normal condition

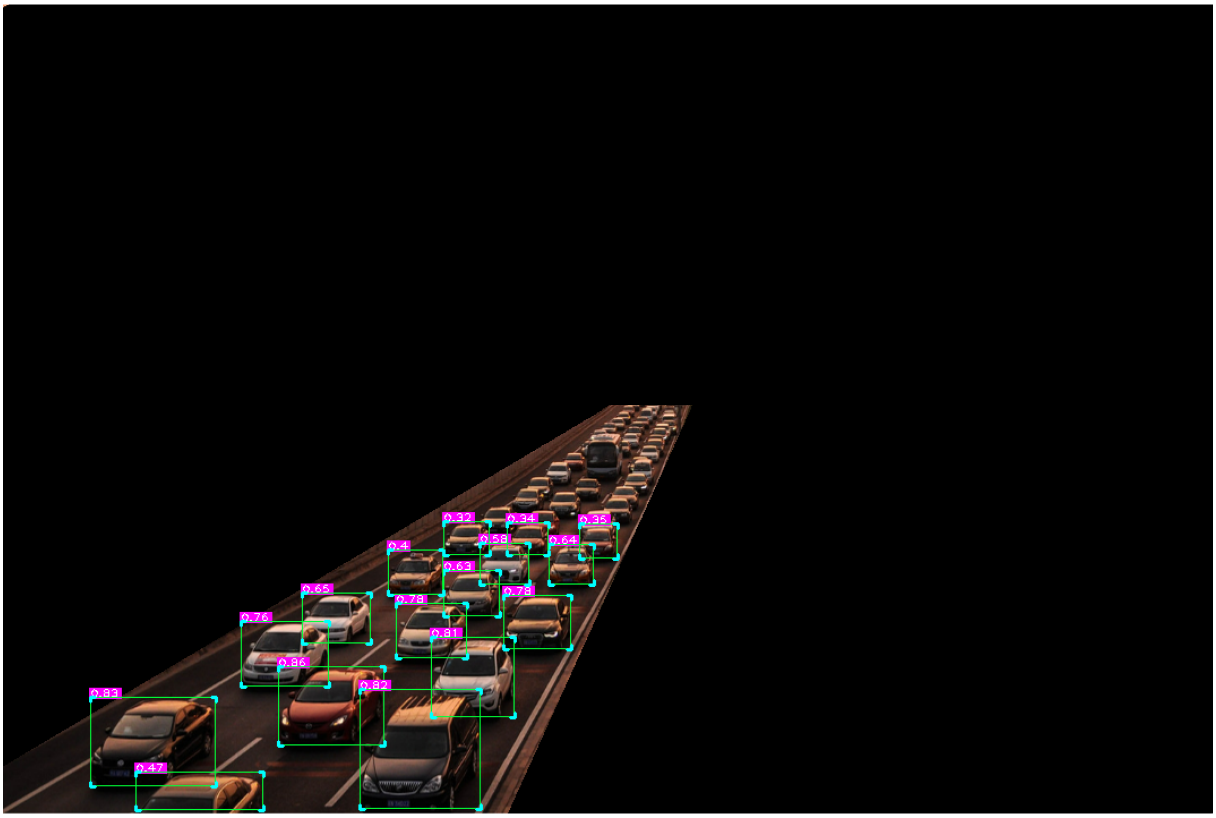

Figure 6.4: Low light condition

Figure 6.5: Low light with blurred condition

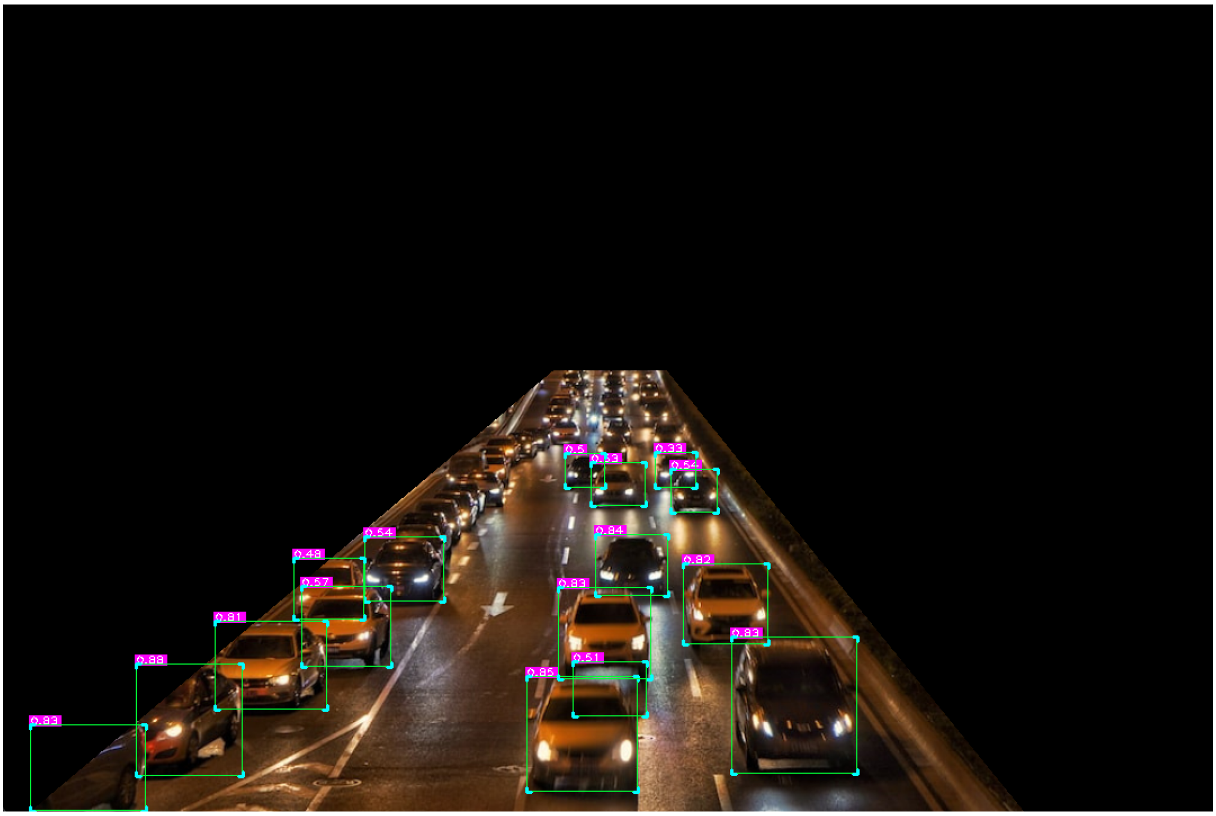

Figure 6.6: Night time condition

```
val: Scanning /content/drive/MyDrive/DATASET/vehicles/valid/labels.cache... 966 images,
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95)
                   all        966      13450      0.523      0.603      0.455      0.326
               big bus        966        273      0.827      0.432      0.722      0.545
             big truck        966       1162      0.825      0.504      0.676      0.444
                bus-l-        966          8     0.0577      0.875     0.0594     0.0364
                bus-s-        966         12      0.218      0.833      0.252      0.217
                   car        966       8537      0.884      0.712      0.836       0.53
             mid truck        966        257      0.754      0.428      0.444       0.34
             small bus        966         49      0.242      0.286       0.13     0.0808
           small truck        966       1721      0.729      0.508      0.615      0.407
              truck-l-        966        433      0.487      0.667      0.466      0.352
              truck-m-        966        629      0.434      0.695      0.389      0.303
              truck-s-        966        221      0.325      0.561      0.253      0.173
             truck-xl-        966        148       0.49       0.73      0.624      0.488
Speed: 0.4ms preprocess, 6.7ms inference, 0.0ms loss, 1.7ms postprocess per image
Results saved to runs/detect/val3
```

Figure 6.7: Validation

Chapter 7

Conclusion

The project titled "**Smart Traffic Light Management System**" is designed and developed using Image processing. The algorithm is developed using the YOLOv8 model for the detection and counting of vehicles. A traffic management system is developed by counting vehicles on every lane.

The system demonstrates optimal performance under normal conditions, effectively optimizing traffic flow, reducing waiting times, and efficiently controlling signals. It also showcases the system's adaptability to low light conditions, where it successfully manages traffic despite reduced visibility.

Furthermore, the system's capabilities extend to handling additional complexities, such as compromised visibility due to low light conditions and blurred vision. These findings highlight the system's ability to overcome challenges and maintain reasonable traffic management outcomes.

7.1 Scope for future work

The integration of density calculation on each lane further enhances the system's effectiveness by providing valuable insights into the flow of vehicles and identifying areas with high traffic density. This information can be used to optimize traffic signal timings, reroute vehicles, and implement targeted measures to alleviate congestion.

Bibliography

[1] Zhijun Li, Chunxiao Li, Yanan Zhang and Xuelong Hu "Intelligent Traffic Light Control System Based on Real Time Traffic Flows", 14th IEEE Annual Consumer Communications and Networking Conference (CCNC), pp.624-625, 2017.

[2] Manikonda, P. Yerrapragada, A.K, and Annasamudram, S.S. "Intelligent traffic management system", IEEE Conference on Sustainable Utilization and Development in Engineering and Technology, pp. 119-122, 2011.

[3] Dr. Vikram Bali, Ms. Sonali Mathur, Dr. Vishnu Sharma, Dev Gaur "Smart Traffic Management System using IoT Enabled Technology", 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), pp.565-568, 2020.

[4] A. Firdous, Indu and V. Niranjan, "Smart Density Based Traffic Light System", 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, pp. 497-500, 2020.

[5] Rohit Prasad, Himanshu Yadav, Devarsh Kumar, Sachin Pandey, Abhimanyu Yadav "Smart Traffic Monitoring and Controlling", Journel of Emerging Technologies and Innovative Research (JETIR), Volume 8, Issue 6, pp.f744-f748, 2021.

[6] U. E. Prakash, K. T. Vishnupriya, A. Thankappan and A. A. Balakrishnan, "Density Based Traffic Control System Using Image Processing," International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR), Ernakulam, India, pp. 1-4, 2018.

[7] D. Ijeri, P. Maidargi and R. Sunagar, "Traffic Control System Using Image Proccesing," IEEE Bangalore Humanitarian Technology Conference (B-HTC), Vijiyapur, India, pp. 1-6, 2020.

[8] S. Chen and W. Lin, "Embedded System Real-Time Vehicle Detection based on Improved YOLO Network," IEEE 3rd Advanced Information Management, Communi-

cates, Electronic and Automation Control Conference (IMCEC), Chongqing, China, pp. 1400-1403, 2019.

[9] Y. Miao, F. Liu, T. Hou, L. Liu and Y. Liu, “A Nighttime Vehicle Detection Method Based on YOLO v3,” Chinese Automation Congress (CAC), Shanghai, China, pp. 6617-6621, 2020.

[10] M. A. Bin Zuraimi and F. H. Kamaru Zaman, “Vehicle Detection and Tracking using YOLO and DeepSORT,” 11th IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE), Penang, Malaysia, pp. 23-29, 2021.

Appendices

Appendix A

Data Sheet of components

A.1 Raspberry Pi

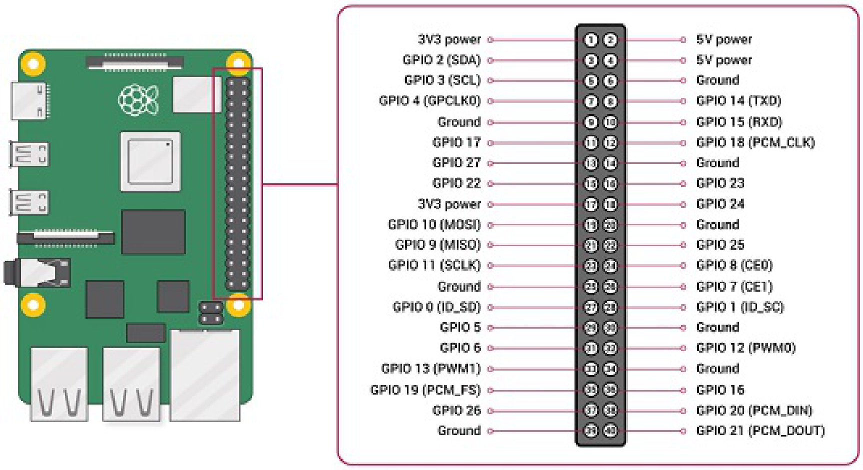

Figure A.1: Raspberry Pi 3

Processor: 64-bit quad-core ARM Cortex-A53

Clock frequency: 1200 MHz

RAM: 1GB

Wi-Fi : Yes

Bluetooth : Yes, 4.1

Power supply : 5v 2.5A

Storage: MicroSD card

Network adapter: Ethernet network card

USB ports: 4

2.5A USB: Yes

Self- Assessment of the Project

Project Title: Smart Traffic Light Management System

Level				
Poor 1	Average 2	Good 3	Very Good 4	Excellent 5

	PO PSO	Contribution from the project	Level
1	Engineering Knowledge: Knowledge of mathematics, engineering fundamentals engineering specialization to form of complex engineering problems.	Learning image processing and deep learning algorithm and applying them to Process the Traffic image.	4
2	System Analysis: Identify, formulate, research literature and analyse engineering problems to derive substaintiate conclusions by first principles of mathematics, natural and engineering science.	Aim to build a model which is able to detect the vehicles of various classes.	3
3	Design/development of solutions: Design solutions of complex engineering problems, design system components or process that meet the specified process with appropriate consideration for the public health, safety and the cultural and environmental considerations.	To learn Python and develop YOLO model to count the number of vehicles on each lane. To learn Python and masking the image, a OpenCV for pre-process the image, to develop a code according to the problem statement.	4
4	Conduct investigations of complex problems: Use research based knowledge and research methods including design experiments, analysis and interpretation of data, and synthesis of information to provide valid conclusions.	Project work is carried out on different input images. To Understand and train the YOLO model for detection and classification of vehicles. Compared the trained model with the existing pre-trained model based on mean precision value.	3

5	Modern tool usage: Create, insert and apply appropriate techniques, resources and modern engineering and tools including prediction and modeling to complex engineering activities with an understanding of the limitations.	The project was carried out using Python and Google Colab for building the YOLO model.	3
6	The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice.	The project recognizes the importance of prioritizing the safety of road users. By optimizing traffic light timings and minimizing delays.	4
7	Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.	In the development of the Smart Traffic Light Management System improves traffic flow management, reduces congestion, and minimizes the traffic delays.	4
8	Ethics: Apply ethical principles and commit to professional ethics and norms of the engineering prac ce.	Our project work is carried out in the provided slots. Teamwork and management skills.	4
9	Individual and Team Work: Function effectively as an individual and as a member or leader in diverse teams, and in multidisciplinary settings.	Equal and active participation is encouraged among the team members.	4

10	Communication: communicate effectively on complex engineering activities with the engineering community and with the society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.	Effective documentation is done using Latex (Overleaf).	5
11	Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.	Cost-efficient, flexible, and helpful for society.	4
12	Life-long Learning: Recognize the need for, and have the prepara on and ability to engage in independent and life-long learning in broadcast context of technological change.	In the context of technological change, the need for lifelong learning and the ability to adapt to new algorithms for achieving high accuracy in the Smart Traffic Light Management System is recognized.	4
13	PSO1: Apply the concepts of electronic circuits and systems to analyses and design systems related to Microelectronics, Communication, Signal processing and Embedded systems for solving real world problems.	Applied and analysed the concept of Deep learning to extract the features from datasets for building the model.	4
14	PSO2: To identify problems in the area of communication and embedded systems and provide efficient solu ons using modern tools/algorithms working in a team.	The project is developed using Python (Google Colab) by building YOLO model. And used the trained model for detection and counting the number of vehicles.	4