OLLSCOIL TEICNEOLAÍOCHTA
BHAILE ÁTHA CLIATH
DUBLIN
TECHNOLOGICAL
UNIVERSITY DUBLIN

Estimation of Vehicle Waiting Time in the Traffic Signal using Internet of Things

By

Rakul Rangaiyagowder Sampath

A thesis submitted to Technological University Dublin, for the degree of Master of Engineering in Internet of Things Technologies.

Supervised by Mr. Brian De Brit

Department of Engineering

TU Dublin – Blanchardstown Campus

August 2020

Abstract

This research thesis aims at providing an appropriate solution for a part of the issues related to IoT based Smart Transportation System. The traffic jam is a serious and negative impact on both society and our day-to-day life, a traffic signal is a significant need in transportation. Management of the traffic requires a strong and innovative technological solution rather than manual strategies, for example, human intervention viz. traffic police, traffic signals, and surveillance cameras. Traffic congestion is the major reason the workers finding it hard to reach their desired destination on time. Examining the streets and the traffic information is the progression taken towards the foundation of traffic conditions on some random street section. Additionally, the current vehicles do not have any inbuilt to alert the Jam before reaching the street. In this thesis, I have aimed to propose a system that effectively captures the traffic rate and alerts the users before reaching the street through a mobile application and keeps informed about the waiting through a dashboard for the people stuck in traffic. My frameworks utilize traffic detecting and updating the users to choose an alternative path having less travel time. The proposed framework encompasses the operations that are controlled by the deep learning neural network and the data is transmitted using the Raspberry PI, the analysed data is provided to the public using Smart Road Signs displays.

Declaration

I certify that this thesis which I now submit for examination for the award of Master of Engineering, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work.

This thesis was prepared according to the regulations for graduate study by research of Technological University Dublin and has not been submitted in whole or in part for another award in any other third level institution.

The work reported on in this thesis conforms to the principles and requirements of TU Dublin's guidelines for ethics in research.

TU Dublin has permission to keep, lend or copy this thesis in whole or in part, on condition that any such use of the material of the thesis be duly acknowledged.

Signature: Rakul Rangaiyagowder Sampath

Date: 31.08.2020

Student Number: B00130829

Acknowledgements

Research is a task requiring patience, perseverance, and strength of mind. There are moments of unfettered joy as well as great disappointments. I have been fortunate enough in having the unflinching support and encouragement of my supervisor Brian De Brit who provided the impetus and inspiration required for this arduous work. I take this opportunity to convey my deep sense of gratitude to him for his keen interest, invaluable suggestions, and untiring supervision throughout the course of investigation. Indeed, without his help this research would not have been possible. I sincerely respect him for his devotion to work simplicity, cooperation, and encouragement during this period. I deem it as a blessing from the Almighty to have the right person for my research guidance.

I am also deeply indebted to my friends for their heartful psychological support and pleasant company during the time of requirement. Most importantly I offer my deepest gratitude to my family for their love, care, encouragement, and constant emotional support right from the very first day of life without which I could have attained nothing.

Finally, I am deeply indebted to my University for providing me with the support and enlightened atmosphere for carrying out my research work.

Abbreviations List

IoT - Internet of Things

FPS - Frames Per Second

YOLO - You Only Look Once

CNN - Convolutional Neural Network

DNN - Deep Neural Network

API - Application Program Interface

AWS – Amazon Web Service

EC2 - Elastic Compute Cloud

UI – User Interface

LCD – Liquid Crystal Display

JSON- JavaScript Object Notation

AVI- Audio Video Interleave

MP4 - Moving Picture Experts Group

AI – Artificial Intelligence

CRISP DM- Cross Industry Standard Data Mining Process

RCNN - Region-based Convolutional Neural Network

UML - Unified Modelling Language

CCTV - Closed-circuit television

TMS - Traffic management Systems

DSG - Data Sensing and Gathering

DFPA - Data Fusion, Processing and Aggregation

RFID- Radio Frequency Identification

IR – Infra Red

GPS – Global Positioning System

GPRS -General Packet Radio Service

OCR- Optical Character Recognition

TCP - Transmission Control Protocol

UDP - User Datagram Protocol

HTTP - Hypertext Transfer Protocol

FTP - File Transfer Protocol

TCP - Transmission Control Protocol

IP - Internet Protocol

SSL - Secure Sockets Layer

TLS - Transport Layer Security

CV- Computer Vision

DLDT – Deep Learning Deployment Toolkit

IaaS – Infrastructure as a Service

PaaS - Platform as a Service

SaaS - Software as a Service

VM- Virtual Machine

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

The traffic issues were common in the real world. The use of more automobiles is the major cause of this issue. Especially countries with a huge density of population face a lot of complications with the traffic. Even after the implementation of a traffic signal system to manage the traffic is getting failed to solve the traffic jam. Current traffic signal and direction framework are typically intended for large street systems, requiring ongoing, high-resolution data. The present traffic research despite everything could not completely fulfil the basic principles of traffic arrangement, nor would it be able to predict the sudden jam. Even though the traffic jam in urban transportation is common, an hour traffic jam is an unpredictable and a nonlinear procedure, and congestion is the total traffic stream in time and flow. An experiment on the utilization of a full-scale traffic stream hypothesis to show traffic jams and got some essential aftereffects of jam prediction is done.[19] The outcome is believed to be a technological solution for the traffic condition in 50 seconds. Despite current research, traffic jam in a street structure is required to be precisely assessed continuously and used to help traffic control management.

The fundamental motive of this framework is to process out traffic density at a specific zone which is then further used to lessen the traffic jam brought by vehicles. During the peak hours of the day, the traffic is at its pinnacle and there are different issues identified with traffic jam. One such issue is fuel utilization. A normal of an hour and a half is spent day by day by a vehicle in a jammed traffic. Individuals neglect to turn off the motors which cause wastage of non-renewable assets, for example, petroleum, diesel, and LPG. This brings about a substantial financial loss to the nation. For the crisis conditions, for example, emergency vehicles, fire motors to go through, the traffic jam issues. To recuperate for the lost time spent in a jam, individuals will in general rush and scatter the clog, causing mishaps.

Traffic jam interfaces with the financial productivity of the city, since it forces extra costs that make all activities costlier and put a downturn of events. In a globalized world like today, the customer's rate is also found to be ever-increasing to an extent. For this, people should focus on jam controlling, and decrease, different sorts of expenses, including those that are identified with transport, for example, the time spent traveling, and the vitality devoured, the degree of air contamination, and the number of mishaps.

Since there are various alternatives methods proposed worldwide to address this issue, a city with the highest mishaps of traffic jam will drive away investors, anyway, positive other significant conditions might be. Although traffic jam may not be the main source, it tends to be a significant factor in the mass migration of different exercises from conventional downtown areas in search of conditions that enables a performance. There is a serious threat that the focus may stay just as the area of government foundations, small-scale organizations, and low-salary workers will start to disappear. Most importantly, the emergency vehicle will suffer a lot from this traffic congestion.

Even developed countries are constantly putting in their best effort to find solutions for the traffic issues through Smart Transportation Systems (STS) and Intelligent Transportation Systems (ITS). However, several factors need to be analysed and considered before enforcing the system. It may involve various factors, such as the budget, time constraints, labours involved, technology used, quality, and much more. To operate these factors in its maximum effectiveness and efficiency, the internet of things technologies has been used.

In this thesis, I focused to address the issues faced due to traffic jam and implement a solution through leveraging sophisticated technologies. The proposed system will aid the emergency vehicles and individuals to take the alternative traffic jam-free path to reach their destination in their desired time frame. On processing the live video of the traffic in real-time through image processing will produce the desired output through a mobile application and the dashboard for the users with the total amount of waiting time and time to pass by the signals.

1.1 Problem Statement

The emergency vehicles are often affected by getting stuck in the traffic jam even there is lots of advancement in technology been used. The duration of the traffic jam is also unpredictable. Despite other technologies used in the traffic management system, to minimize the traffic or to find an alternative solution for emergency vehicles to reach the destination in a very short time, this IoT research provides the best solution in predicting the traffic signal waiting time.

1.2 Research Question

- How IoT and Deep Learning helps in predicting the effective traffic signal waiting time?
- How the communication is provided to the user about the timely information about the particular traffic signal?

1.3 Research Objective

The entire project is focused to fulfil the research objective:

- To reduce the vehicles waiting time during the traffic signal by sending a proper alert to the users.

- To propose a technological solution that performs with high accuracy and precision in producing the output.
- To work on real-time data through leveraging image processing techniques and other sophisticated technologies.
- To build a mobile application that will incorporate and process the real-time data and synchronizes with the developed hardware system.
- To minimize the mishaps and reduce the time frame to send the output from the processed data.

1.4 Methodology

The proposed work is based on the Applied research methodology, since it provides the best solution for real-time traffic problems exist in the metropolitan cities having dense traffic. This research project also makes use of a Deep learning object detection algorithm using a Computer Vision module producing accurate results in image processing. Processing the vehicle detection and calculating the waiting time of the vehicles based on the count of vehicle and speed in which the vehicles are moving when the Signal timer is ON while crossing the traffic signal. If the movement of one direction vehicles trying to cross the road is estimated for one minute i.e., usual timer count and adding the timer count of other direction signals, then performing sum operation, vehicle who are trying to approach the particular signal get notified and choose an alternative path if the signal waiting time is more than usual. This is done by running the deep learning algorithm in the cloud, getting the output, and transmits instantly to the mobile application and the LCD. This research work does not use the Qualitative approach but uses the Quantitative approach in estimating the wait time. Since high accuracy and high processing speed are the only important factors considered in this project, YOLO, the pre-trained object detection framework is used to produce the result efficiently. Deploying all the programs in the cloud server and giving the input data result in a faster way of getting the output. The input video gets converted into several frames for the vehicle detection. However, the output data must be transmitted to the output devices through the medium is known as API. The output devices include mobile phones and LCD through which the user can view the timely information about the specific traffic signal.

1.5 Thesis Outline

The document structure is as follows. In the Second Chapter, the literature review is done with research focussing on the traffic issues, deep learning algorithm and overview of the components used. The Third Chapter focusses on the technology review where all the technical components used in the project is explained. The Fourth Chapter design focusses on the design methodology, hardware architecture and software architecture. The implementation and hardware setup are explained in Fifth Chapter. The results derived or obtained are explained in the Sixth Chapter. Finally, the challenges faced and the project future scope are explained in Chapter 7 and Chapter 8 respectively.

2 LITERATURE REVIEW

This is the important phase of the thesis. Before proceeding with the research implementation, the literature review should be done. The following study has been done to develop this research project.

2.1 Modern Transportation Systems

2.1.1 IoT in Transportation

IoT is a system that is connected to the cloud through internet to exchange data, control operations and to process the software application. In modern era, IoT in transportation uses information sensing devices like RFID, IR sensors and GPS, etc. The use of RFID via internet is to identify the object automatically and to share information as well. The main purpose is to track, monitor and manage the automobiles movement through remote sensing and controlling. It is because IoT can transmit the information timely and at the same time reliably since it uses intelligent processing.[1]

Considering an IoT based dynamic traffic management system, in a traffic signal if there are more vehicles on one end and very less vehicles on another side, the direction in which more vehicles waiting should be cleared by turning ON the green signal in one direction for a long time[4][7][8][10]. This should be controlled by the traffic police. The system is designed in such a way by using Raspberrypi Automation android application downloaded from Google Play store and this can be interfaced with traffic lights using Raspberrypi hardware. They use dynamic traffic light controlling algorithm and use WebIOPi (REST API in Raspberrypi) and it extends the GPIO of the Raspberrypi over internet through HTTP protocol. The IP address of the Raspberrypi will be configured with the android application. So, the authentication access will be given to traffic officer and the web camera will be attached to the traffic signal monitor the traffic and sends information to the control room. The traffic officer can monitor everything via android phone and can control the traffic lights when there is any emergency. The application login credentials will be encrypted and secured and given limited access to the officers [4]. The vehicle detection is done using Haar feature-based Cascade classifiers function which is a machine learning approach. Using OpenCV, vehicle speed is measured and based on the results obtained, green signal is turned ON dynamically [7]. Using Fuzzy control system, calculating the vehicle density with the help of sensors placed at certain distance from the intersections. Based on the count, fuzzy logic is applied to increase the signalling time dynamically [8].

By tracking the vehicle count at Active road lane, the threshold can be set. The vehicle density medium is classified as HIGH, LOW and MEDIUM. If the traffic density is more than 60%, signalling time will follow the fixed slot. If it is less than 60%, then signalling time can be decreased dynamically and the extra time will be given to the busy lane. When there is equal traffic, then fixed signalling time gets run. During emergency, the priority will be given to the specific lane in addition to 40% extra time slot. This real time traffic management algorithm is created using Code Composer Studio, an IDE to develop Texas Instrument Microprocessor applications [5].

Once the vehicle movement is recorded, removing all the noise and using edge detection algorithm. It is based on mathematical operations of derivatives of images. With that, calculating the distance between two vehicles, speed can be easily calculated. Once the speed is calculated, the estimated waiting time of other vehicles at a traffic signal can also easily estimated [9].

In order to control and provide information about the traffic at road networks by using sensors, RFID, and surveillance cameras fixed on the roads, this system provides a solution [6][12]. It has been classified into data acquisition and collection layer, data processing and decision-making layer, application, and actuation layer. They make use of RFID and ultrasonic sensors to detect the vehicles, camera to record the traffic. Blob detection algorithm is used to remove the noise in video streaming at local server. Once the traffic is detected, local server sends (density measurement via image processing) information to the microcontroller. The information is shared between many layers and based on the density of sensor values, vehicle count through video the results of traffic is calculated accurately, and the estimated results are displayed [6].

2.1.2 Smart Transportation Systems

The coordination of IoT and ITS leads to a new transportation system known as STS or Smart Transportation System. Considering some factors like environmentally friendly, reliable, security, convenient and predictable traffic, and transportation development, this came into existence. The main objective of this system is to reduce the carbon emission, pollution control, fuel consuming and emissions in addition to ease the traffic congestion. This is enhanced using mobile communication by providing best trip routines and auto payment options while crossing the toll gates thus making travellers a convenient for travel [1]. Since the travellers and traffic data were collected and stored in one database, it makes very easy to predict the traffic at different time helping to improve the infrastructure planning along with the traffic capacity [1][11]. Our proposed system is based on the Smart Transportation system.

Considering a smart transportation application, they are using route optimization algorithm to find the best vehicle route during the peak hours [14]. Here, the Bus Stop Interface (BSI) is connected with the Driver Assistance System (DAS) using wireless 3G network, GPS, microcontroller, GPRS, display and android application. The DAS connected with GPS shows the bus location in real time. The user through BSI placed at bus stop and by using android application can apply for request to board the bus service. However, there will be two kind of routes that buses follow. In major cities due to traffic some vehicles will be diverted by the drivers in specific routes. The bus which runs through this kind of roads are to be told as dynamic buses. The buses which follow the same routes whatever the situation is said to be as static buses. The user when tries to request for the bus service, by the concept of Dijkstra algorithm, the bus which runs through shortest route and passing the bus stop and considering space available in the bus will be receiving a request. If the condition is not satisfied, it looks for another bus service. This system can save time while using public transportation and improves the existing infrastructure at minimum cost [2].

Considering a metropolitan city where there are 4000 automobiles are sold every day and the traffic congestion is usual. During the peak time manual control of traffic or manual operating the traffic signal timer will not resolve the issue. The vehicles get stuck in the signal more than the time taken to reach their destination. Even emergency vehicles are affected because of this traffic. So, automatic creation of green corridor is essential. Here, the 360” rotational camera along with the acoustic sensor are placed at the traffic signal. The camera will capture all the

4 sides of the road and where there is more vehicle density, it increases the timer count. Likewise, it clears the traffic. Moreover, the acoustic sensor senses the emergency vehicle sound and based on the sound intensity from the vehicle approaching direction the signal will be made open. By this way, vehicle waiting time is reduced in the traffic signals [3].

2.1.3 Intelligent Transportation Systems

The ITS is an advanced technology relatively based on perfect road infrastructure. It uses electronics, sensors, control systems, to provide accurate information for the efficient management of transportation system thereby contributing to ease the traffic congestion[1][13].The system includes traffic detection, traffic control, communication technology, data processing and providing information regarding status of the traffic and the best trip route. This technology is implemented in most of the countries like United States, Japan and European countries with the relevant research and strategies to solve traffic problems. Since it solves most of the traffic problems still in the initial stage of implementation in most of the countries as it has some drawbacks [1].

Consider, assessing the adequacy of ITS developments in reducing mishaps that affect the study development of models (for example, event postponement and blocking models) that can reliably forecast the frequency of term in addition to the scale of non-recurring clog.[21] The study of the traffic movement of the expressways under congestion took place in the context of the norm of traffic features, using the case of the occurrence of the traffic jam. [22]

2.2 Traffic Management and Risks

2.2.1 Risk parameters in a traffic jam – a study

A model was built for local traffic jam delay in a closed system, variable capability of the urban transport network with a broad volume and restricted details. This may be done to solve the problem generated by traffic.[16] Various viewpoints have been focussed on the creation and degeneration of traffic jams by exploring that it is impossible that road construction or car integration would be able to alleviate any big urban traffic problems sooner rather than later, for example in the next 5 to 15 years. Traffic jam arises where a flow of traffic or a flexible break creates a larger demand for room than an available road cap. There are many obvious factors that induce or avoid congestion; the overwhelming majority of them restrict the potential limit of a path at a certain point or across a certain duration or increase in the amount of cars, based on the number of people or products. [17]

2.2.2 Traffic Jam Analysis

Limit delivery contemplates uncovering the methods such as laissez-faire allocation, traveller load distribution, incline metering, road and parking charging, ride justification allocation, rationing, and mixed techniques may be used to minimize traffic jam. Measurement of traffic congestion should be feasible by comparing the frequency and operating characteristics of the production of traffic. An inquiry on the calculation of traffic queues is carried out to determine the degree of congestion that can be used as a fair and enhanced proportion of adequacy. This is the statistical definition of the degree of traffic control in a mathematical context.[18]

Also, displaying congestion has provided a theoretical basis for identifying the multiple forms of cars in a traffic circle, which is useful for promoting congestion management arrangements. The show period criteria may be used to measure the degree of congestion. A technique is developed to determine the duration of the traffic jam in a specific street area and the probability that the traffic jam might have been the starting point would stop within the defined timeline.[20]

2.2.3 Risk Management

Risk assessment is a central concern of project management. The initial risk assessment phase is a chance to differentiate between risk recognition. It requires the identification of potential risk-causing elements and a description of the possibilities. Insightful Transit Networks are witnessing improvement from initiative presentation to being a regular solution that is open to transport organisers.[23] Consequently, the evaluation is one of the most important and substantial measures to be taken before any the plan can be forwarded. Protection has been growing as late as a domain of common interest, thought, and awareness transport. While recent studies have shed some light on driving-rate variables as well as on impact measurement, there is still a shortage of knowledge to take specific assessments into account. [24]

2.2.4 Technological Solution of Risk Management

Traffic jams give rise to danger and, lastly, may contribute to mishaps where urban mishaps have the highest rate of effect (75 per cent) of the overall mishaps; along these lines, they are talking about a crucial incident that can contribute to unfortunate outcomes. Artificial intelligence may be valuable in offering ground-breaking methods of identifying the primary explanation for crashes and traffic jams. Accident prediction forecasts or where the research methodology is employed daily to maximize the decline in the number of incidents that arise because of interstate improvements.[25]

Conventional dark spot projects have been set up with the goal of enhancing the protection of the cities, identifying them as accident prone to a full amount of accidents.[26] Events, categorized as impromptu occasions which, by the way, reduce the road limit, lead significantly to urban congestion. Transportation departments also set up emergency response systems to assist road capability and to adapt to incidents. Traffic analysis, considering all, cannot accurately foresee in which circumstances a "traffic jam" (rather than chaotic, yet quickly streaming traffic) may unexpectedly arise. Traffic jam is a common steady situation. A few metropolitan centres have worked out how to break out of their car dependency. A ton more of them have not sacrificed themselves to crowds. The approach introduced in-community to the problem of urban congestion and transport is an awareness of their needs and a test of how successful their actions can be. Urban planning and potential trends of economic growth should be crucial factors for any metropolitan pollution and preference of transport.[27] The proposed urban transport system would not take into consideration of rise in the number of motor cars, the expansion of the area or the climate. Given the great benefit to the protection of each resident, the expense and the benefits of inner city connectivity, one would assume that the reliability of the network and the strategic development of the accessible vehicle will be one of the main fields of concern of the national government.

2.2.5 Smart Traffic Management System

A smart traffic management system is partially implemented in Cambridge Area, where the road is guarded by queue detectors that identify the traffic line and alert the focal control unit that takes the same decision. Although the mechanism is placed together, it may be removed due to device management problems. Researchers used security cameras to identify traffic and OCRs to recognise vehicles by number plate recognition, which is a common exploration technique, but the system would fail in Pakistan as there are multiple forms of traffic, like bikes, donkey trucks without a number plate. [28]

An application system is introduced in which security cameras have been used to measure the traffic density using MATLAB, the traffic controller and the remote transmitter used to submit pictures to the worker after the worker has calculated the traffic density by using certain pictures in each line. This system has been built utilizing fixed edges that depend on different road vehicles. The measurement was used to calculate the duration of the red light cycle for the particular route of the crossing point, which is regulated by the traffic density on the road and sent to the microcontroller and the worker afterwards.[29]

Similarly, by using observation sensors, MATLAB and KEIL (Microcontroller coding) to monitor traffic jam. The paper also addresses the priority road clearing and the red sign dealer (Number plate recognition). Owing to the usage of extensive facilities, it is challenging to track and turn out to be costly.[30]

For emergency responders utilizing IOT, a model using the Intelligent Traffic Framework is introduced. In order to reduce traffic congestion and include a clear path to urban crisis vehicles, an revolutionary ITS infrastructure has been introduced, taking into consideration the needs of crisis vehicles, depending on the form of event, and a proposal to recognize and respond to traffic light hacking in this article. Emergency Control Room gathers emergency details and communicates to the Central Traffic Controller using the first approach by which intrusion is not used, and also manages traffic signals to clear roads by disaster vehicles. In the case of hacking, if the particular traffic light profile is deviating from the ordinary template, it may very well be deciphered as a compromised signal by the focal traffic regulator by using a technique such as the identification of empirical abnormality employed in intrusion detection may be used to discern hacking. Once hacking is remembered, the regulator restarts ordinary traffic patterns in absolutely hacked traffic lights. [31]

2.2.6 Real time Traffic Management System

A real-time synchronization-based technique is evaluated on the system for the efficient handling of the traffic source. Using handheld phones, the sensors were used to sense traffic from vehicle to vehicle, and the device communication to the driver was done. Sensors are positioned in the middle of the crossing point where vehicles and criteria and protocols are implemented utilizing the FIFO technique. [35]

Consider, the smart traffic routing system that selects the most selective path with the least congestion. Sensors are used to capture traffic density details, and these sensors use both solar and battery power. Sensors proceeded to transmit infrared light and, while the car was closer, they discern traffic intensity by testing the reflected light from the automobile. In either event, the readings can vary with the temperature and humidity modification.[32]

In the same manner, another system through which the essential objectives were to classify vehicles and acquire their region by using sensors and RFIDs after obtaining information transmitted to a centralized control centre by utilizing wireless contact for additional handling. Analysts used cloud networking, RFIDs, GPS, remote sensor organization (WSN), specialists and other latest technologies and developments to capture, archive, track, and control traffic data.[33]

2.2.7 Smart Traffic Management Frameworks

A framework is introduced to handle traffic jams using the Raspberrypi module, switches, ultrasonic sensors, etc. Detection of traffic and warning service offered by the unit. The traffic detection algorithm is established, the configuration of which is easy and programmable. The smart traffic-ready system uses the ultrasonic sensor to detect the traffic jam at that stage, by means of the switches used to connect the Raspberrypi board with the staff, it transmits a sign to the Email staff. Therefore, the e-mail worker sends a traffic congestion alert to the traffic police in a certain location. The bottom line of this system is that it reacts by submitting alerts and warnings to the police. The cost element is the main disadvantage as Raspberry pi is expensive and ineffective.[34]

Traffic jam can be tracked by simulation is suggested and has a research centre model named the Data Acquisition System. It is structured to withdraw from contradictory and potentially dangerous instructions, such as the shutting down of traffic lights. This paper affirms that if the system will provide an effective answer in different rush hour traffic jam situations as needed. In this model, Arduino sheets are used to monitor the set of indications, whether they are virtual or not. This paper evaluated the new traffic system by evaluating their responses and agreeing on their competence. In this context, the key explanation for the management was to focus on the process reactions in the circumstances. This paradigm is of use both for instructional purposes and for the present system. This paper suggested a flexible system for traffic lights and the fundamental goal is to direct the perplexing structure for traffic management by using different modules. [41]

Consider a smart traffic management system that offers insight on how to track or control traffic enforcement by using the Internet of Things. The Internet of Things has different focus points, such as limited commitment, strong unwavering consistency. And it is never affected by the awful atmosphere. This paper discussed the problem of traffic jam and suggested a potential approach. The coded position of automobiles using the vehicle tag is dropping in the snare due to its poor identification rate and its unfavourable or horrible environment. This paper introduced a worldwide unique electronic force regulating code as a personality ID or vehicle identifier rather than tag as is often utilized for RFID to scan for an EPC file. Next invention is GPS through which the condition of the vehicles can be gathered and in conclusion, GPRS which offer quick administrations to the clients. Mainly RFID peruse is used to interpret the EPC code that can be downloaded on the car. Furthermore, the challenge is that it is not sufficient for a professionally configured search and for cars and roadways in a canny rush hour traffic jam to follow the system. [40]

There is another framework for thickly populated territories gives a thought regarding the traffic condition in thickly populated territories like Amsterdam and Los Angeles. This paper proposes an advanced traffic management framework which is actualized utilizing Internet of things. Afterward, a design was proposed dependent on enormous data analysis including Hadoop. This paper likewise gives a thought of directed realizing which will help in introducing or deciding the norms of street, traffic cost, computing the normal speed of various vehicles out and about. This paper additionally proposed a thought for the inadequacy of the conventional traffic management framework. The key benefit of this framework is that it gives Big Data examination and RFID. Also, this paper has some structure multifaceted nature issue. [38]

IOT gadget has been produced for Traffic management framework have an idea on the advancement of internet of things for the traffic management framework. They have utilized an IOT cloud worker. They have likewise presented GREEN LIGHT Stage TIME which helps in decreasing the line length and holding up season of vehicles on the streets. The framework moreover replaces the current arrangement of traffic police on the job in the first part of the day and night top hours. In this paper, IOT Cloud worker is given, which is the fundamental preferred position for overseeing and checking the traffic jam. There are sure limitation present like fixed cycle TLS cannot adapt to drastically increment of enlisted vehicles.[39]

2.2.8 Challenges and Innovative Approaches on Traffic management Systems

A methodology used to boost TMS efficiency in Broad and Smart Cities has been applied. These strategies have been developed to such a degree that they may be applied to the continued growth of the community and to the creation of potential well-off metropolitan areas. In TMS, the most important stage is the DSG, where heterogeneous road control hardware tests traffic limits and periodically sends these readings to a focal point. At that point, the data will be melded and collected during the DFPA stage to separate useful traffic data. Data Manipulation requires the data to be processed. At last, the operation data stage at which TMS forwards this knowledge to the end customer. The upside of this system is that it offers the possible usage of shrewd cars and internet-based existence to enable swift and reliable detection and relief of traffic jams. Containment is also present, which means that this device means insufficient to generate a safe and secure TMS capable of handling vehicle ascents in genius urban populations. [36]

2.2.9 Programmed Intelligent Traffic Control System

Programmed Intelligent Traffic checking framework has been proposed to look on the issue looked by emergency vehicle, need based vehicles and how to control the traffic thickness. This is a continuous situation. Initially, they proposed an answer for emergency vehicle by utilizing RFID idea. RFID is utilized to green the rescue vehicle path. At that point, this paper gives a thought for need based vehicles. For this, they have utilized Infrared transmitter and recipient for greening the signs of vehicles path. By this, congestion must be forestalled. What is more, in conclusion, they concentrated on the density of traffic. The upside of behind this framework is that it gives IR transmitter and Receiver and RFID for crisis or need based vehicles. The constraint behind this is, there is not any programmed control on the traffic density and quick treatment of the patient is beyond the realm of imagination. [37]

2.3 Deep learning and Neural network

Deep learning is a form of computational learning in artificial neural networks and machine learning. Deep learning has seen strong progress in recent years. Deep learning comprises several layers of features extracting and transforming each layer using the previous layer data to produce results. Deep learning is the most common methodology and sub-division in machine learning. Deep learning has three major features, such as data, image processing and signal processing. RAM is used to hold a broad variety of data in various formats for training purposes. The Signal is to interpret the machine learning algorithm signals and the graphics processing unit is to execute the task effectively and rapidly. Deep learning architecture consists of a neural network, a deep network with hidden layers for task success. The deep learning layer includes neurons in the neural network. Every neuron attached to separate neurons for the estimated effects of the activity of the neural network consists of a mathematical procedure.[53] The below image shows that differences between simple neural network and deep learning neural network.

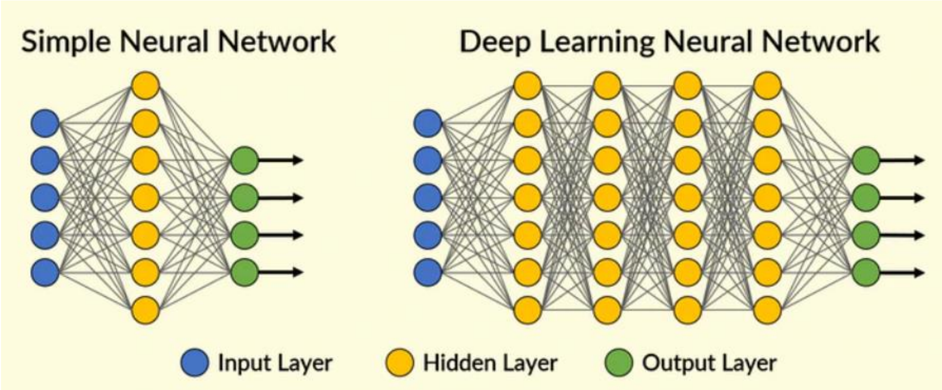


Figure 2.1: Simple Neural Network and Deep learning neural network differences [54].

2.3.1 Neural Networks

Neural network, collection of algorithms that aim to identify fundamental associations in a data set using a mechanism that mimics the way the human brain works. In this way, neural networks apply to neural structures, either biological or artificial in nature. Neural networks may be tailored to the evolving data. Neural networks are a collection of algorithms that simulate human brain operations to identify associations with large quantities of data. They are seen in a broad spectrum of applications in financial services, from forecasts and communications analysis to fraud identification and risk management. The application of neural networks to forecast stock market values differs. Using data input, the model was trained, and the response data produced (information) was produced, so that output results could be the predictor of future performance. Classification and regression are two forms of controlled learning. Unsupervised learning is research, which allows to identify a secret trend or basic framework of data, which may often immediately understand the structure of data. The neural network comprises of layers of entangled nodes. Neural networks are commonly used, with uses for financial processes, market forecasting, banking, data management and product servicing. Neural networks have also achieved mainstream recognition in company applications such as forecasting and marketing analytics tools, analysis of fraud and risk evaluation.[56]

The neural network analyses pricing data and uncovers possibilities for business decision-making dependent on data processing. Networks may discern subtle non-linear interdependencies and trends other methods of theoretical study are not feasible. Analysis reveals that the precision of neural networks in rendering market estimates for stocks varies. Several models forecast the right market values from 50 to 60 per cent of the time, whilst others estimate the precision in 70 per cent in all cases.1 Others have suggested that a 10 per cent increase in performance is what an investor should ask for from the neural network. There will still be data sets and activity groups that can be best evaluated utilizing previously defined algorithms. It is not so much the algorithm that matters; it is the well-prepared input data on the goal predictor that eventually decides the degree of performance of the neural network.[56]

2.3.2 Convolutional Neural Network

Convolutional layers are the main building blocks used in CNN. A convolution is the basic application of a filter to an input that results in the triggering of the filter. Repeated application of the same filter to the input results in an activation chart called a function chart, showing the position and intensity of the function identified in the data, such as an image. The breakthrough of CNN is the capacity to automatically learn large number of filters parallel to the training data set under the constraints of a predictive modeling query, such as image classification. The effect is very precise functionality that can be found everywhere in the input pictures. CNN is a modified type of neural network model built to operate with two-dimensional picture data, although it may be used for one-dimensional and three-dimensional images. The integral component of the CNN is the convolutional layer that gives the network its name. This layer conducts a "convolution" process. In the sense of a CNN, a convolution is a linear process that requires the multiplication of a series of input weights, just like a regular neural network. Since the procedure was developed for two-dimensional information, a multiplication is done between an array of data input and a 2D array of weights, called a filter or a kernel. Usually, the filter is smaller than the input data and the multiplication form used by the buffer-sized input patch and the buffer is the dot element. The dot product is the element-wise multiplication between the buffer-sized area of the data and the output, which is then summed up, often resulting in a single number. The principle of applying convolutional activity to image data is not new or unique to CNN; it is a standard technique used in computer vision. Historically, the filters were crafted by hand by computer vision professionals and were then added to the image to result in a feature map or filter output, allowing the interpretation of the picture smoother in any sense.[55]

2.3.3 Region-based Convolutional Neural Network

RCNN is one of the first large and successful application of CNN to the localization, detection, and segmentation of objects. The approach was demonstrated on benchmark datasets, attaining state-of-the-art results on the VOC-2012 dataset and the ILSVRC-2013 object detection dataset of 200 classes. Their R-CNN model proposed consists of three modules. They are:

Module 1: Proposal for a Region. Generate and extract category proposals for independent regions, e.g. bounding boxes for candidates.

Module 2: The Extractor feature. Extract feature from each candidate region, e.g. through the use of a deep neural convolution network.

Module 3: Grader. Classify functions as a known class, e.g. linear SVM classifier

"Selective search", a computer vision technique is used to propose candidate regions or bounding boxes of potential objects, although the design is flexibility allows for the use of other region proposal algorithms. The models feature extractor was the AlexNet deep CNN which won the ILSVRC-2012 competition for image classification. The CNN output was a 4,096-element vector that describes the content of the image fed for classification to a linear SVM, particularly one SVM is trained for known class each. It is a simple and straightforward application of CNNs to the localization and recognition of object problems. A drawback of the approach is that it is slow, requiring a passing of a CNN-based feature extraction on each of the candidate regions generated by the proposal algorithm for region. This is an issue since the paper describes the model that operates on approximately 2,000 proposed regions per image at test-time.[57]

2.3.4 Fast R CNN

Fast R-CNN, a single model, rather than a pipeline for directly learning and output regions and classifications. The model's architecture takes the photograph as input a set of region proposals passed through a deeply CNN. For extraction of features a pre-trained CNN such as a VGG-16 is used. The CNN output is then interpreted by a fully connected layer, then the model bifurcates into two outputs, one for the class prediction via a Softmax layer, and another for the bounding box with a linear output. For every region of interest in a image given this process is then repeated multiple times. The model is faster to train and predict, but it still requires a set of candidate regions to be proposed together with each input picture.[57]

2.3.5 Faster R-CNN

The architecture was designed as part of the training process to both propose and refine regional proposals, referred to as a Region Proposal Network, or RPN. Then these regions are used in a single model design in concert with a Fast R-CNN model. These improvements both reduce the number of regional proposals and then speed up the model test-time operation to near-real-time. This architecture is a single unified model, it consists of two modules:

Module 1: Proposal Network to the Region. CNN in which regions are proposed and the type of object to consider in the region.

Module 2: Fast R-CNN. CNN designed to extract features from the proposed regions and output the bounding box and class labels.

Both modules act on the same deep CNN output. The region proposal network acts as a mechanism of attention to the Fast R-CNN network, informing the second network about where to look or pay attention to. The RPN works by getting the output of a pre-trained deep CNN, like VGG-16, and passing a small network over the feature map and multiple region proposals and a class prediction. Region proposals are boundary boxes, based on so-called anchor boxes or predefined shapes designed to accelerate and enhance the region's proposal. The class prediction is always binary, indicating the object is present or not, of the proposed region's so-called "objectness". An alternating training procedure is used where both sub-networks are simultaneously trained, though interleaved. This allows simultaneous tailoring or fine tuning of the parameters in the deep CNN feature detector for both tasks.[57]

2.3.6 YOLO

The approach involves a single, trained end-to-end neural network that takes a photo as input and directly predicts boundary boxes and class labels for each bounding box. The technique gives low predictive accuracy although a speed-optimized version of the model operates at 45 frames per second and up to 155 frames per second. The model works by first segregating the input image into a cell grid, where each cell is responsible for predicting a bounding box if it falls within the center of a bounding box. Each grid cell predicts a bounding box that includes the x, y coordinate and the width and height and trust. A prediction of class is based on each cell also.[57]

2.4 Network architecture

Network Architecture refers to the way arrange gadgets and administrations are organized to serve the availability needs of customer gadgets. Network gadgets normally incorporate switches and routers. Sorts of administrations include DHCP and DNS. Customer gadgets contain end-client gadgets, workers, and brilliant things.

2.4.1 Different Network Architecture

Computer networks are worked to serve the necessities of their customers. Portrayed beneath are three basic kinds of enterprise networks:

- Access networks, for grounds and branches, are worked to bring clients and things locally available, for example, associating representatives inside a place of business.
- Networks for server farm associate workers that have information and applications and make them accessible to clients.
- Wide-area networks (WANs) associate clients to applications, some of the time over significant distances, for example, interfacing medical clinic laborers to wellbeing applications.

Today, to serve the demanding needs welcomed on by innovation progressions and computerized change activities, systems are approached to accomplish more. Access networks need to perceive, confirm, and approve client gadgets and brilliant things before welcoming them ready. Server farm systems need to associate applications in different server farms and mists. WANs need to limit expenses and upgrade client experience when serving appropriated applications to dispersed clients. Systems additionally should be dynamic, coordinated, and in lockstep with business needs. Conventional, physically concentrated strategies for overseeing PC systems are ending up being unreasonable. New methodologies are important, ones that require ground-breaking changes in how systems are architected.[58]

2.5 Single Board Computers

A computer that is to be built in a small board that includes microprocessors, memory units, and attached with the input and output pins is said to be a single board computer. These miniature computers have all the features of a computer with less capacity [59]. These are made possible for increasing the capacity of integrated chips that will reduce the overall cost of the computer. These SBC's are mostly used for demonstration purposes in educational institutions [60]. The main advantages of using a single-board computer are the slots of these are to be provided for interconnections that are produced very easily compared to a PC. These are to be used in the process of robotic systems which are often considered as an alternative for the microcontrollers [61]. In an IoT system these SBC's can be used for connecting to the cloud server that can process the data of the virtual machine. An SBC can be connected to the cloud by enabling the SSH tunnel with a use of an X-term. They work with numerous computer languages like Java, Python, C++ etc.

2.6 Cloud Systems

A cloud framework or distributed computing innovation alludes to the processing segments (equipment, programming and foundation) that empower the conveyance of distributed computing administrations, for example, SaaS (programming as an assistance), PaaS (stage as a help) and IaaS (framework as administration) by means of a system (for example the Internet). Cloud framework clients access figuring administrations utilizing internet browsers, which speaks to a processing model that moves the registering outstanding burden to a distant area. Web based email applications are a great representation of a cloud framework that gives a stage to the conveyance electronic informing administrations. Distributed computing is likewise occasionally alluded to as utility processing, since purchaser use of cloud frameworks is metered and charged in a way like a product like water or electric administrations.[62] The Key components include:

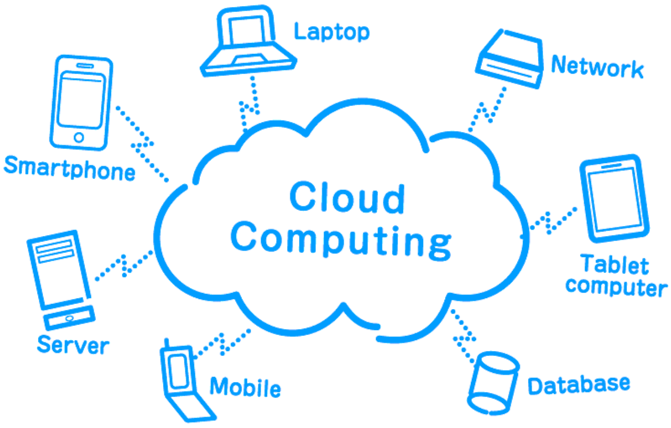


Figure:2.2 Cloud System

There are two key components at the centre of distributed computing innovation, which are: SOA (Service Oriented Architecture) and cloud virtualization, which are depicted in further details below:

SOA Architecture – this component of cloud innovation permits associations to get to cloud based figuring arrangements with highlights that can be adjusted on request, as business needs change. Administration Oriented Architecture permits autonomous web administrations to speak with one another by means of the Internet progressively, giving the adaptability that is needed to quickly reconfigure the administration conveyance for a distributed computing offering. SOA places the duty and expenses of advancement, arrangement and upkeep of web administration segments on the web administrations supplier, which permits a web administrations buyer to get to different web administrations without the cost or overhead that is related with conventional strategies for IT administrations conveyance. SOA is a ground-breaking innovative part of distributed computing since it encourages brought together dissemination and segment reuse, which altogether drives down the expense of programming advancement and conveyance.[62]

Cloud Virtualization is another significant part of a cloud framework that encourages the proficient conveyance of distributed computing administrations. The usage of virtual figuring assets in the cloud, that imitate the usefulness of physical processing assets, fills in as an adaptable burden adjusting the board device that considers the quick change of registering administrations conveyance on request. Virtualization innovation gives associations an apparatus that advances significant levels of accessibility, versatility and dependability as far as the cloud frameworks or distributed computing innovation that an endeavour can admittance to meet its data innovation needs. Virtualization is likewise a significant part of distributed computing innovation for the reasons for catastrophe recuperation and fizzle over help.[62]

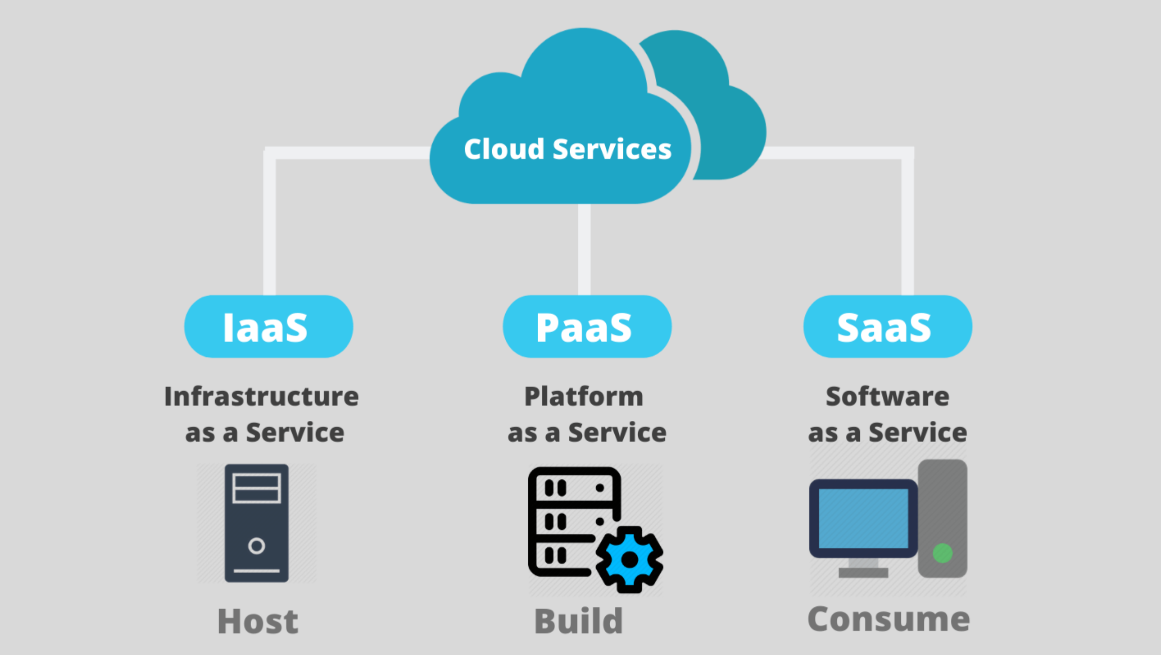


Figure:2.3 Cloud Services

2.7 Data Visualization

2.7.1 Mobile App

Versatile applications permit portable clients (tablets and cell phones) to associate with web applications which were verifiably just gotten to by means of work area or scratch pad computers. Mobile applications like web applications are accessed through an Internet association utilizing a Web Browser on the cell phone for example Safari, Opera and so on. Utilizing online advancements makes the portable application completely cross-stage viable for example it very well may be gotten to utilizing any versatile device. Mobile applications empower clients to utilize web applications on their compact gadgets utilizing the application unquestionably more satisfactory as they are reason worked for littler screens. A portable application might be a Website arranged for use on cell phones, an application, for example, Gmail for versatile, or one of the numerous different applications explicitly worked for clients of versatile phones. Website and entrepreneurs ought to consider at any rate a Website improved for cell phones or a reason manufactured versatile application.[63]

2.7.2 Web App

Web applications are normally coded in program upheld language, for example, JavaScript and HTML as these dialects depend on the program to deliver the executable program. A part of the applications are dynamic, requiring worker side preparing. Others are totally static with no handling required at the worker. The web application requires a web worker to oversee demands from the customer, an application worker to play out the assignments mentioned, and, some of the time, a data set to store the data. Application worker innovation ranges from ASP.NET, ASP and ColdFusion, to PHP and JSP.[64]

A typical web application flow looks like:

- User triggers a solicitation to the web worker over the Internet, either through an internet browser or the application's UI.
- Web worker advances this solicitation to the proper web application worker.
- Web application worker plays out the mentioned task –, for example, questioning the information base or preparing the information – at that point creates the consequences of the mentioned information.
- Web application worker sends results to the web worker with the mentioned data or prepared information.
- Web worker reacts back to the customer with the mentioned data that at that point shows up on the client's presentation.

3 TECHNOLOGY REVIEW

This section describes about the network protocols, software tools and hardware components used in the project.

3.1 Networks Protocols

A network consists of interconnected computers for sharing resources, exchanging data, enabling electronic communications. It is possible to link the computers on a network via wires, telephone lines, radio signals, satellites, or beams of infrared light. A set of procedures followed by the network called network protocols. Network protocols perform the timely, safe, and controlled data or network communication behaviour, policies, and affairs of the end-to-end process. They describe communications laws and conventions. To ensure network\data communication, network protocols must be checked and implemented by the sender and the recipient. It also applies the communication software and hardware nodes on a network. There are many kinds of network protocols.[42]

3.1.1 HTTP

The HTTP is the backbone of worldwide web data exchange. The hypertext is a structured text, using hyperlinks between text-containing nodes. The HTTP is the distributed and interactive hypermedia information system application protocol. The default port of HTTP is 80, and 443 is the protected port.[42]

3.1.2 FTP

The FTP is the most used protocol on the internet and private network file sharing. The default FTP port is 20/21. Utilizing separate data and control connections between client and server, FTP is designed on a client-server model architecture.[42]

3.1.3 TCP/IP

The TCP is the fundamental Internet Communication suite application. It emerged in the implementation of the network under which it replaced the Internet Protocol. Hence the whole suite is generally named TCP / IP. It provides users with a standardized network node addressing scheme and is not reliant on the underlying hardware configuration. The development of logical subnetworks in TCP / IP for network segmentation by the business process is easy. The IP is the main protocol for the data transfer through networks in the Internet protocol series. Historically, in the initial Transmission Control Network, it was the connectionless datagram service; the other, the connection-oriented protocol (TCP). TCP offers a distribution of octets that are stable over an IP network. Its key features are ordering and error reporting. All main Internet services, such as the World Wide Web, email, and transmission of data, are focused on TCP.[42]

3.1.4 UDP

UDP is a protocol to the Transport Layer. UDP is part of the Internet Protocol suite, defined as UDP / IP. Unlike TCP, the protocol is unstable and connectionless. Therefore, no need to create a link before transmitting data. Use UDP for real-time applications such as electronic games, speech, or video communication; UDP enables packets to be dropped instead of processing delayed packets, so high performance is required. While TCP is the dominant transportation layer protocol used by most Internet networks; offers guaranteed distribution, stability, and much more but all these networks cost us extra overhead and latency. UDP pops up in the photo here. UDP does not search for errors because it also saves bandwidth. UDP is more latency and bandwidth efficient.[43]

3.1.5 HTTPS

HTTPS is used to deliver the same functionality as HTTP, but with a protected link that is supported by SSL or TLS. The default HTTPS port is 443. HTTPS uses Data encryption protocol. The protocol is classified as TLS, although it was formerly classified as SSL. This protocol secures correspondence using what is regarded as a shared key asymmetric infrastructure. When a user connects to a webpage, they can submit the webpage via their SSL certificate which contains the public key required to start the protected session. The two machines, the client and the server, then go through a procedure called a handshake from SSL / TLS, which is a sequence of back-and-forth interactions used to create a stable connection.[42]

3.1.6 Selection of Network Protocol

The network protocol used in this project is HTTP. Since the output is displayed in a webpage using Flask Application having cloud-based pipeline, to give request and return response, HTTP is used.

3.2 Single Board Computers

There are many single board computers used in the tech industries. Some of them are explained below

3.2.1 Raspberrypi 1 Model B

The Raspberrypi 1 Model B features include:

- 512 MB RAM
- Ethernet Port
- 2 USB Ports,
- HDMI video output
- Audio - 3.5mm output jack
- 26 pin GPIO header

It has the following advantages:

- Low cost device
- Small Size
- Powerful GPU
- Supports Linux distributions – Debian, Fedora, Arch Linux

It has the following disadvantages:

- Operate at 700MHz
- No WiFi
- No Bluetooth

3.2.2 Banana-Pi

Its features include:

- Allwinner A20 , Cortex A7,1 GHz, Dual core Processor
- 1 GB DDR3 RAM
- 2 USB 2.0 ports
- 2 MicroUSB

- 26 pin GPIO header
- Ethernet
- 3.5mm audio output jack

It has the following advantages:

- Built-in WiFi
- Mass storage via SATA

It has the following disadvantages:

- Setup hardware is harder
- Generates more heat
- High Price

3.2.3 Nano-Pi (A64)

It has the following main features:

- Allwinner A64, 64-bit Quad-core Cortex-A53, 648MHz to 1.15GHz CPU
- 1 GB DDR3 RAM
- One Gigabit Ethernet
- 3.5mm audio output jack
- 802.11b/g/n WiFi
- Onboard IR receiver
- 1 x Micro SD Slot Storage

3.2.4 Selection of Single Board Computer (SBC)

The hardware SBC chosen is Raspberrypi since it is a cheaper compared to other Pi's. It runs on Linux having GPIO pins allowing to control the electronic component for calculation and to explore IoT. This proposed system involves output display in LCD screen which receives data from the cloud through the API connected with Raspberrypi is sent to LCD. This is done using Raspberrypi because it is user friendly and very ease to configure the hardware with the cloud.

3.3 Computer Vision:

Computer Vision is a technique that is used to reconstruct, interrupt and realize a 3D scene from its 2D images, with their structural properties in the scene. It dealt with modelling and replicating human vision using hardware and computer software. It is widely used in Image processing, Pattern Recognition and Photogrammetry.[49] It has different modules like OpenCV, OpenVINO, VisionAI, WatsonVR, etc

3.3.1 Computer Vision Requirements

The major requirement of using computer vision is type of data used. If it is a video or image, it should be dealt in different manner. More the quantity of the video or image used in image processing, more accuracy will be achieved. The initial setup needs more space in the system memory and at least 8GB RAM is required in order to do pre-processing the dataset. [49]

3.3.2 OpenCV

This OpenCV library is mainly used to develop computer vision applications in real time. It is used to read and write images, process images, record and store videos, performing feature detection, detect objects like cars in the videos or images, etc. [49] It has the following Pros. and Cons.

Pros	Cons
Open source	Does not provide good characteristics when compared to MATLAB
Library written in C/C++	Has own flann library causing conflict issues when OpenCV library used with PCL library
Process speed High	-
Low usage of RAM (60 to 70 MB)	-

3.3.3 OpenVINO

This is a toolkit to deploy application faster and solutions the human vision emulation. It is based on CNN extends CV workload across Intel hardware and increase the performance. It includes DLDT which has Deep learning Model Optimizer and Deep learning Inference Engine. The optimizer import, convert and optimizes the models which are trained in frameworks like Tensorflow, Kaldi, etc. The inference engine is a unified API to allow high performance inference on Intel CPU, Intel Integrated Graphics, etc. It functions as a Post-training optimization tool to calibrate model and to execute in INT8 precision.[50]

3.3.4 VisionAI

It is a computer field trains computer to replicate human vision system. The digital devices are enabled to identify and process the objects in videos and images like humans. It provides many options to combine CV models into the websites and applications. Some of them are,

- AutoML Vision Image Classification to train custom models that automatically classify images. Then, define them with the label of own choice and finally deploy in cloud
- AutoML Vision Object Detection to train custom models that detect multiple objects including their position within the image through the bounding box and deploy the model into cloud.
- AutoML Vision Edge Image Classification enables to build high accuracy custom models to classify images at the edge, and initiate real time actions based on local data and finally deploy in cloud
- AutoML Vision Edge Object Detection enables to build quick ad high accuracy custom model to detect many objects in image at the edge and initiate real time actions based on local data and finally deploy in cloud.[51]

3.3.5 WatsonVR

Watson Visual Recognition makes it easy to extract thousands of labels from the images of your organization and detect them out-of-the-box for specific content. It can also build custom models to detect images within our applications for specific content. It has some features which include easy custom training, visually inspect anything and bring AI into iOS applications with ML core. [52]

3.3.6 Selection of CV Library

OpenCV (cv2) library is selected in this project for performing mathematical modelling, image processing and colouring the image are done quickly compared to other object detection library modules. So, the cv2 library is chose for implementation.

3.4 Cloud Service Platform

Cloud Service Provider	Features	Pros	Cons
AWS	Flexible, Reliable, Scalable, Simple to Use, Cost is low. Developed using IaaS, PaaS and SaaS. EC2-VM in the cloud having OS level control. LightSail – tool to deploy automatically,	Provide faster deployments. Easy to use. Cost-effective and can pay only for what we use. Provide hybrid capabilities.	Have common computing issues when moving to cloud Sets default limits on resource from place to place. To get more assistance,need to be opt for paid support
Azure	It is a Microsoft's public cloud computing platform,	Easy to connect with hardware devices.	Lack of integrated backup.

	Provides IaaS, PaaS, SaaS and serverless, Provides range of cloud services	Pay-as-you-go bases charge	Poor GUI Management and tools;
Google Cloud	Provides storage, database, networking services, big data services. Provide function as a service	Better price than other cloud service providers. Improved performance. Redundant backups	Lack of managed services. Poor documentation. Broken SDKs
IBM Cloud	Combines PaaS with IaaS. Can deploy cloud native applications while ensuring workload portability. Support workload in hybrid environment	Provide tools to monitor and manage the cloud deployments. Offers object, file storage for cloud. Includes ML, NLP and visual recognition. Includes IBM IoT to provide service for IoT devices	Lack of security. Control data loss Network**.** Network dependency

3.4.1 Selection of Cloud Platform

Amazon AWS is selected to use in this project. Since its processing speed very high and configurations are very easy compared to other service providers. It has more storage and less cost. Moreover, it is a Pay-only-use service.

3.5 Data Visualization

3.5.1 Web Application Framework

The most used web application frameworks are explained in the below table.

Framework	**Description**	**Specifications**	**Websites using the framework**
Ruby on rails	Can develop 10 times faster than Java framework. Using Model view controller pattern, database driven web app is created	Ruby is the language used. Rails 5.0.0.beta2 is the version	Used by Airbnb, Github, Shopify
Django	Can build quality application. It is developed to ease the creation of web application. Applications created are fast, secure and scalable	Programming language is Python Version is Django 1.9.2	Used by Pinterest, Instagram, etc
ASP.NET	Helps in building robust web applications. Has high performance and lightweight framework for developing web applications	C# is the programming language Version is ASP.NET 5(ASP.NET Core 1.0)	Stackoverflow, GettyImages

Flask	Highly flexibility, Flask documentation is good, Lightweight and modular design	Python Python version 3	Redhat, Netflix

3.5.2 Selection of Web application framework

Flask is selected to use in the project. Because it is simple to code and requesting the response by GET, PUT is very common, and the output displayed is very fast.

3.5.3 Mobile Application Framework

The different mobile application frameworks used to design app is explained in the below table.

Framework	Description	Key Features	Language
React Native	Built by facebook. Used for app creation on both android and iOS. App creation consumes less time. Powerful and reliable.	Maintainability is easy. Component reusable. Hybrid platform framework Javascript	Javascript
Flutter	Built by Google. Builds fast Supportable for both android and iOS phones Single code written	Fast Development UI Flexible Cross Platform Framework Native performance	Dart
Ionic	Open source app development framework, Supports app for iOS and android, Good UI design	Easy to adopt, Hybrid platform network	Angular

Xamarin	Open source app development framework, Supports app for iOS and android	Native API access Easy API integration UI Support native Huge community	C#

3.5.4 Selection of Mobile Application Framework

The framework selected to use in the project is Flutter. Because, using widgets are easier to design and develop. Moreover, all the options are inbuilt since it is a Google product. Performance and speed also will be high.

4 DESIGN

This section explains the design of the research and the flow in which the experiment is being performed. The design is the phase that relates the problem objective and the existing system in a manageable method. In this phase, the system architecture is split into many sub-activities, finally combined to form the organized system.[44]

4.1 Proposed Model

It involves various software and hardware components for estimating the waiting time in the traffic signal. Additionally, the design involves the Yolo model which uses the deep learning algorithm. This algorithm is mainly used to detect real-time objects with high processing speed. Using Yolo model, identifying the real-time objects is a straightforward approach that displays the bounding boxes directly from the frames after processing. It can also be optimized from end to end, based on its performance.[15]

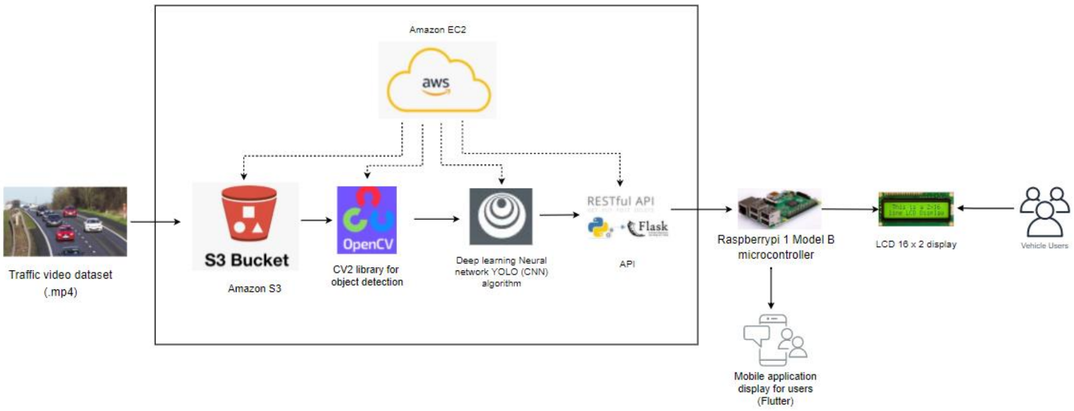


Figure 4.1: Proposed model for estimation of vehicle waiting time in the traffic signal

The data set used in this research is a traffic video that is stored in Amazon S3 bucket storage and the AWS EC2 instance is running on the cloud is connected with Linux operating system in which the python script is written to process the input. The Yolo model consists of the cv2 library which used to solve the Computer Vision problems in modifying the pixel, transforming the objects, optimizing the code, etc.,

Flask Restful API gateway is used to gather the processed data and convert them into a machine-readable JSON format. Then the Raspberrypi collects the data from the API and parses the data to display the output in LCD and to the mobile application for providing traffic signal information to the users.

4.2 Design Methodology

This research work involves AI which uses the Deep Learning Neural Network algorithm used for object detection. Since data analysis plays a vital role in this research so the best design methodology to adopt is CRISP-DM. This methodology is broadly classified into six stages providing the step by step implementation-based approach which is essential in carrying out the research. The figure 4.2 shows the different phases of the CRISP-DM model.[45]

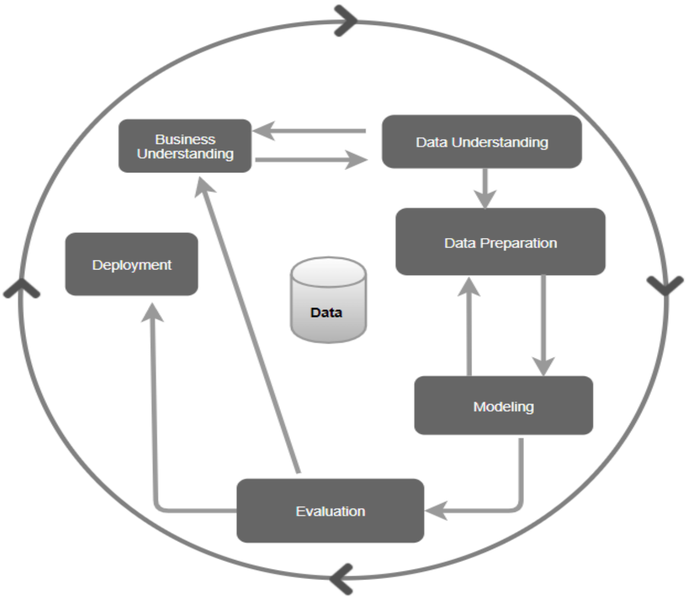


Figure 4.2 : CRISP-DM Methodology

From the above Figure 4.2, the process flow and the dependencies are indicated by the arrows. The six different phases of CRISP-DM methodology include,

1. Business understanding
2. Understanding of Data
3. Preparation of Data
4. Modeling
5. Evaluation
6. Server Deployment

4.2.1 Business Understanding

The initial phase of the design methodology is the business understanding. This helps to carry out the tasks in a business point of view and helps in developing a system to determine the business objective, access situation, and to achieve the goal.

Traffic is a major problem in most of the highly populated urban areas especially in peak hours and it is very tough for the people to predict and travel. If the traffic can be predicted, then the travel can be planned accordingly. The emergency vehicles like Ambulance and Fire service will make use of this system efficiently. The main aim of this research is to detect the number of vehicles in the road standing for the red signal and to calculate the waiting time. This makes the user to avoid valuable time in the traffic and provide information to follow another route where the traffic is minimal. Since AI are emerging rapidly in all sectors, the waiting time predictor implemented in Autonomous vehicle, then it avoids getting stuck in the traffic and follows an alternative route. The patients who are at emergency to save their life get to benefit the most, compared to the normal people because of predicted waiting time in the signal. This system is efficient in terms of time-saving and predicting the traffic accurately and to understand better about the traffic. Through this system, business development can be done excellently since the waiting time is avoided by predicting it earlier.[45]

4.2.2 Data Understanding

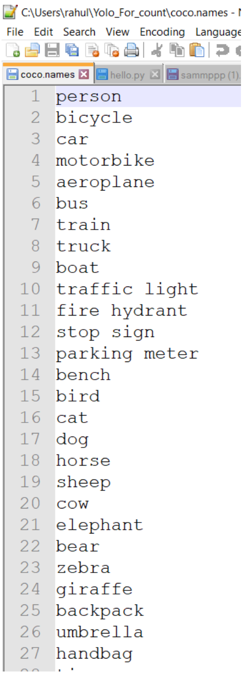
In order to understand the data required for this research, traffic video datasets are needed. The dataset available from Kaggle or other website could not be used for this research. Because in the video, CCTV Camera was placed in an inclined direction at the traffic signal covering vehicles on all the directions. So, while performing image processing using OpenCV, it detects vehicles on all sides thereby it cannot be used in the project. So, the data is collected from the real-time traffic in the city by recording the video focussing the vehicles approaching in one direction towards the traffic signal using the phone camera which captures 60fps. The recorded video is saved in .mp4 by default. The more the length of the video, the more will be accuracy and the result. So, the video is recorded for about 20mins in the city traffic signal intersections.

4.2.3 Data Preparation

The data preparation phase plays a key role in image processing. From the raw input till the final processed video output, data dependencies are very high. The image processing performance is based on the type of algorithm used and the video input. The file size should be minimum to calculate the waiting time quickly. The algorithm going to be used will be having more processing speed. But the conversion of video into frames and processing takes more time. Since the research objective is to process the real-time data having 60fps and the response should be very accurate, and it is used in this project.

4.2.4 Modelling

The most used algorithm for object detection in deep learning is R-CNN but when it receives input video with more number of frames, its processing speed gets slower. It uses a mechanism that once it receives the input data, it creates potential bounding boxes and begin to classify the type of the object. Once the post-processing is done for rebounding the boxes and removes duplicate detections. This pipeline makes the processing slow and displays the output with delay.[46] In order to remove these complications and to make straightforward processing, the Yolo framework is used. Unified detection is the main feature of Yolo and it can process real-time images from 45 fps to 150 fps per second and the output speed response is very high.[15] Yolo uses a predefined dataset named coco.names. It has 80 different trained classes of objects shown in Figure 4.3.



```
person
bicycle
car
motorbike
aeroplane
bus
train
truck
boat
traffic light
fire hydrant
stop sign
parking meter
bench
bird
cat
dog
horse
sheep
cow
elephant
bear
zebra
giraffe
backpack
umbrella
handbag
```

Figure 4.3 : Coco.names file

These trained objects are having good accuracy since yolo.weights and yolo.cfg having the properties to do unified detection. So, when any of those objects are used in the proposed model and compared with objects in input frames, then it displays the bounding boxes with a higher confidence level. The working methodology includes the conversion of video into frames for image processing. There are different stages for identifying the vehicles in each image using Yolo. At first, it takes an input image and then the frameworks divide the input image into grids where each grid image classification and localization are applied.

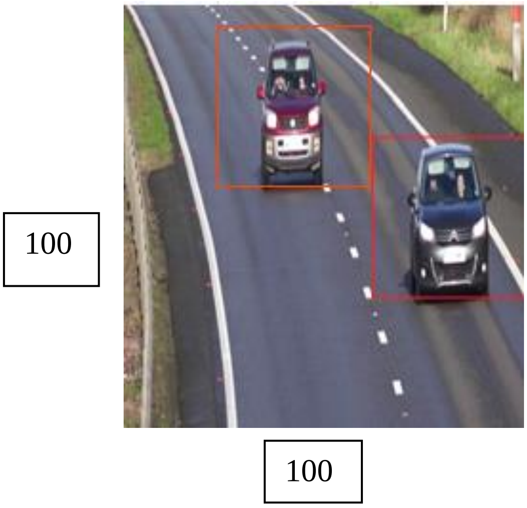


Figure 4.4: Detected vehicle shown

Grids are formed by dividing the input image by Yolo (say a 3x3 grid).

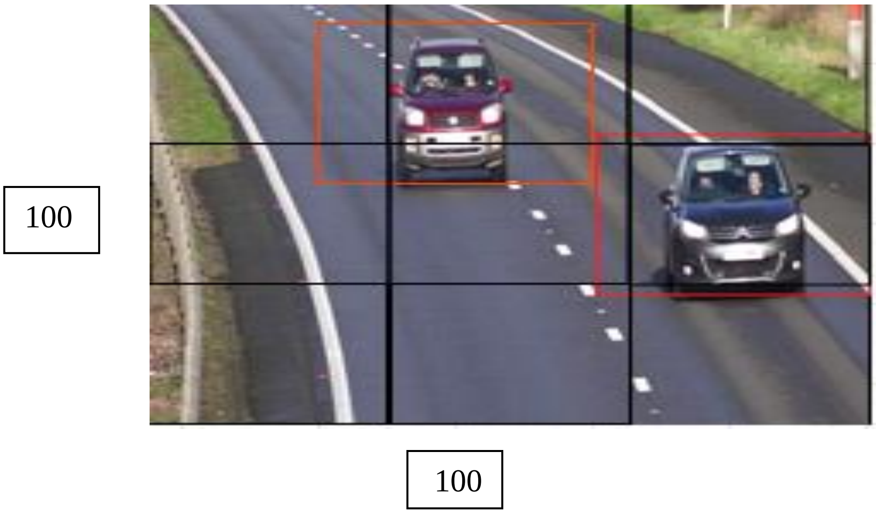


Figure 4.5: Grids formed on the detected vehicle image

Each grid is then applied with image classification and localization. Later, Yolo predicts the bounding boxes and the corresponding class probabilities for objects. To train the model, pass the labelled data to the model. The objects are classified into three classes if the image is divided into a grid of size 3x3. If Pedestrian, Motorcycle are subjected to be classes and so for each grid cell, there will be an 8-D vector named label y.

Y =	pp_c
	Bb_x
	Bb_y
	Bb_h
	Bb_w
	Cc_1
	Cc_2
	Cc_3

Table 4.1 : Y label showing 8 dimensional vector

From the above table,

- pp_c is the probability of an object present in the grid or not
- Bb_x, Bb_y, Bb_h, Bb_w specify the bounding box if there is an object
- Cc_1, Cc_2, and Cc_3 are classes, and if the object is a car, Cc_2 will be 1, and Cc_1 & Cc_3 will be 0 and so on.

For example, if the first grid from the above image is selected,

Figure 4.6: First grid of the image

There is no object in the grid, and the pp_c will be 0. Then, y label for the grid will be,

Y=	0
	?
	?
	?
	?
	?
	?
	?

Table 4.2: Y label showing 8 dimensional vector showing no object in the grid

In the above table, '?' means there is no object in the grid and so the values of Bb_x, Bb_y, Bb_h, Bb_w, Cc_1, Cc_2, and Cc_3 are not considered. Consider another grid containing Car where the class will be Cc_2=1:

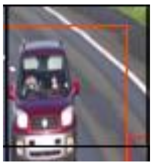


Figure 4.7: Another grid of the image

The Yolo decides whether an object is present in the grid or not before writing y label. In the case of the above image, there are two Objects (Cars), and Yolo will take the mid-point of these objects, and these objects will be assigned to the grid, which contains the mid-point of the objects. Here, the car with the left-centre grid has y label.

Y=	1
	Bb_x
	Bb_y
	Bb_h
	Bb_w
	0
	1
	0

Table 4.3 : Y label showing 8 dimensional vector with values

In the above table, pp_c will be 1 because of the object in this grid. The calculation of Bb_x, Bb_y, Bb_h, and Bb_w will be related to the grid cell. Since the car is second class Cc_2=1, and Cc_1 and Cc_3=0. There will be an eight-dimensional output vector for each of the 9 grids, and the output for this image will have a shape of 3x3x8. The input image and the corresponding target vector are available and using the above example (the input image 100x100x3 and the output 3x3x8), the model will be trained as,

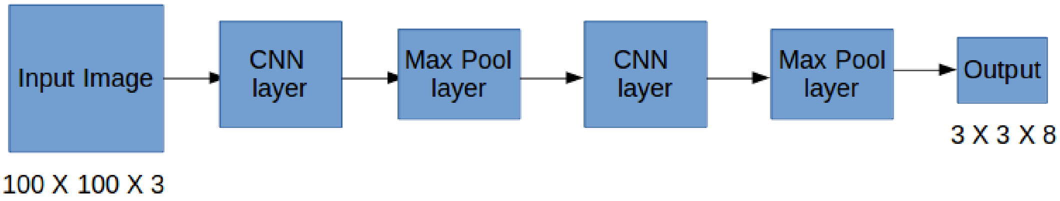


Figure 4.8: Yolo Trained Model

Both forward and backward propagation run will be done to the model during the testing phase, and an image will be passed to the model with forwarding run propagation until getting the label y. A 3x3 grid is explained here to keep things simple, but larger grids 19x19 are practiced in real-world applications. When an object spans out to more than one grid, the grid with the object location mid-point will be assigned. There is also the possibility to reduce the appearance of multiple objects in a single grid by increasing the number of grids (for example, 19x19).[48]

4.2.5 Evaluation

To know whether the predicted bounding box is giving a good or a bad outcome, Intersection over union method is used. This calculates the Intersection over-union of the actual bounding box to the predicted bounding box i.e., it will calculate the area of these two boxes.

Intersection over union (IoU) = Area of the intersection / Area of the union

The prediction will be decided as useful if the IoU is more significant then 0.5, whereas 0.5 is an arbitrary threshold considered, and it will change accordingly to the problems specified. The more the threshold point, the better the prediction of the outcome. Then the testing is done by considering the image during the training period and the number of grids will be considered for the new image as well, and it will divide into the exact number of grids like the training period image. The model in each grid will predict a 3x3x16 shape output. The 16 values look precisely like the one in the training label. The first 8 values correspond to the anchor box, where the first value is the probability of an object in that grid. Values from 2 to 5 will be the bounding box coordinates for that object, and the last 3 values are the class that the objects belong to. The next 8 value corresponds to the anchor box 2, and the same format follows here as well. i.e., first the probability, the next 3 with the bounding box coordinates, and last three for the classes, the object belongs.[48]

4.2.6 Deployment

The entire object detection algorithm is deployed on the Linux machine running on the EC2 instance. The video input gets processed and sends http response through Flask Restful API. The cloud storage obtained has 5 GB storage and when there is a modification in the code has done and the code is deployed. The output can be viewed by giving http URL in web browser.

4.3 System Architecture

This section mainly describes the structure and behaviour of the proposed work. It includes the technical framework, end-user requirements, software, and hardware components used in this research project. The representation of the system is given using a UML diagram. There are two kinds of UML diagrams. They are structure diagrams and behaviour diagrams. Structure diagrams are the diagrams that show the static system structure and its components on various levels of abstraction and implementation, and how they are interrelated. Class diagrams and Package diagrams fall under Structure diagrams. Behaviour diagrams are those that display the dynamic behaviour of the objects in a system, which over time can be represented as a series of system changes. Use case diagram and activity diagrams are some of the Behaviour diagrams.[47] The below architecture diagram Figure 4.9 shows the overall hardware and software components used in the system.

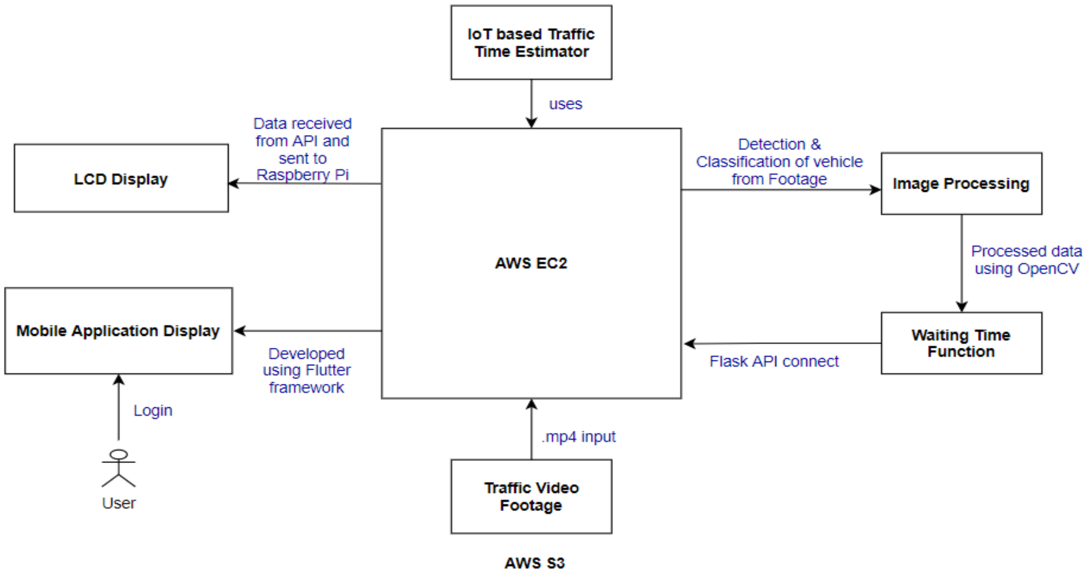

IoT based Traffic Time Estimator
uses
Data received from API and sent to Raspberry Pi
LCD Display
AWS EC2
Detection & Classification of vehicle from Footage
Image Processing
Processed data using OpenCV
Mobile Application Display
Developed using Flutter framework
Waiting Time Function
Flask API connect
Login
User
.mp4 input
Traffic Video Footage
AWS S3

Figure 4.9: Overall System Architecture

Here, the input components are Amazon S3 bucket having video data, the AWS EC2 acts as the main core of the entire system. The object detection algorithm written in Python language is running in this cloud server connect with Linux OS. The video once gets processed, it sends to the Flask to establish the API in a web page. The Flask application is deployed in an Apache server running in Linux OS. Then, the JSON output is used by the flutter mobile application displays the processed output. Similarly, using the HTTP URL the data is sent to the Raspberrypi to display the output in LCD.

4.3.1 Class Diagram

The Class diagram is the blueprint mainly used to create an Object. It provides a software system overview by showing classes, attributes, operations, and relationships.[47] The “trafficoutput”, “imageprocessing”, and the “trafficsignal” are the Classes used in this project as shown in the Figure 4.10. The recording the vehicles, getting the signal time and vehicle position will be done initially. Once it is done, the program gets executed to detect the vehicles, classify the vehicle, and displays the result with bounding boxes. When all the processing and calculations are performed, the output will be displayed in the mobile application and the LCD.

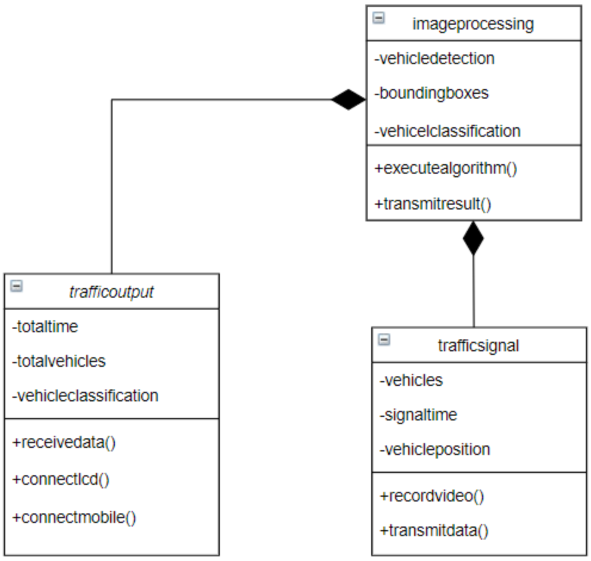


Figure 4.10: Class Diagram

4.3.2 Package Diagram

The Package diagram is used to display the packages and dependencies between the packages.[47] Here, the “numpy” package is used to perform the scientific calculations in a multi-dimensional array, the vehicle waiting time is calculated which is a powerful multidimensional object. The “imutils” package is used to perform the image processing functions such as rotation, translation, and also to display the images using the cv2. The “argparse” is used to obtain the command line arguments in the program where the sequence of arguments is passed to parse the command line. The os package is used to interact with the operating system and its files. The cv2 library is used in the image processing problems and it is very easy to integrate with the other library files. This is used to convert the “numpy” multi-dimensional array values. Both “numpy” and “argparse” are responsible for the algorithm to perform object detection.

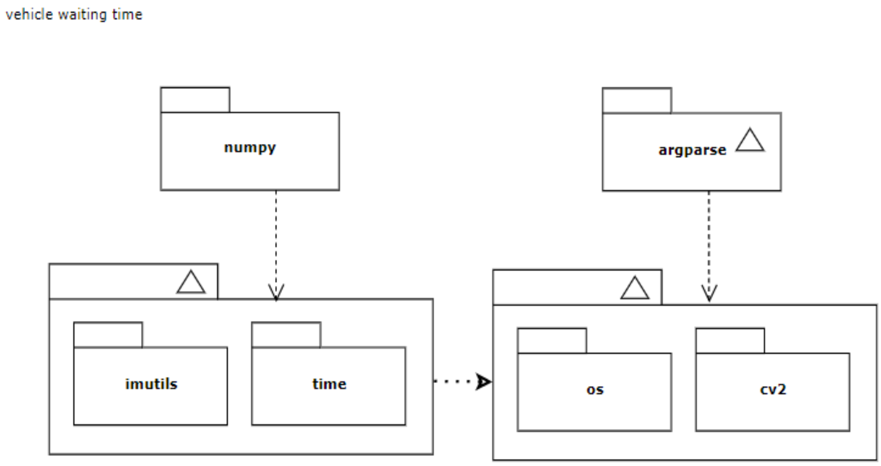


Figure 4.11: Package Diagram

4.3.3 Activity Diagram

The activity diagram is a flowchart to display the workflow in a graphical representation so that everyone can understand it easily.[47] Here, the activity diagram explains about the process of the initial activity of recording the traffic video that is used to perform the image processing if the traffic signal is red else, the process is terminated. The counting of the vehicles is performed based on the signal timer and then the waiting time is calculated. Next, the API is used to link the hardware and software and the output is displayed as total waiting time and the number of vehicles are standing at the traffic signal is displayed in the LCD screen and the mobile application.

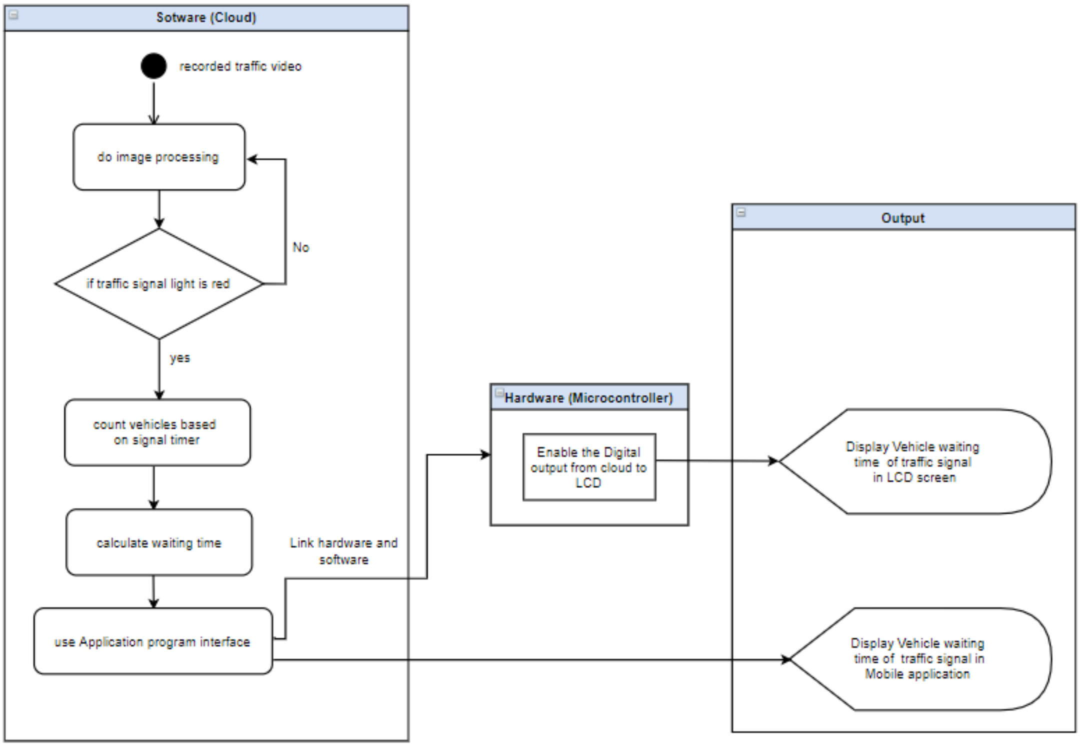

Sotware (Cloud)
recorded traffic video
do image processing
if traffic signal light is red
No
yes
count vehicles based on signal timer
calculate waiting time
Link hardware and software
use Application program interface
Hardware (Microcontroller)
Enable the Digital output from cloud to LCD
Output
Display Vehicle waiting time of traffic signal in LCD screen
Display Vehicle waiting time of traffic signal in Mobile application

Figure 4.12: Activity Diagram

4.3.4 Use case Diagram

The Use case diagram explains the process involved throughout the experiment in each phase and explains the relationship of the user with various users. The first case involves that the traffic red signal indicates all vehicles to stop and in the second case is the video record to identify the traffic and then the image processing is performed to analyse the objects. Furthermore, the vehicle count is calculated, and then, all the data is sent to the cloud platform and the script is executed and the processing is performed. Then finally the notification is showed in the display board and the mobile application.

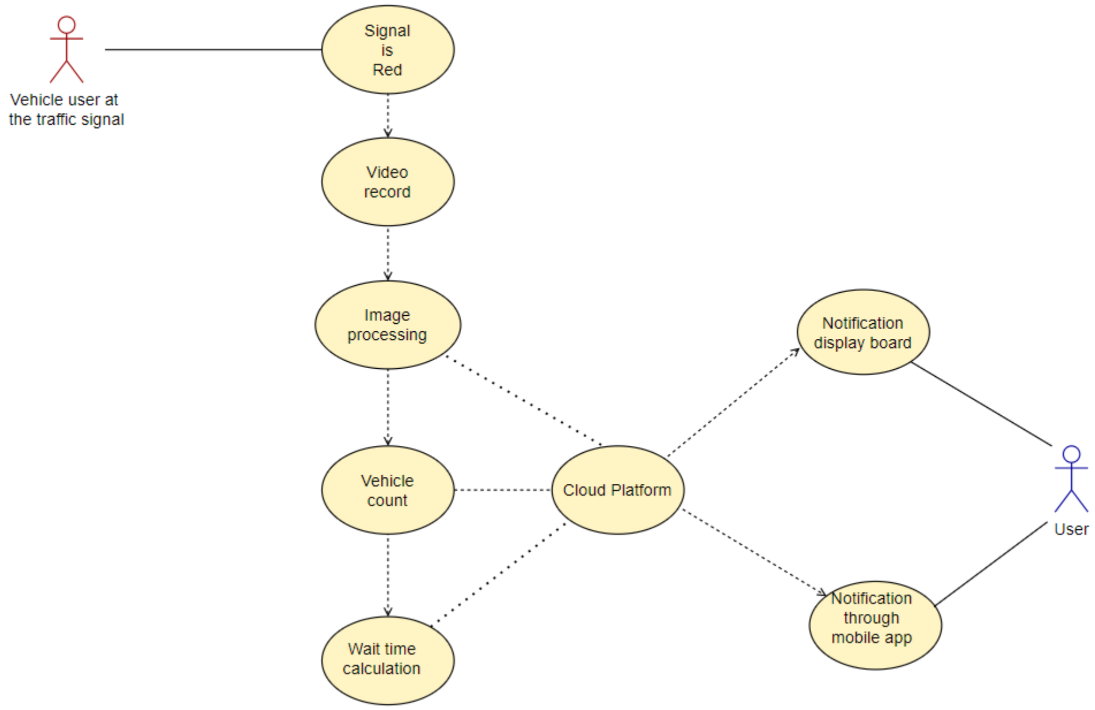


Figure 4.13: Use case Diagram

4.4 Hardware Architecture

The hardware components consist of Raspberrypi 1 Model B kit, connecting wires, 16x2 LCD, variable resistor, and the power supply. The main purposing of using the hardware is to display the output in LCD considering it as Smart Road Sign Display setup in highways. The Raspberrypi has 26pins and 2 USB ports. The python program is written to execute the program to display the strings. The input will be received from API. The Vcc of both Raspberrypi and LCD pins are connected to the power supply and Vss of the same components are connected to the Ground. The other pins are interfaced directly with the Raspberrypi GPIO pins and the LCD pins. The variable resistor used to adjust the brightness of the LCD has 3 pins. The first pin is connected to the Vcc, the middle pin connected to the Pin 15 of LCD, and the third pin is connected to the Ground. The below image shows the hardware architectural diagram of this system.

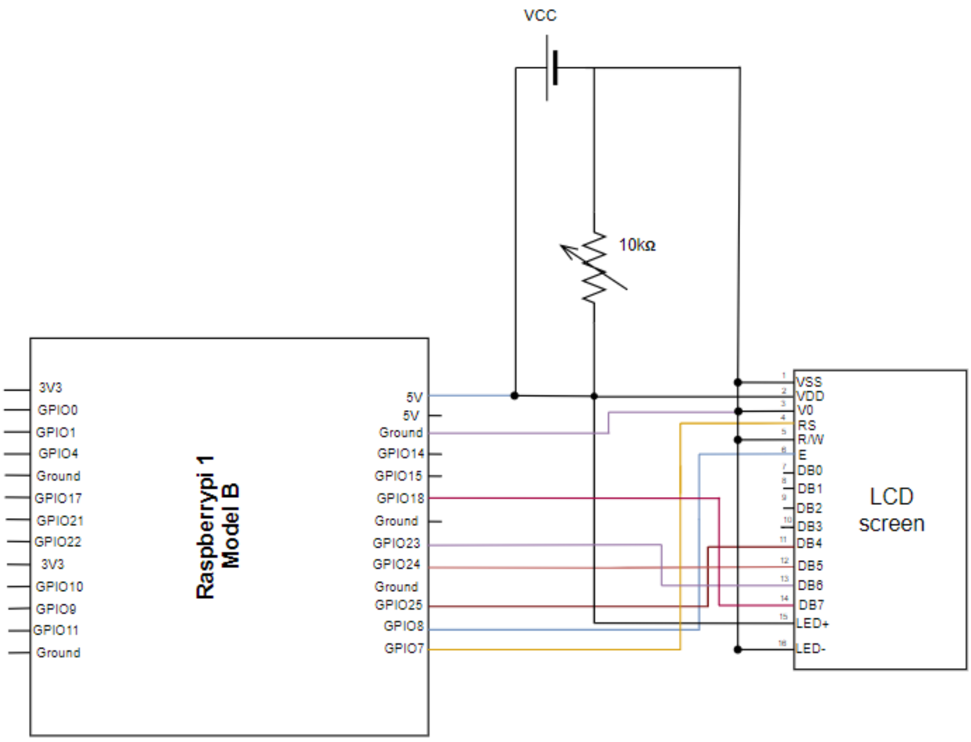


Figure 4.14: Hardware Circuit Design

4.5 Mobile Application Design

The mobile application is designed to display the output to the customer in a simple UI design. It is designed in the Cross-platform Flutter framework using Dart Programming Language. The requirements for the design is known before developing the application. The main advantage of the Flutter framework is it has all inbuilt classes and widgets. The Http call request is received by the "http/http.dart" package. The flutter package "flutter.material.dart" is used to import all the icons and screen designs. The fixed component is known as Stateless widget and the dynamic components are known as Stateful Widget. It is simple to display the Text and icon using these packages and widgets. However, the time conversion is made simple to understand in min and sec and the total number of vehicles is mentioned as integer count. The whole application is developed in

Android Studio IDE and the output is viewed using the Emulator screen. Finally, the application developed for Android and iOS phones in a single written programming code.

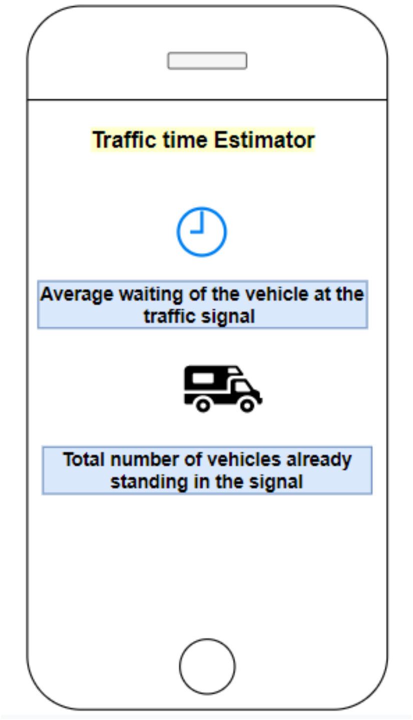

Traffic time Estimator
Average waiting of the vehicle at the traffic signal
Total number of vehicles already standing in the signal

Figure 4.15: Mobile Application Design

5 IMPLEMENTATION

This section describes the implementation techniques used in this project. The implementation is classified into data collection, data processing, and displaying the results. However, in this project, the object detection algorithm (Yolo model) and cloud system contribute a major part in developing the system.

5.1 Implementation to collect input data

The data collection is the main part of the proposed system. Since deep learning concepts are involved, this system acquires data from the real-time environment and is stored in the cloud.

- The input is obtained by placing the phone camera in the traffic signal and recorded the traffic video.
- The camera which has 30 fps is used to record the video.
- In order to perform image processing using the Yolo framework, the video is recorded in .mp4 format.
- The camera is positioned in such a way that all the vehicles on the traffic get captured.
- The vehicles like bicycles, cars, trucks, buses get covered in the video to produce accurate results.

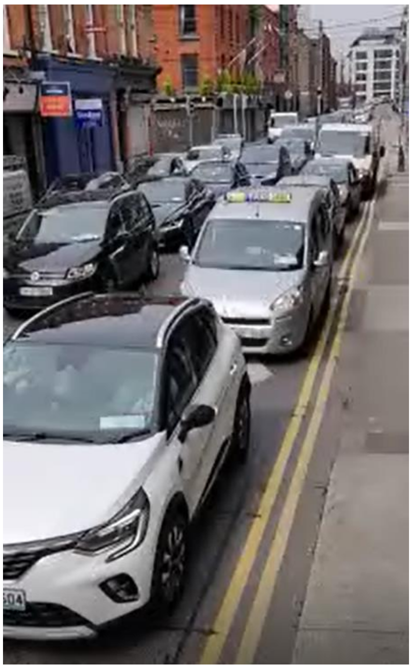
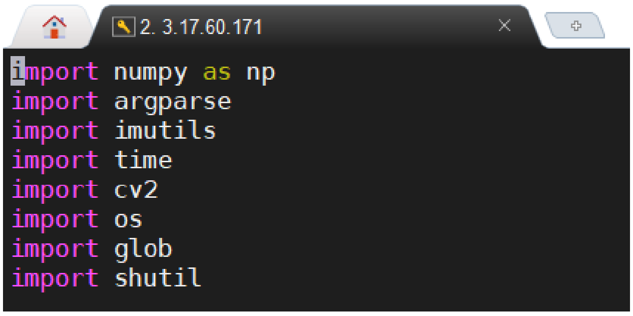


Figure 5.1: Vehicles at a traffic signal

5.2 Deep Learning Setup

Since this system involves video, image processing consumes more processing time. Inorder to minimize the processing time and to get more accuracy in object detection, the Yolo framework is used.

Since it is one of the fastest real-time object detection algorithms compared with other CNN algorithms. Yolo applies a neural network to the entire image in order to predict the bound boxes and their probabilities. Along with these the pre-trained weights of YoloV3, the configuration file, and names file are included. This model is trained for the image size 416*416 which has moderate speed and accuracy. The dataset named coco.names has a list of eighty objects in the Yolo model already has trained objects. The input video given to the system gets compared with the trained object models and produces the required output. In this algorithm, the object refers to vehicles. The programming language used in this algorithm in Python. The python file is created in the Linux machine and deployed in the AWS cloud server. The necessary modules like “cv2, “imutils”, glob, “shutil”, “numpy”, “argparse” and “time” are imported in the programming.



```
import numpy as np
import argparse
import imutils
import time
import cv2
import os
import glob
import shutil
```

Figure 5.2: Modules imported in Python programming

In programming, function get_vehicle_count() is created to detect and count the vehicle from the image and defined in boxes. In the process() function, the Yolov3 weights and configuration file with the help of dnn module of OpenCV is added. The coco.names files that have different object names are trained in this model for the identification. All the objects were stored in a variable called net. Inorder to run the forward pass using the cv2.dnn module, it is essential to send in the names of layers for which the output is computed. The indices of the network output layer is returned by net.getUnconnectedOutLayers() function. The video input placed in a storage is retrieved and assigned to a variable vs for frame conversion. Based on the length of the video, the frames are created and for one second, thirty frames are converted from the video input. Once it is done, to predict the objects with dnn more accurately, pre-process the data by using the blobFromImage() function and perform scaling, channel swap, and mean subtraction. This function accepts frames from the video, model, and produces output layers as parameters. The forward() function of the cv2.dnn module returns a nested list. The list contains all the detected objects in the frame which includes x and y coordinates of the center of the detected object, height h and weight w of the bounding box, scores for all the object classes specified in coco.names. The class which has the highest score is considered as the predicted class. To store the confidence related to each frame object a list called scores are created. Then, using np.argmax() function, the class index with maximum confidence is identified. From all those above functions, the vertices of boxes (predicted bounding box) and “classIDs” (class) is found. Using those bounding boxes are drawn and “object” label is assigned to it. However, in processing the above functions, the bounding boxes with low confidence are removed but there might be some objects detected multiple time and will have more than one bounding box for the object. To avoid these cv2.dnn.NMSBoxes() function is used. Then to select one bounding box, confidence threshold value and NMS threshold value parameters are passed. Once the processing is completed, the types of vehicles are identified and stored in a list.

5.3 Estimate the Waiting time

The type of vehicles stored in the list is used for estimating the vehicle waiting time of the traffic signal.

- When the signal is turned Green, the number of vehicles crossing the signal is counted for 60 seconds and compared with the vehicle speed of bus, car, truck, bicycle for 1 second at the signal.
- The result time is added with the other three sides of the traffic signal timer.
- By sum operation, the estimated wait time of the one direction can be calculated.
- Once all the processing and calculations operations are performed, finally all the frames are merged into output video and get stored in a storage file path.

- The result value is then transformed into API running in the AWS EC2.

5.4 Cloud Environment Setup

Since this project has to be implemented in real-time, the cost of managing and maintaining the normal server will be high. In order to reduce the cost, efficient cloud service provider AWS is used.

- An account is created in the AWS cloud website.
- Initial configuration was made by setting up the storage plan and selecting the region.

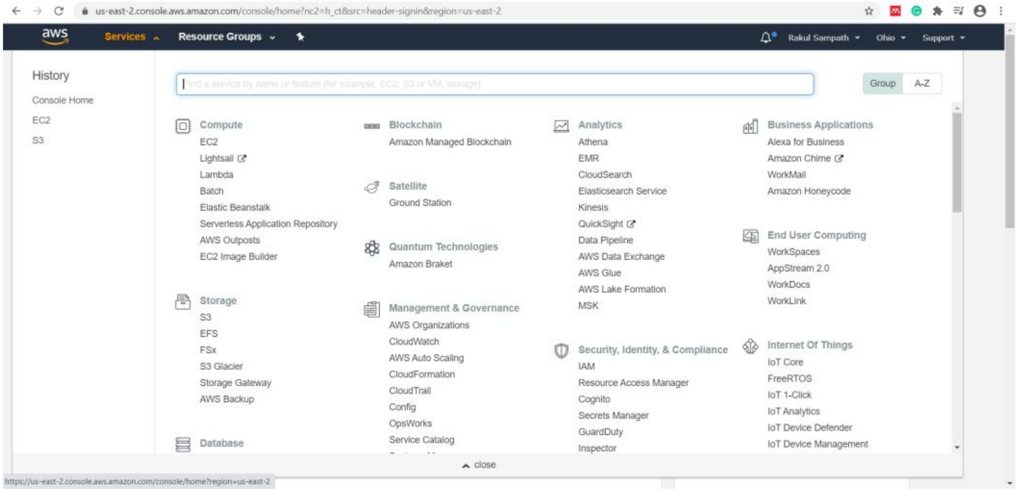
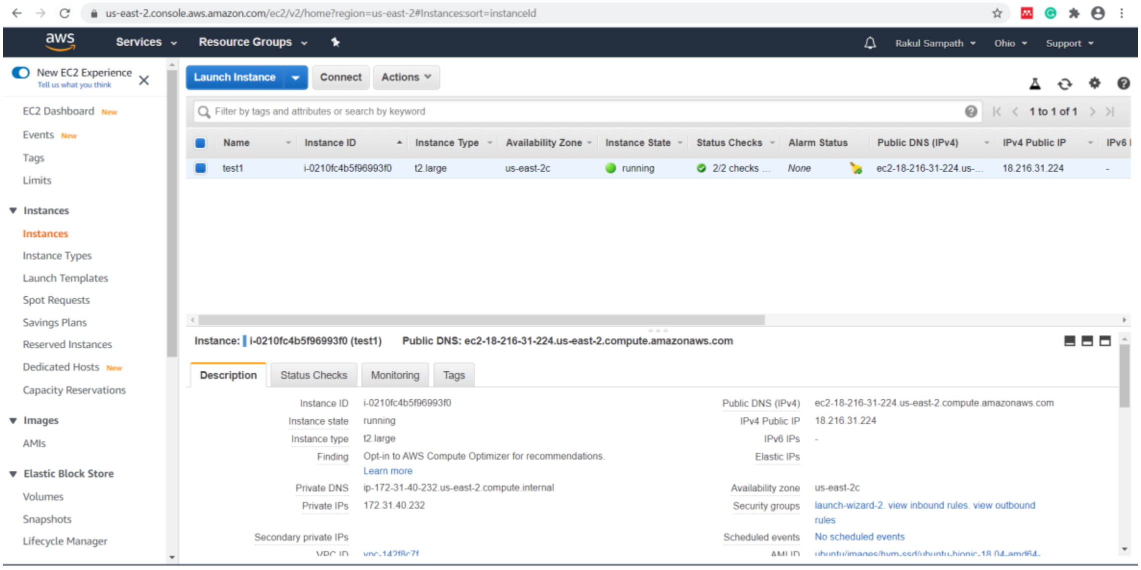


Figure 5.3 : Services tab in AWS

- Under the Services tab, instances are created in EC2.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 I
test1	i-0210fc4b5f96993f0	t2.large	us-east-2c	running	2/2 checks ...	None	ec2-18-216-31-224.us-...	18.216.31.224	-

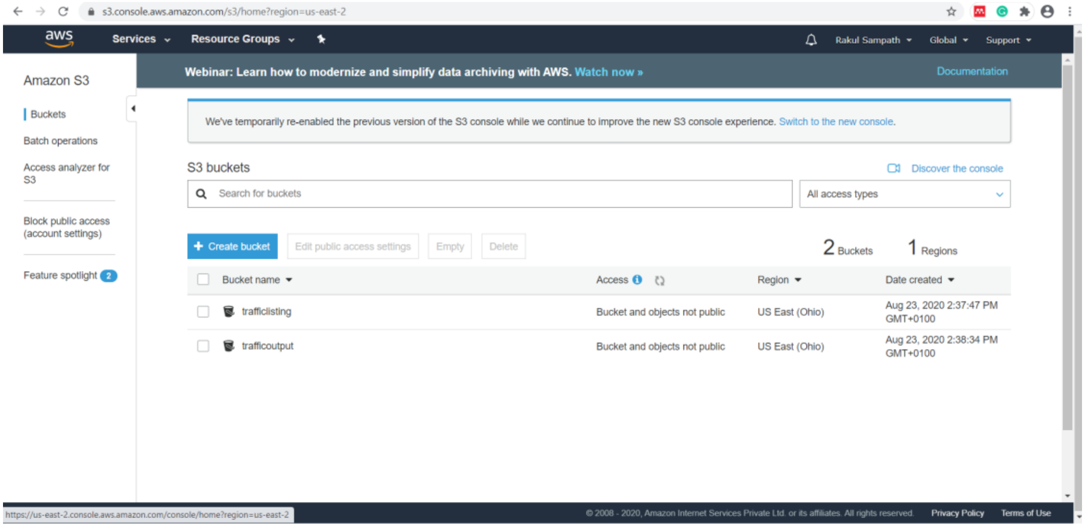
Instance: i-0210fc4b5f96993f0 (test1)   Public DNS: ec2-18-216-31-224.us-east-2.compute.amazonaws.com

Instance ID i-0210fc4b5f96993f0
Instance state running
Instance type t2.large
Finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Private DNS ip-172-31-40-232.us-east-2.compute.internal
Private IPs 172.31.40.232
Secondary private IPs
VPC ID vpc-142f8c7f

Public DNS (IPv4) ec2-18-216-31-224.us-east-2.compute.amazonaws.com
IPv4 Public IP 18.216.31.224
IPv6 IPs -
Elastic IPs
Availability zone us-east-2c
Security groups launch-wizard-2. view inbound rules. view outbound rules
Scheduled events No scheduled events
AMI ID ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-

Figure 5.4: AWS EC2 instance creation

- Then, in the S3 buckets tab, two buckets namely “trafficlisting”, “trafficoutput” are created.
- In “trafficlisting”, the input traffic video is uploaded, and the processed final output was stored in the “trafficoutput” bucket.
- Also, the calculated waiting time and the other vehicle information is stored in the text file output_file storage of EC2 server.


s3.console.aws.amazon.com/s3/home?region=us-east-2
aws
Services
Resource Groups
Rakul Sampath
Global
Support
Amazon S3
Buckets
Batch operations
Access analyzer for S3
Block public access (account settings)
Feature spotlight 2
Documentation
Discover the console
Search for buckets
All access types
Create bucket
Edit public access settings
Empty
Delete
https://us-east-2.console.aws.amazon.com/console/home?region=us-east-2
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Webinar: Learn how to modernize and simplify data archiving with AWS. Watch now »

We've temporarily re-enabled the previous version of the S3 console while we continue to improve the new S3 console experience. Switch to the new console.

S3 buckets

2 Buckets 1 Regions

Bucket name	Access	Region	Date created
trafficlisting	Bucket and objects not public	US East (Ohio)	Aug 23, 2020 2:37:47 PM GMT+0100
trafficoutput	Bucket and objects not public	US East (Ohio)	Aug 23, 2020 2:38:34 PM GMT+0100

Figure 5.5: Amazon S3 buckets

5.5 API Setup

To display the output in the mobile application and LCD display, the data should be transferred from the cloud to the other devices through API.

- API does program requesting services from the EC2.
- It uses the Flask package in the python API program file to create a web application for displaying the processed result in JSON format.

5.6 Develop Mobile Application

To display the real-time traffic update to the user, a mobile application is created.

- In order to develop the application that should be installed in both android and iOS mobile phones, a hybrid mobile application framework Flutter is used.
- The tool required to write the program in Android Studio.
- By installing flutter packages and dependencies, code is written using the Dart programming language.
- Since it is a Google product, it has several features than any other cross-platform framework and ease to implement it.
- The UI components, objects, and methods are mentioned here as Widgets.
- Widgets will be classified as stateful and stateless widgets.
- Stateless widgets are those which are fixed and will not change its state. For example, Background wallpaper.

- Stateful widgets are those which changes its state dynamically. For example, Date and time display.
- Once the application is designed and developed, it is converted into .apk file format which is the required format in android phones.
- It fetches traffic data from the API and display it in this mobile application in a simple manner.

5.7 Hardware Implementation

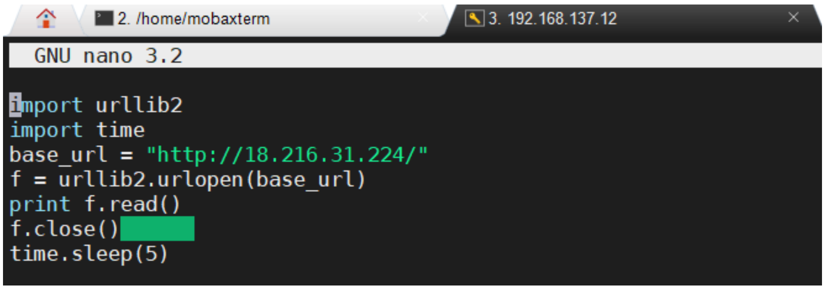
To display the waiting time in the LCD, Raspberrypi needs to be interfaced with the 16x2 LCD kit. Initially, the LCD is fixed on the breadboard. Raspberrypi pins are connected with pins of LCD display and variable resistor. The following LCD Pin is given a connection with the other device pins.

- The Ground Pin or Pin 1 is connected to the ground of the power source.
- The VCC Pin or Pin 2 connected to the supply pin of power source
- The V0 Pin or Pin 3 connected to the ground
- The Register Select Pin or Pin 4 connected to the Raspberrypi GPIO7 pin
- The Read/Write Pin or Pin 5 connected to ground
- The Enable Pin or Pin 6 connected to Raspberrypi GPIO8 pin
- The Pin 11 is connected to Raspberrypi GPIO25 pin
- The Pin 12 is connected to Raspberrypi GPIO24 pin
- The Pin 13 is connected to Raspberrypi GPIO23 pin
- The Pin 14 is connected to Raspberrypi GPIO18 pin

- The Pin 15 is connected to the middle pin of the variable resistor
- The Pin 16 is connected to Ground
- The first pin of the variable resistor is connected to Vcc.
- The third pin of the variable resistor is connected to Ground

Once all the pins are connected to the hardware, a Python program is written to receive the input from API and to display the waiting time in LCD.

- To receive the request, “$sudo pip install requests” command in the Linux terminal is executed
- Then, the python program is created and compiled to display the traffic waiting time result in LCD



```
import urllib2
import time
base_url = "http://18.216.31.224/"
f = urllib2.urlopen(base_url)
print f.read()
f.close()
time.sleep(5)
```

Figure 5.6: Python program to display the waiting time from API

5.8 Result of Implementation

The video input captured is of 500MB which is in MP4 format is given to the S3 Bucket. Once the input is received, the object detection algorithm gets executed and converts the one sec video into 30 frames. Once the vehicle is detected, the algorithm checks for the total number of vehicles in the frame and is stored in a list. Finally, the processed frames are merged into a single output video. The output (.avi) video with the confidence level is displayed on top of the vehicle image frames. This can be viewed by downloading the output file from the S3 bucket. The vehicle type and count are estimated by executing the program running in EC2. The result is stored in a text file. Then, by giving the IPv4 address in the Web page the API displays output in JSON format. Then it transformed to Raspberrypi and the mobile application to show the wait time for the user.

6 TESTING AND RESULTS

In this section, the testing method suitable for this system is manual or eyeball testing is explained briefly. Manual testing is a testing technique which involves writing test cases manually and do not use any automation tools. The following are the different types of tests done for this system.

6.1 Test for confidence level

To test the level of confidence, need to follow the below steps.

6.1.1 Test Scenario

Check whether the confidence level is displayed accurately on top of the bounding boxes of all the traffic videos or not

6.1.2 Pre-requisite

The video which is captured from live traffic is also taken and the traffic video downloaded from the internet is also taken to do the object detection using Yolo Model.

6.1.3 Steps to Execute

Step 1: Place the traffic video1 as an input in the S3 bucket of AWS

Step 2: Download the output video once the process gets completed

Step 3: Check the confidence level is displaying all the vehicles with the bounding boxes or not

Step 4: Place another traffic video2 as an input in S3 bucket of AWS

Step 5: Repeat Steps 2 and 3 by giving different traffic signal videos

6.1.4 Expected Result

Result 1: The confidence level on all the vehicles should be displayed

Result 2: Bounding boxes should be properly aligned on top of the detected vehicle

Result 3: Type of vehicle should be identified properly

Result 4: Vehicle type should be differentiated in various colours.

6.1.5 Actual Result

Result 1: The confidence level on all the vehicles is displayed

Result 2: Bounding boxes are properly aligned on top of the detected vehicle

Result 3: Type of vehicle is identified properly

Result 4: Vehicle type is differentiated in various colours.

6.1.6 Output Image

The below two images are the output for testing the confidence level.

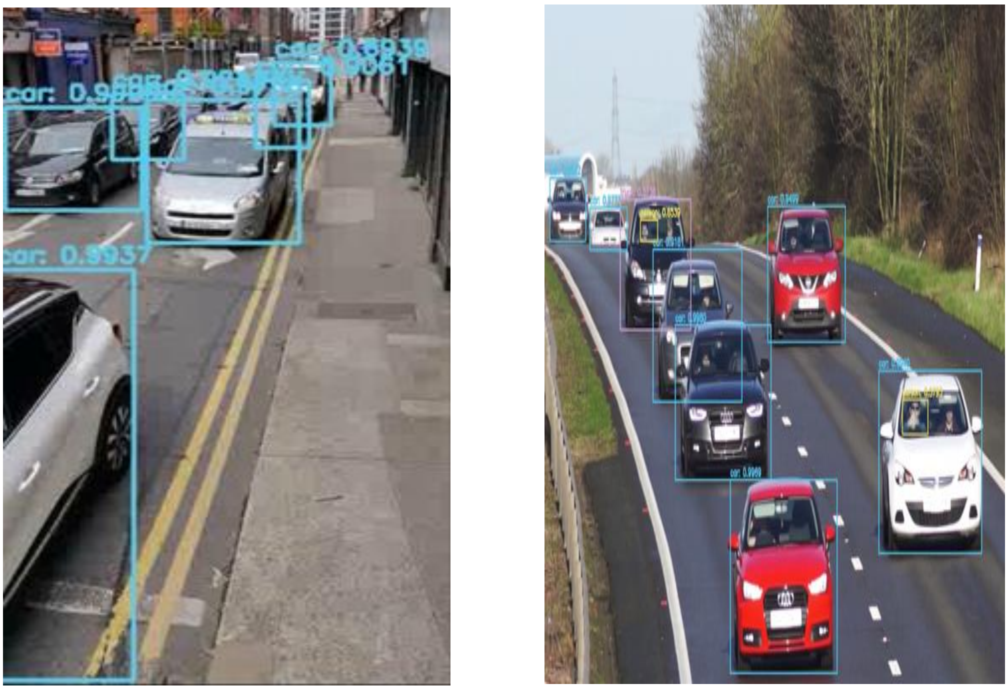


Figure 6.1: Vehicle Confidence level display of two different video inputs

6.2 Test to compare the vehicle count and its type

To compare the estimated vehicle count and classification of the vehicle type, need to follow the below steps.

6.2.1 Test Scenario

Check whether the vehicle count and type is properly done by the OpenCV and Neural network or not

6.2.2 Pre-requisite

The video which is captured from live traffic is also taken and the traffic video downloaded from the internet is also taken to do the vehicle count.

6.2.3 Steps to Execute

Step 1: Place the traffic video1 as an input in the S3 bucket of AWS

Step 2: Download the output video once the process gets completed

Step 3: Play the video for 1 second and pause it for the entire video.

Step 4: Count the number of vehicles manually

Step 5: Observe the type of the vehicles in that video

Step 6: Open new browser tab

Step 7: Enter the IP address to view the result from API

Step 8: Compare the vehicle count and vehicle classification result displayed in t the web page with the manually calculated result

Step 9: Repeat the above procedures by giving different video inputs

6.2.4 Expected Result

Result 1: The vehicle count at the traffic signal calculated manually should match with the vehicle count displayed at the API

Result 2: The classification of vehicles like bus, truck, car, motorbike, bicycle should get displayed correctly

6.2.5 Actual Result

Result 1: The vehicle count at the traffic signal calculated manually should match with the vehicle count displayed at the API

Result 2: The classification of vehicles like bus, truck, car, motorbike, bicycle are displaying correctly

6.2.6 Test Result

The below table displays the output of vehicle count on observation and the output obtained from the API.

Video input	**Vehicle count on observation**	**Vehicle count on API output**
Traffic video 1	73	73
Traffic video 2	40	40

Table 6.1 Display of vehicle count

Similarly, the classification of the below table displays the output of vehicle count on observation and the output obtained from the API.

Video input	**Vehicle classification on observation**	**Vehicle classification on API output**
Traffic video 1	Car:62, Truck:8, Bicycle:3	Car:62, Truck:8, Bicycle:3
Traffic video 2	Car:37, Truck:3	Car:37, Truck:3

Table 6.2 Display of vehicle classification

All the above results are the same while performing the testing manually by observations as well as in the API output.

6.3 Test in Mobile Application

To display the Waiting time at the traffic signal in an efficient manner, the information about the traffic signal result is sent to the mobile application installed on the phone.

6.3.1 Test Scenario

Check whether the application developed is installed in the phone and check the timely update about the traffic is signal is displayed or not

6.3.2 Pre-requisite

Users should install this traffic time to update mobile applications on their phones and they should have internet connectivity. The application is available for both Android and iOS phones.

6.3.3 Steps to Execute

Step 1: Install the flutter application in mobile

Step 2: Click on “Click to view” button in the application

Step 3: Check the traffic waiting time and number of vehicles in the signal

6.3.4 Expected Result

Result 1: The application should be installed properly in mobile phones

Result 2: The traffic waiting time from the API should be shown clearly

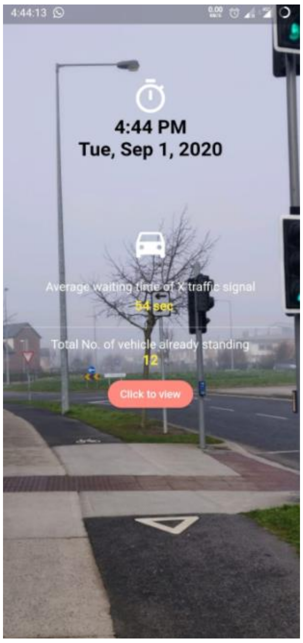
6.3.5 Actual Result

Result 1: The application properly installed in the mobile phone.

Result 2: The waiting time of the traffic signal is displayed correctly in a simple UI

6.3.6 Output display

The below image is the traffic signal information displayed in the traffic signal



4:44:13

4:44 PM
Tue, Sep 1, 2020

Average waiting time of X traffic signal
54 sec

Total No. of vehicle already standing
12

Figure 6.2: Traffic waiting time information displayed in Android phone

6.4 Test in LCD

To display the output in LCD, considering it as Smart Road Sign Display for the user those who avoid using phone while driving and wanted to know the waiting time in the traffic signal.

6.4.1 Test Scenario

Check whether the Raspberrypi is properly receiving the data from the cloud and check the waiting time in the traffic signal is displayed in LCD or not

6.4.2 Pre-requisite

LCD should be connected to the pins of the variable resistor and the Raspberrypi. Then power supply needs to be provided to the hardware components.

6.4.3 Steps to execute

Step 1: Run the Python program

Step 2: Check the output data receiving from the cloud server through API

6.4.4 Expected Result

Step 1: Python program should be executed to receive the input from API without any error

Step 2: Check the output data is properly getting displayed in LCD

6.4.5 Actual Result

Step 1: Python program is executed successfully

Step 2: Check the output data is properly getting displayed in LCD

6.4.6 Output display

The below image is the traffic signal information displayed in the Raspberrypi

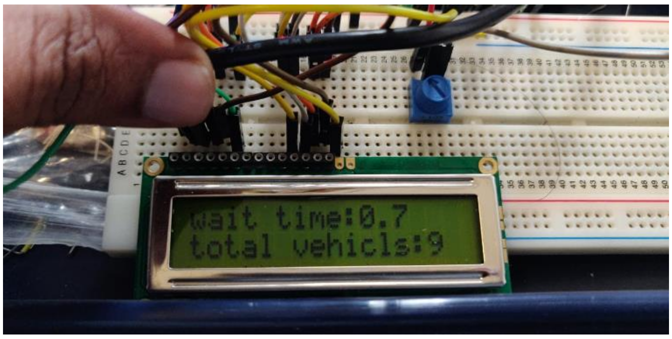


Figure:6.3 LCD displaying output

7 CONCLUSION

The final phase of the thesis is summarized in this section along with the future work of developing and implementing this smart transportation technique in real-time using IoT.

7.1 Project Challenges

There are so many challenges in completing the implementation of this research project. Some of them are explained below.

7.1.1 Data Collection

- The “required” traffic dataset for this research implementation is not available on Internet dataset websites. Because the video required should be a CCTV camera footage placed straightly to the focus one of the four intersection roads. But, the videos are available focusing on the traffic signal junction covering all the intersection roads. While trying to process the video, the vehicles on all the roads are getting counted. The main objective is to make use of video footage which is obtained by focusing the vehicles approaching the signal from one of the roads from the intersection.
- While recording a video from the busy traffic signal, human intervention was high since it is taken by standing on the pavement. Also, the vehicles standing one behind another is not covered in the video. The objective is to take a video from the high position to cover all the vehicles waiting in the traffic signal.
- Since lockdown because of the pandemic virus spread is announced in most of the countries, no vehicles are seen on the busy roads to take the video. Even, after the lockdown gets removed, the traffic on the roads is very low. Because,

to get a better output using Deep learning, more objects (vehicles) should be trained in the model.

- To get the CCTV video footage of the traffic signal from the Traffic Police Department, there are so many formalities to be done since it is a confidential data, that could not be done be in a shorter time
- Finally, the video is taken from the street traffic signal, and only the bicycles, car, and truck approaches in that road, not the buses and motorbikes are made used in this implementation.

7.1.2 Changes in Implementation

- Initially, it is proposed to use a high speed and high transmission range wireless devices in this system, due to Covid-19, ordering the components, and receiving is getting delayed. So, it is not used.
- It is planned to record the video in all the four intersections of the traffic signal junction and then to calculate the total vehicles in each signal and to process. As the video record issues are explained in Section 7.1.1, it could not be able to achieve.

7.2 Summary

The main idea of the project is to provide some achievable solution to real-time problems. The emergency vehicles are often getting stuck in the traffic signal. To resolve the issue, an implementation can be done by using the existing CCTV cameras on the traffic signal used for traffic monitoring. The proposed system recorded the video in the traffic signal and done image processing by using object detection deep learning algorithm to identify the vehicles and to calculate the waiting time of the vehicles in the traffic signal. To process the video and to display the output, the Yolo framework along with the OpenCV module is used.

The whole system is running on the AWS Cloud platform. But, converting the video which has more size and resolution into frames and then to process, it took more time. But, once the image processing is done the result is immediately transferred to the API. From API, the data is obtained and displayed in LCD and the mobile application. The users who want to know about the particular signal average waiting time, then this implementation will be highly useful. By displaying the output in the Smart Road Sign display (considering LCD) then the travelers get to know about the traffic signal wait time update even they do not use mobile. The mobile application is developed to know about the traffic status about the particular signal from any place. The ambulance and fire service vehicles can get a timely update about the signal wait time and change their plan in a different route instead of getting stuck in the signal for a long time.

8 FUTURE WORK

8.1 Project Improvement

- The transmission of video footage should be sent very fast to the cloud system, so high-speed transmitters should be used

- To convert the video into frames quickly, new fps conversion technique need to be implemented

- To increase more processing speed on object detection compared to Yolo, Fast R-CNN and Yolo combination can be tried

- API call response sometimes gets failed when no objects are available, a mechanism to overcome the issue need to be found while processing the video.

- In order to achieve the efficient output, the CCTV camera on the traffic signal should be mounted straight and should focus one of the traffic signal intersections only

- Night time video recording should be made effective by using the night lens in the camera and to record the video by preventing vehicles light reflection

8.2 Future Scope

- The traffic signal waiting time information can be provided to Google Maps as an additional feature

- This system can be implemented in the autonomous vehicles along with GPS and based on the wait time it can choose the shortest or alternative path to reach the destination

Bibliography

[1]. Zhang, M., Yu, T., & Zhai, G. F. (2011). Smart transport system based on "The Internet of Things". In *Applied mechanics and materials* (Vol. 48, pp. 1073-1076). Trans Tech Publications Ltd.

[2]. Sadanandan, L., & Nithin, S. (2017, September). A smart transportation system facilitating on-demand bus and route allocation. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1000-1003). IEEE.

[3]. Satyanarayana, M. S., Mohan, B. M., & Raghavendra, S. N. (2018). Intelligent traffic system to reduce waiting time at traffic signals for vehicle owners. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems* (pp. 281-287). Springer, Singapore.

[4]. Misbahuddin, S., Zubairi, J. A., Saggaf, A., Basuni, J., Sulaiman, A., & Al-Sofi, A. (2015, December). IoT based dynamic road traffic management for smart cities. In *2015 12th International conference on high-capacity optical networks and enabling/emerging technologies (HONET)* (pp. 1-5). IEEE.

[5]. Thakur, T. T., Naik, A., Vatari, S., & Gogate, M. (2016, April). Real time traffic management using Internet of Things. In *2016 International Conference on Communication and Signal Processing (ICCSP)* (pp. 1950-1953). IEEE.

[6]. Javaid, S., Sufian, A., Pervaiz, S., & Tanveer, M. (2018, February). Smart traffic management system using Internet of Things. In *2018 20th International Conference on Advanced Communication Technology (ICACT)* (pp. 393-398). IEEE.

[7]. Pandey, K., & Jalan, P. (2018, December). An Approach for Optimizing the Average Waiting Time for Vehicles at the Traffic Intersection. In *2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)* (pp. 30-35). IEEE.

[8]. Kafash, M., Menhaj, M. B., Sharif, M. J. M., & Maleki, A. (2013, August). Designing fuzzy controller for traffic lights to reduce the length of queues in according to minimize extension of green light time and reduce waiting time. In *2013 13th Iranian Conference on Fuzzy Systems (IFSC)* (pp. 1-6). IEEE.

[9]. Khan, S., Ali, H., Ullah, Z., & Bulbul, M. F. (2018, December). An Intelligent Monitoring System of Vehicles on Highway Traffic. In *2018 12th International Conference on Open Source Systems and Technologies (ICOSST)* (pp. 71-75). IEEE.

[10]. Mao, M., Zhang, Y., Wang, B., Liu, H., & Yin, B. (2016, December). Vehicle detection through traffic video in congested traffic flow. In *2016 6th International Conference on Digital Home (ICDH)* (pp. 247-256). IEEE.

[11]. Saarika, P. S., Sandhya, K., & Sudha, T. (2017, August). Smart transportation system using IoT. In *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)* (pp. 1104-1107). IEEE.

[12]. Hardjono, B., Tjahyadi, H., Widjaja, A. E., & Rhizma, M. G. A. (2017, October). Vehicle travel distance and time prediction using Virtual Detection Zone and CCTV Data. In *2017 IEEE 17th International Conference on Communication Technology (ICCT)* (pp. 1832-1837). IEEE.

[13]. Harahap, E., Wijekoon, J., Purnamasari, P., Darmawan, D., Ceha, R., & Nishi, H. (2018, August). Improving Road Traffic Management by A Model-Based Simulation. In *2018 4th International Conference on Science and Technology (ICST)* (pp. 1-6). IEEE.

[14]. Shi, M. K., Jiang, H., & Li, S. H. (2016, November). An intelligent traffic-flow-based real-time vehicles scheduling algorithm at intersection. In *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)* (pp. 1-5). IEEE.

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

[16]. Mallinckrodt, J. (2010). VCI, a regional volume/capacity index model of urban congestion. *Journal of transportation engineering*, *136*(2), 110-119.

[17]. Johnston, R. A., Lund, J. R., & Craig, P. P. (1995). Capacity-allocation methods for reducing urban traffic congestion. *Journal of Transportation Engineering*, *121*(1), 27-39.

[18]. Maitra, B., Sikdar, P. K., & Dhingra, S. L. (1999). Modeling congestion on urban roads and assessing level of service. *Journal of Transportation Engineering*, *125*(6), 508-514.

[19]. Flynn, M. R., Kasimov, A. R., Nave, J. C., & Seibold, B. (2009). Traffic modeling- phantom traffic jams and traveling jamitons. *Traffic*, *8*(29), 2016.

[20]. Stathopoulos, A., & Karlaftis, M. G. (2001, August). Spectral and cross-spectral analysis of urban traffic flows. In *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585)* (pp. 820-825). IEEE.

[21]. Garib, A., Radwan, A. E., & Al-Deek, H. (1997). Estimating magnitude and duration of incident delays. *Journal of Transportation Engineering*, *123*(6), 459-466.

[22]. Nam, D. H., & Drew, D. R. (1998). Analyzing freeway traffic under congestion: Traffic dynamics approach. *Journal of Transportation Engineering*, *124*(3), 208-212.

[23]. Wang, M. T., & Chou, H. Y. (2003). Risk allocation and risk handling of highway projects in Taiwan. *Journal of management in Engineering*, *19*(2), 60-68.

[24]. Zhang, K. R., & Li, G. F. (2007). A Review of the Relationship between Driving Speed and Traffic Safety. In *International Conference on Transportation Engineering 2007* (pp. 3854-3859).

[25].Benekohal, R. F., & Hashmi, A. M. (1992). Procedures for estimating accident reductions on two-lane highways. *Journal of transportation engineering*, *118*(1), 111-129.

[26]. Sayed, T., & Abdelwahab, W. (1997). Using accident correctability to identify accident-prone locations. *Journal of transportation engineering*, *123*(2), 107-113.

[27]. Smith, B. L., Qin, L., & Venkatanarayana, R. (2003). Characterization of freeway capacity reduction resulting from traffic accidents. *Journal of Transportation Engineering*, *129*(4), 362-368.

[28]. Javaid, S., Sufian, A., Pervaiz, S., & Tanveer, M. (2018, February). Smart traffic management system using Internet of Things. In *2018 20th International Conference on Advanced Communication Technology (ICACT)* (pp. 393-398). IEEE.

[29]. Osman, T., Psyche, S. S., Ferdous, J. S., & Zaman, H. U. (2017, January). Intelligent traffic management system for cross section of roads using computer vision. In *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)* (pp. 1-7). IEEE.

[30]. Jadhav, P., Kelkar, P., Patil, K., & Thorat, S. (2016). Smart traffic control system using image processing. *International Research Journal of Engineering and Technology (IRJET)*, *3*(3), 2395-0056.

[31]. Chowdhury, A. (2016, September). Priority based and secured traffic management system for emergency vehicle using IoT. In *2016 International Conference on Engineering & MIS (ICEMIS)* (pp. 1-6). IEEE.

[32]. Swathi, K., Sivanagaraju, V., Manikanta, A. K. S., & Kumar, S. D. (2016). Traffic density control and accident indicator using WSN. *Traffic*, *2*(04).

[33]. Al-Sakran, H. O. (2015). Intelligent traffic information system based on integration of Internet of Things and Agent technology. *International Journal of Advanced Computer Science and Applications (IJACSA)*, *6*(2), 37-43.

[34]. Mahalank, S. N., Malagund, K. B., & Banakar, R. M. (2016, March). Device to device interaction analysis in IoT based Smart Traffic Management System: An experimental approach. In *2016 Symposium on Colossal Data Analysis and Networking (CDAN)* (pp. 1-6). IEEE.

[35]. Bui, K. H. N., Camacho, D., & Jung, J. E. (2017). Real-time traffic flow management based on inter-object communication: a case study at intersection. *Mobile Networks and Applications*, *22*(4), 613-624.

[36]. Djahel, S., Doolan, R., Muntean, G. M., & Murphy, J. (2014). A communications-oriented perspective on traffic management systems for smart cities: Challenges and innovative approaches. *IEEE Communications Surveys & Tutorials*, *17*(1), 125-151.

[37]. Linganagouda, R., Raju, P., & Patil, A. (2016). Automatic intelligent traffic control system. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, *5*(7), 5902-5906.

[38]. Lakshminarasimhan, M. (2016). Advanced traffic management system using Internet of Things. *Researchgate*.

[39]. Chong, H. F., & Ng, D. W. K. (2016, December). Development of IoT device for traffic management system. In *2016 IEEE Student Conference on Research and Development (SCOReD)* (pp. 1-6). IEEE.

[40]. Xiao, L., & Wang, Z. (2011). Internet of Things: A new application for intelligent traffic monitoring system. *Journal of networks*, *6*(6), 887.

[41]. Andrei, G. R., Serban, T. R., Alexandru, S. V., & Dorin, B. L. (2016, June). Simulation of traffic management systems using arduino boards. In *2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)* (pp. 1-6). IEEE.

[42].Jithin, "Common Network Protocols and Their Ports," 04 08 2016. [Online]. Available: https://www.interserver.net/tips/kb/common-network-protocols-ports/

[43]. Geeksforgeeks, "User Datagram Protocol (UDP)," 12 08 2019. [Online]. Available: https://www.geeksforgeeks.org/user-datagram-protocol-udp/.

[44]. tutorialspoint, "System Analysis & Design - System Design," [Online]. Available: https://www.tutorialspoint.com/system_analysis_and_design/system_design.htm.

[45]. "CRISP-DM: A Practical Guide," [Online]. Available: https://www.datascience-pm.com/crisp-dm-2/#crisp-dm-agile-or-waterfall.

[46]. Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587).

[47]. study.com, “Systems Architecture: Definition & Characteristics,” [Online]. Available: https://study.com/academy/lesson/systems-architecture-definition-characteristics.html.

[48] Analyticsvidhya, “A Practical Guide to Object Detection using the Popular YOLO Framework – Part III (with Python codes),” [Online]. Available: https://www.analyticsvidhya.com/blog/2018/12/practical-guide-object-detection-yolo-framewor-python/.

[49]tutorialspoint.com, “OpenCV - Overview,” [Online]. Available: https://www.tutorialspoint.com/opencv/opencv_overview.htm.

[50]OpenVINO™ Toolkit, “OpenVINO™ Toolkit Documentation,” [Online]. Available: https://docs.openvinotoolkit.org/latest/index.html.

[51]Google Cloud, “Get started with Vision AI,” [Online]. Available: https://cloud.google.com/vision/overview/docs/get-started.

[52]IBM Cloud, “Watson Visual Recognition,” [Online]. Available: https://www.ibm.com/cloud/watson-visual-recognition.

[53] Goodfellow, I., Bengio, Y. and Courville, A. (2016) Deep Learning, Cambridge, Massachusetts; London, England: The MIT Press.

[54] P. C.-I. Ray Bernard, “Deep Leaning to the rescue,” [Online]. Available: https://www.securityinfowatch.com/video-surveillance/video-analytics/article/21069937/deep-learning-to-the-rescue.

[55] “How Do Convolutional Layers Work in Deep Learning Neural Networks?,” [Online]. Available: https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/.

[56]J.CHEN,"Neural," [Online]. Available: https://www.investopedia.com/terms/n/neuralnetwork.asp

[57] J. Brownlee, "A Gentle Introduction to Object Recognition With Deep Learning," [Online]. Available: https://machinelearningmastery.com/object-recognition-with-deep-learning/.

[58]"What Is Network Architecture?," Cisco, [Online]. Available: https://www.cisco.com/c/en/us/solutions/enterprise-networks/what-is-network-architecture.html.

[59]"Single Board Computer," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Single- board_computer.

[60]"WHAT'S A SINGLE BOARD COMPUTER AND WHICH ONE IS RIGHT FOR ME?," [Online]. Available: https://www.circuito.io/blog/single-board-computer/.

[61]"Single Board Computer (SBC)," [Online]. Available: https://www.iotone.com/term/single-board-computer-sbc/t440.

[62]"Cloud Systems and Technologies," Atos, [Online]. Available: https://apprenda.com/library/cloud/cloud-systems-and-technologies/.

[63]"Mobile Apps (Applications)," Storm Web Development, [Online]. Available: https://www.stormweb.ie/mobile-apps.php.

[64] "What is a Web Application?," 31 05 2016. [Online]. Available: https://blog.stackpath.com/web-application/.

Appendices

Python Code – Yolo Framework for traffic estimation time calculation

https://github.com/RakulRS/ThesisCoding/blob/master/yo_ag.py

Python Code – API Connectivity to display the output in webpage

https://github.com/RakulRS/ThesisCoding/blob/master/hello.py

Python Code – Vehicle waiting time information - Raspberrypi LCD output display

https://github.com/RakulRS/ThesisCoding/blob/master/trafficfinal.py

Dart Code – Vehicle waiting time information Mobile app Flutter output display

https://github.com/RakulRS/ThesisCoding/blob/master/main.dart.txt