A Project Report on

Real Time Traffic Management Using Machine Learning

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

in

Information Technology

by

Jyoti Tiwari(16104028)
Ankita Deshmukh(16104031
Gayatri Godepure(17204002)

Under the Guidance of

Dr. Uttam Kolekar
Prof. Kaushiki Upadhyaya

Department of Information Technology
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI

Academic Year 2019-2020

Approval Sheet

This Project Report entitled ***"Real Time Traffic Management Using Machine Learning"*** Submitted by ***"Jyoti Tiwari" (16104028),"Ankita Deshmukh" (16104031), "Gayatri Godepure"(17204002)*** is approved for the partial fulfillment of the requirement for the award of the degree of ***Bachelor of Engineering*** in ***Information Technology*** from ***University of Mumbai***.

Prof. Kaushiki Upadhyaya
Co-Guide

Dr. Uttam Kolekar
Guide

Prof. Kiran Deshpande
Head Of Department of Information Technology

Place:A.P.Shah Institute of Technology, Thane
Date:

CERTIFICATE

This is to certify that the project entitled ***"Real Time Traffic Management Using Machine Learning "*** submitted by ***"Jyoti Tiwari" (16104028),"Ankita Deshmukh" (16104031),"Gayatri Godepure" (17204002)*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Information Technology.***,to the University of Mumbai,is a bonafide work carried out during academic year 2019-2020.

Prof. Kaushiki Upadhyaya
Co-Guide

Dr. Uttam Kolekar
Guide

Prof. Kiran Deshpande
Head Of Department of Information Technology

Dr. Uttam D.Kolekar
Principal

External Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane
Date:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

_____

(Signature)

_____

(Jyoti Tiwari, 16104028)

_____

(Signature)

_____

(Ankita Deshmukh, 16104031)

_____

(Signature)

_____

(Gayatri Godepure, 17204002)

Date:

Abstract

The number of vehicles on the road is increasing day by day and the management of such huge traffic by traditional approach is not sufficient enough. In today's scenario the traditional approach works efficiently only if the count on the road is sparse, as the density of vehicles on a particular side of road increases or if the traffic is comparatively larger on one side than other side in such case the approach fails. Hence we aim to redesign the traffic signal, from static switching to signal which can performs real-time signal monitoring and handling. So in this project the switching time of a signal will be decided on the basis of count of vehicles. Thereafter on the basis of count, switching time will be assigned to different led's which differs every time on the basis of density of traffic and this process will be repeated in a continuous loop. This practice can prove its most effectiveness in releasing the congested traffic at an efficient and faster rate.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

People in today's era usually have tendency of using their own private vehicles for commutation rather than using public or pooled means of transport and this results in large number of private vehicles on road. This endless increasing number of vehicles on road gives rise to many problems amongst them traffic congestion tops in every aspect. In such scenario one cannot restrict individual to limit the usage of their private vehicles but what we can do is at least manage traffic flow in a way that it doesnot alleviate congestion issues.

There are many projects emerging in order to convert the current transport system of cities to 'Smart system' and there are many initiatives under this, one of this is Intelligent Transport System. Many initiatives were taken to design a system that can perform real-time monitoring of traffic signals i.e. the traffic signal switching time will not be predefined o n e, instead the switching time will depend on the count of vehicles on each side of the road. This process of getting the count of vehicle on the road can be achieved using various detection techniques.

Our aim is to design and develop a miniature to depict the current road situation along with monitoring and handling the traffic issues. Hence to proceed with this project we are using a pretrained model YOLO to perform the task of object detection. The pretrained model YOLO uses OpenCV for object detection along with multiple foreground and background subtraction and removal of noise from the input image. The CCTV cameras that are being used for surveillance purpose can be made use to capturing the footage of the road, this image will be passed to the pretrained model as input image. To do so each side of the road will be divided into particular frames of same height and width for capturing the image. The count obtained from the image is the fed to the Raspberry board. As per the count obtained, switching time will be assigned for each side of road. The program will initially check if the count of vehicle in all frame is approximately same then the switching will remain at its predefined regular interval for all sides of signal, the real-time switching for the signal will be performed if the count of vehicles in all frames varies as threshold difference which be provided.

1.1 Objectives

- Solve the socially arising problem of traffic congestion.
- Achieve wide range of transport and environmental objectives.
- To provide the sophisticated control and coordination on traffic.
- This project can prove its most effectiveness in releasing the congested traffic at an efficient and faster rate.

1.2 Problem Statement

With the highly rising traffic congestion all around the world, and it's management by traditional approach are not efficient for smooth commutation purpose hence there is a need to come up with a solution which can be globally accepted and would lead for the better management of traffic. In today's traditional approach the signal switches at its predefined regular interval, but the density of vehicles of the road at every signal doesn't remains the same as shown in fig 3.1, hence the static approach fails. Under such scenario, if the signal remains the same to switch at its regular interval then the side of road which is densely populated will always remain completely packed.

As mentioned in above systems designed till date are to getting vehicle count only, so that comparative study and analysis of traffic can be done. This model can work at its best at condition when traffic reaches to its peak, where the management by an individual becomes difficult. So our aim is to design a model, depicting the real-time traffic scenario and performing signal switching as per our criteria and conditions of threshold value.

Figure 1.1: Solution for current Traffic scenario on roads.

As shown in above diagram, the traffic at LANE 1 is densely populated with vehicles as compared to other three lanes. In normal signal switching method signal switching is done in clockwise manner which result in clogging of vehicles in a particular lane. To overcome the problem of assigning same switching timing for the signal even if the count of vehicles on the road is varying from lane to lane.

1.3 Scope

- The project aims to monitor signal and makes decisions taking real time traffic scenarios into consideration, thus signal switching will be done in a smart and efficient way.
- Traffic Congestion problems at various places will decrease.
- Automatic traffic controlling without human intervention.
- Improve safety on the road network.
- This project can also be implemented on multiple signals.

1.4 Technology Stack

The project can be achieved by using below listed modules.

- Raspberry pi 3
- Pi Camera
- Servo motor
- OpenCV
- YOLO

Chapter 2

Literature Review

The papers refered are mentioned below:

[1] Indrabayu , Rizki Yusliana Bakti , Intan Sari Areni , A. Ais Prayogi "Vehicle Detection and Tracking using Gaussian Mixture Model and Kalman Filter" , 2016 International Conference on Computational In telligence and Cybernetics Makassar , Indonesia , 22-24 November 2016.

In this research object Detection and Tracking, Gaussian Mixture model along with Kalman filter were used to perform the task named object detection, to do so video input from signal were taken for processing. The input video taken as example was by considering both day and night time traffic scenario and its limitations. This research used (.mov) format video as input with frame rate of 25 fps and resolution of 640 x 480. Data was taken from top of a pedestrian bridge with static camera position. The detected image was marked with blob area where the blob area corresponds to the object detected as vehicle. The image detected with blob forms a colorful bounding box around itself representing the detected object, whereas the image without blob area is left unbounded. The system created was validated for vehicle detection using Receiver Characteristic analysis.

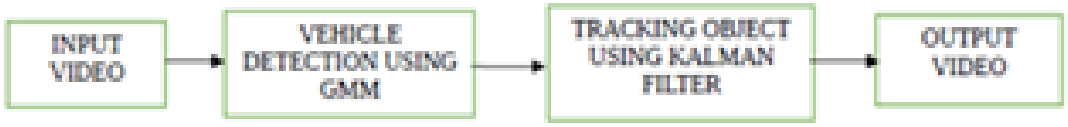

Figure 2.1: Vehicle Detection Tracking.

[2] Li Xun , Nan Kaikai , Liu Yao , Zuo Tao "A Real-Time Traffic De tection Method Based on Improved Kalman Filter", 2018 3rd Interna tional Conference on Robotics and Automation Engineering (ICRAE) Guangzhou , China , 17-19 November 2018.

As the current available extraction methods are not so efficient for processing the traffic basis data, an efficient acquisition method need to be presented in order to process the traffic condition. The method used in this paper is based on improved Kalman filter and gaussian to resolve the conflict of multi-moving vehicle targets detection. Also heuristics improvement

method was applied to improve the efficiency of detection. The method proposed can effectively improve the noise interference and also possess the capability of detecting vehicle from continuous video frame. The main concept presented in this paper was related to no missing, no re-inspection, error detection while detecting the vehicle from the captured images.

[3] Safoora Maqbool , Mehwish Khan , Jawaria -Tahir , Abdul Jalil , Ahmad Ali , Javed Ahmad Vehicle Detection, Tracking and Counting" 2018 IEEE 3rd International Conference on Signal and Image Pro cessing (ICSIP) Shenzhen , China , 13-15 July 2018.

The paper presented mainly focuses on the basic idea of creating an environment that uses low-cost camera and its functioning was on the basis of camera-based algorithm in order to process and control the traffic flow on the road. The vehicle detection form the image captured was done by subtracting the background and foreground images. Tracking of vehicles is done with the help of Kalman. The algorithm proves its efficiency by maintaining its detection accuracy in day as well as night time from the videos acquired from CCTV camera and IR camera. Here, the vehicle detection, counting and tracking was done with the help of computer vision. Also to clearly discern the vehicle from background, BLOB analysis was performed. The model had enlarged its scope in detection and is more flexible in terms of cluster covariance.

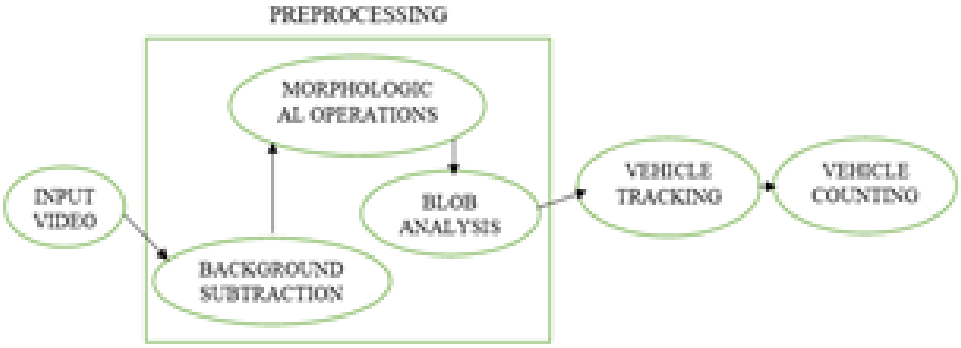

Figure 2.2: Vehicle Locating Counting.

[4] R. Krishnamoorthy , Sethu Manickam "Automated Traffic Moni toring Using Image Vision" ,The 2nd International Conference on In ventive Communication and Computational Technologies (ICICCT 2018) Coimbatore , India , 20-21 April 2018.

The paper proposed an idea of creating an automated system to control the traffic signal and its flow management with the help of multiple CCTV cameras which are connected all over the internet. To execute this task the whole process is divided in to two subtasks: Vehicle Detection System and Traffic Scheduling Algorithm. Vehicle Detection was carried out by using Digital Image Processing and by applying a simple kernel-based Edge Detection and the concepts of Machine Learning was applied to classify the vehicle type into its categories. Scheduling algorithm is designed and optimized by keeping some major points in to consideration such as are low complexity, greater efficiency, and faster process time without compromising on the accuracy of the results. The system designed is extremely efficient and has also proved its beneficiary while using a simple 2MP CCTV camera.

[5] Jess Tyron G. Nodado , Hans Christian P. Morales , Ma Angelica P. Abugan , Jerick L. Olisea , Angelo C. Aralar , Pocholo James M. Loresco "Intelligent Traffic Light System Using Computer Vision with Android Monitoring and Control" , Proceedings of TENCON 2018 - 2018 IEEE Region 10 Conference Jeju.

The system designed was with basic components such as CCTV camera, raspberry pi 3 board and a mobile application. The surveillance camera placed at the junction on the road were used in this project to capture the input data of the traffic signals. The captured images data then sent to the server where the data is prepared before being preprocessed by the microcontroller. The vehicles from the images are detected using raspberry pi 3 microcontrollers with the help of image processing and the result of the traffic condition is send to the android-based application via Bluetooth. The information sent to the mobile user has all information regarding the traffic scenarios whether it is dense, sparse or moderate. The mobile user can switch the traffic light via an app also. Thus an authorized user can view all the traffic patterns on mobile phones and monitor the traffic easily at the intersection point on the road.

2.1 Overview

	Literature Review 1	Literature Review 2	Literature Review 3	Our Paper
Paper Title	Vehicle Detection and Tracking using Gaussian Mixture Model and Kalman Filter	A Real Time Traffic Detection method Based on Improved Kalman Filter	Vehicle Detection, Tracking and Counting	Real Time Traffic Management using Machine Learning
Technology Used	Gaussian Mixture Model, Kalman Filter	Gaussian Mixture Model, Extended Kalman Filter, Heuristic Calculation	Gaussian Mixture Model, Kalman Filter, BLOB Analysis, Hungarian Algorithm	Yolo
Advantage	System Validation for object detection is conducted using ROC analysis, The parameters are Precision and Sensitivity	Detection of Multiple Vehicle targets and reduce the noise disturbance and the foreground blur	System is more efficient and accurate results that are much closer vehicles can detect easily	Smart Traffic Signal Management where system efficiently control traffic flow without creating chaos at road
Drawback	Unable to differentiate between two close vehicles	Cannot handle signal switching for complex signal	No real time prediction	System unable to give preference to Ambulance
Publication Details	Electrical Engineering Study Program, Hasanuddin University Makassar, Indonesia	College of Electric and Information, Xi'an Polytechnical University Xi'an, P.R.China	Department of Electrical and Electronics Engineering, International Islamic University, Islamabad, Pakistan	2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), VIT Tamilnadu

Table 2.1: Overview of paper referred

Chapter 3

Project Design

3.1 Basic Diagrams

Any real-world system is used by different users. The users can be developers, testers, business people, analysts, and many more. Hence, before designing a system, the architecture is made with different perspectives in mind.

The most important part is to visualize the system from the perspective of different viewers. The better we understand the better we can build the system. Hence UML diagrams are used to represent the idea of users with the help of various diagrams. UML diagrams is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.

3.1.1 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

The use case for handling traffic signal management involves various modules and working of it which is represented in diagrammatic way. Use case helps in understanding the function of managing system i.e. what are the role that are being played by managing system. Hence use case provides an easier way for understanding the roles of the actors to the different users involved in project. In this project managing system is the major actor its performs various tasks such as Image capturing, Image processing, Vehicle detection and counting, Passing to Raspberry board etc. Thus different users can understand role of actor by having a glimpse of use case diagram.

Figure 3.1: Use Case Diagram

3.1.2 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

Class diagram in traffic signal management helps in understanding various different modules that are functioning such as raspberry pi module, trained model(how vehicle is

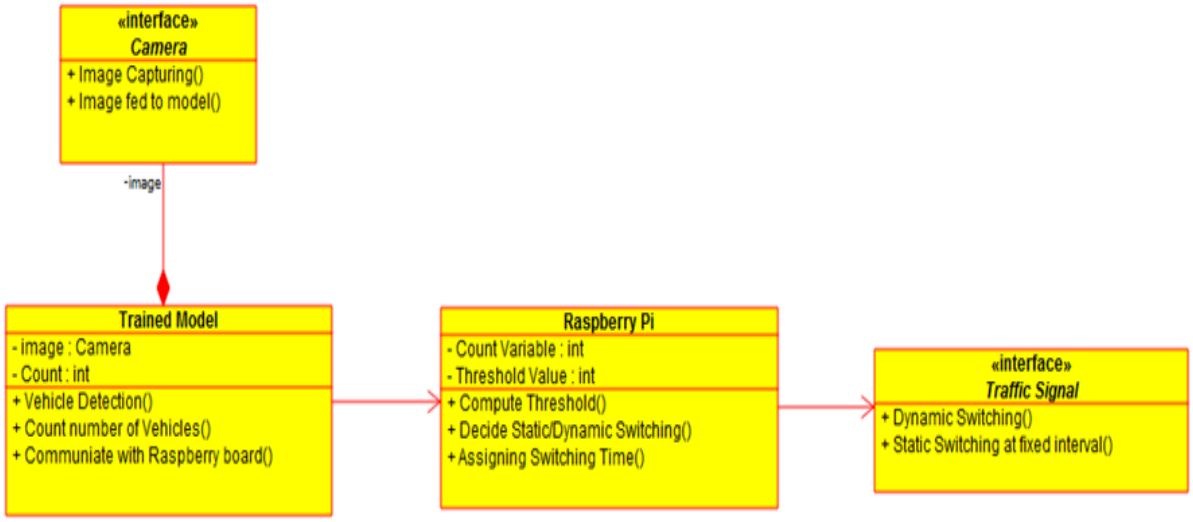

Figure 3.2: Class Diagram

detected using YOLO module). Class diagram not only provide visual overview of project instead it helps in understanding various operations(such as VehicleDetection(), CountNumberOfVehicles() etc.) that a particular module performs. It also helps in understanding various attributes(such as Input image, threshold value etc.) involved in the project.

Class diagram also explains the interfacing components that plays an vital role in functioning of main modules. For example in this project their are two interfacing modules one is camera which provides image as input to trained model module so as to perform object detection and counting. Other interfacing module includes the Traffic Signal switching which decides whether static or dynamic switching should be performed and display the output on the road. Hence we can conclude that class diagram not only provides brief description of project and its module but also provides a layout of executable code along with parameters.

3.2 Flow of Modules

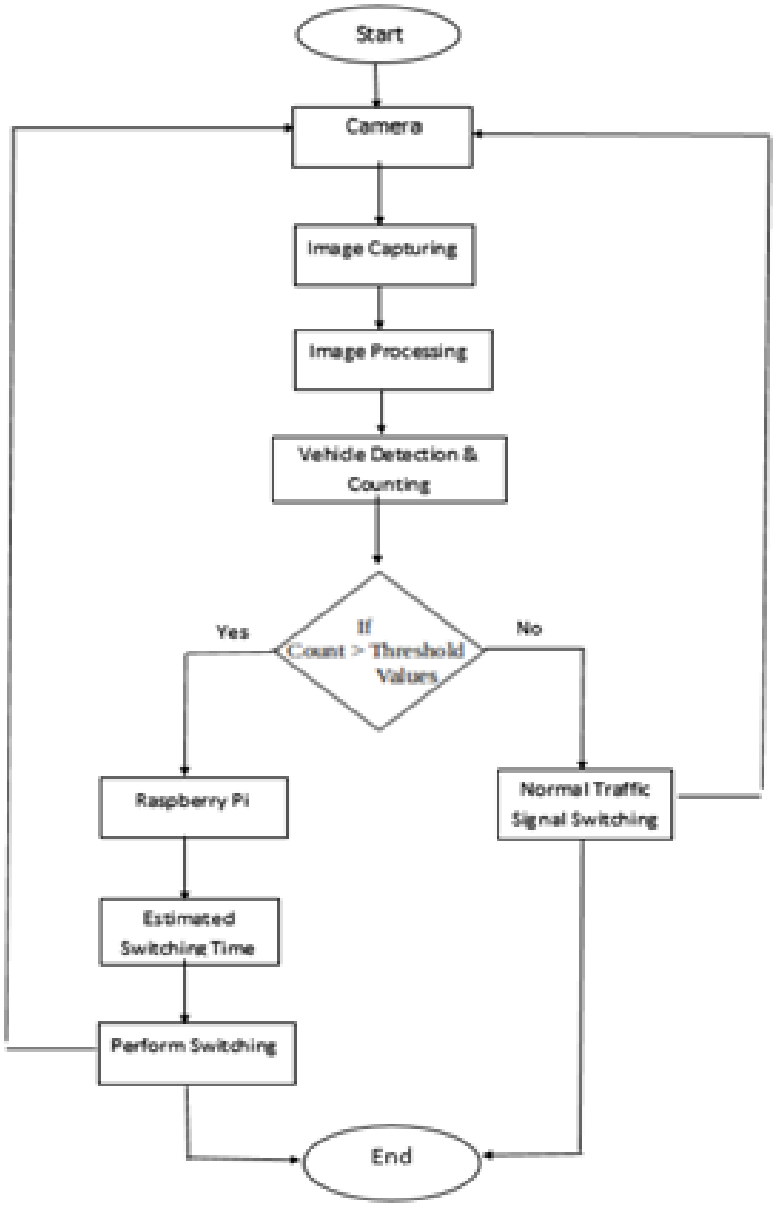

Figure 3.3: Flow of Modules

The project flow is divided into of sequence of activity following one after the another starting from capturing the images at the traffic signals, applying image detection algorithms to detect the vehicles, comparing the count with threshold and finally passing the count to the raspberry board to perform switching.

To make this model working the project is divided into three main sub modules starting from Camera module followed by YOLO module and Raspberry Pi module.

3.2.1 Camera Module on Client's Side

- Camera module used is a pi camera, because of its compatibility and less interfacing with the raspberry board required to make connection.
- One single camera module is fitted on the top of servo/stepper motor so that it can rotate 360 degree and capture image of all sides of road.
- The motor's rotational speed is adjusted in a way that it gives clear images and rotates continuously to capture image.
- The image captured by the camera is then passed to the next module i.e. server machine to perform detection.
- Hence, the task of the camera is to capture images constantly at an fixed interval and pass the images to server by forming a socket between client and server.
- The connection between client and server is done by creating a socket and binding the machines with particular ip address. Thus both client and server machine are binded with some specific ip and ports which remains unique.

3.2.2 YOLO Module on Server Side

- The image captured from camera module is constantly send to object detection module i.e. from client to server machine.
- Image is captured in client OS i.e. in Raspbian OS which is passed to the host system since the host system has more computational power when compared to client's system.
- Further task of vehicle detection and counting is performed on host side and count of vehicle is obtained, this count is then passed to raspberry board to perform the task of switching.
- YOLO is one of the most powerful pretrained and is a combined version of R-CNN and SSD, it performs the task of vehicle detection and extract the count of vehicles.
- R-CNN uses selective search algorithm and proposes accurate bounding box that definitely contains objects and SSD that helps is speed processing of an image
- It divides the image in to M x M grid and calculate confidence and threshold value.
- Confidence score is the score that tells us whether object is present or not.
- YOLO computes its prediction in terms of precision and recall, precision measures how accurate is your predictions and recall measures how good you find all the positives.

3.2.3 Raspberry Pi Module

- The output given by the model is provided as input to the board i.e. the count value.
- The board compares the count obtained from all four sides of the road, after that it computes the difference in count with respect to each other lanes and cross check with the threshold value set manually in the model.

- If difference in vehicle count is less than threshold value then normal switching at predefined regular interval is performed else the raspberry board will to compute and decide different switching time for all signals.
- The switching time decided by the board is then passed to the led and accordingly the led glows for all lanes i.e. either green, red or yellow.
- This process is executed in continuous loop so that every lane's led turns to green atleast once in every cycle for minimum number of seconds.

Chapter 4

Project Implementation

The project aims to provide a solution for current traffic issue by managing traffic signal on the basis of real time scenario. Here a pretrained model YOLO is used to perform the basic task of object detection, and correspondingly the count of the vehicles are stored in order to process further request of signal processing. Also the model is compatible with almost every type of camera, even the cheaper ones including the normal surveillance camera can be used to capture image at an initial level. Now the captured image will be passed to the model for vehicle detection purpose followed by vehicle counting process as shown in figure 4.1.

This whole process of capturing image and detection will be repeated for all four sides of the road using one single camera i.e. pi camera, as pi camera is the most compatible camera for raspberry pi3 board. The camera will be fixed on a rotational motor (servo motor/stepper motor), so that it can rotates 360 degree to capture image from all sides.

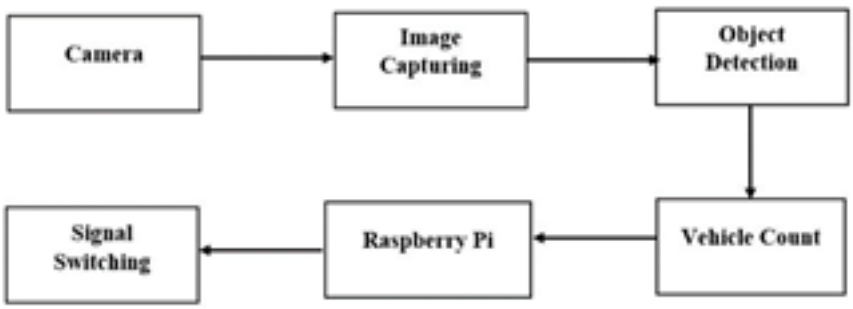

Figure 4.1: Block Diagram

The whole task of capturing and processing is done with the help of two machines i.e. client and server machines. The client machine performs the task of capturing image by pi camera on raspbian OS and passing the image to server machine as shown in Fig 4.3. The server machine runs the object detection library on the image and detect the vehicle and obtain count from the image as shown in Fig 4.2. Client and server communicates with each other by forming a socket between them and binding the socket with the help of IP addresses and port number. Initially to do so both the systems should be connected to same home network so that image being transferred remains secure and transmission can occur between client and server machines.

```
s = socket.socket()
host = '192.168.43.240' #ip of host
port = 12345
print("waiting")
s.bind((host, port))
print("waiting")
s.listen(5)
connection = s.accept()[0].makefile('rb')
print("waiting")
try:
        image_len = struct.unpack('<L',connection.read(struct.calcsize('<L')))[0]
        image_stream = io.BytesIO()
        image_stream.write(connection.read(image_len))
        image_stream.seek(0)
        image = Image.open(image_stream)
        image.show()
        print('Image is verified')
        file_name = str(int(time.time()))
        directory = "sample"
        if not os.path.exists(directory):
            os.makedirs(directory)
        try:
            np.savez(directory + '/' + file_name + '.npz', train=X, train_labels=y)
        except IOError as e:
            print(e)
```

Figure 4.2: Server side socket program

After the camera captures image, Server side's program is initiated first, the server program remains in waiting state until the client system responds to server. Server system initiate socket forming process and bind client machine with specific port of host machine so that image can be transferred from client to server without compromising security. Once the image is send by client the image is verified and saved on server side machine and then the image is passed to image detection modules.

```
s = socket.socket(socket.AF_INET , socket.SOCK_STREAM)
ip = '192.168.43.240'
port = 12345
print("ok")
s.connect(('192.168.43.240', 12345))
print("ok")
connection = s.makefile('wb')
print("ok")
try:
    camera = picamera.PiCamera()
    camera.resolution = (640, 480)
    camera.start_preview()
    time.sleep(2)
    stream = io.BytesIO()
    camera.capture(stream , format='jpeg')
    connection.write(struct.pack('<L', stream.tell()))
    connection.flush()
    stream.seek(0)
    connection.write(stream.read())
```

Figure 4.3: Client side socket program

Client Program checks whether server is in waiting state or not, if yes then the camera captures image at fixed rate and keeps it sending to the server machine. Client machine capture one image and send then waits for few seconds with the help of sleep command and follow this process continuously so that current traffic signal scenario is fed to the model.

```
C:\Users\DELL\Desktop\yolo-object-detection\yolo-object-detection>python -i server.py
waiting
waiting
waiting
Image is verified
```

Figure 4.4: Server's waiting and Image verification

```
pi@raspberrypi:~/Desktop/yolo-object-detection $ python -i client.py
ok
ok
ok
```

Figure 4.5: Client system sending captured image

As shown in above figures, the server is in waiting state until the clients program initiates. As client program starts it captures and sends image to server and the image is saved on host computer. When image is successfully transferred to host then the host system verify the image whether the image is useful and contains valuable vehicle data or not, if not then system discard the image and waits for another image.

```
np.random.seed(42)
COLORS = np.random.randint(0, 255, size=(len(LABELS), 3),
    dtype="uint8")

weightsPath = os.path.sep.join([args["yolo"], "yolov3.weights"])
configPath = os.path.sep.join([args["yolo"], "yolov3.cfg"])

# load our YOLO object detector trained on COCO dataset (80 classes)
print("[INFO] loading YOLO from disk...")
net = cv2.dnn.readNetFromDarknet(configPath, weightsPath)
image = cv2.imread(args["image"])
(H, W) = image.shape[:2]
ln = net.getLayerNames()
ln = [ln[i[0] - 1] for i in net.getUnconnectedOutLayers()]

blob = cv2.dnn.blobFromImage(image, 1 / 255.0, (416, 416),
    swapRB=True, crop=False)
net.setInput(blob)
start = time.time()
layerOutputs = net.forward(ln)
end = time.time()
```

Figure 4.6: Image extraction code

The captured image is then passed to a filter where the region is defined in terms of height and width, only vehicles present in that region are detected and counted. This regions size remains constant for all image being captured. OpenCV is the library that plays important role in object detection, also as and when the object gets detected it forms a rectangular box around the object, so that one can even visually verify that the object detected as vehicle is actually a vehicle only as shown in figure 4.9. For vehicle detection image is passed to darknet classifier where image is read using imread command, image is then converted into blob image by passing through blob classifier. Fig. 4.6 depicts the correctness of image i.e. whether vehicle is present in image or not. To perform vehicle detection various weights and config paths are used to calculate the confidence score of the object detection.

```
# show timing information on YOLO
print("[INFO] YOLO took {:.6f} seconds".format(end - start))

boxes = []
confidences = []
classIDs = []

for output in layerOutputs:
    for detection in output:
        scores = detection[5:]
        classID = np.argmax(scores)
        confidence = scores[classID]

        # filter out weak predictions by ensuring the detected
        # probability is greater than the minimum probability
        if confidence > args["confidence"]:
            box = detection[0:4] * np.array([W, H, W, H])
            (centerX, centerY, width, height) = box.astype("int")

            x = int(centerX - (width / 2))
            y = int(centerY - (height / 2))

            boxes.append([x, y, int(width), int(height)])
            confidences.append(float(confidence))
            classIDs.append(classID)

idxs = cv2.dnn.NMSBoxes(boxes, confidences, args["confidence"],
    args["threshold"])
```

Figure 4.7: Image classification by forming bounding boxes

The model calculates confidence score from the image and if the object is detected successfully than a colored rectangular box is formed around the detected vehicle from the image. The model also calculates number of vehicles present in a captured image so that this count value can be passed further to led's.

Also the model is trained using huge dataset hence it can detect image placed in any random arrangement i.e. it can detect object even if they are rotated in 360 degree. YOLO

is that efficient that it can even distinguish between two very closely placed objects. Unlike traditional approach of applying classifier on each image and making prediction, Yolo look at the image once and but in a clever way. It divides the image into N numbers of partitions and into MxM grid. Now Yolo applies its algorithm one by one in partitions and predict confidence score, confidence score is the scores that tells us whether object is present or not. On the basis of the confidence score Yolo detects an object.

Figure 4.8: Input to the Vehicle Detection Module

Figure 4.9: Output to the Vehicle Detection Module

After image is detected the count of vehicles is displayed on command prompt which will be further passed to raspberry to perform signal switching. The model also calculates time taken to process image and displays to the system. The count obtained from the image obtained from all for side of the road is now passed as input to the raspberry board. The raspberry computes the result by comparing all the count obtained from four different images. The model has some of its fixed threshold value fitted in to it, if the result form the four images is in limit of threshold then simple static switching will be practiced and every signal will be allotted with same switching time, as the traffic on all sides is either sparse or is densely populated by vehicles from all sides. Also in case if the computed result from either of signal crosses the threshold value than dynamic signal switching comes in to action.

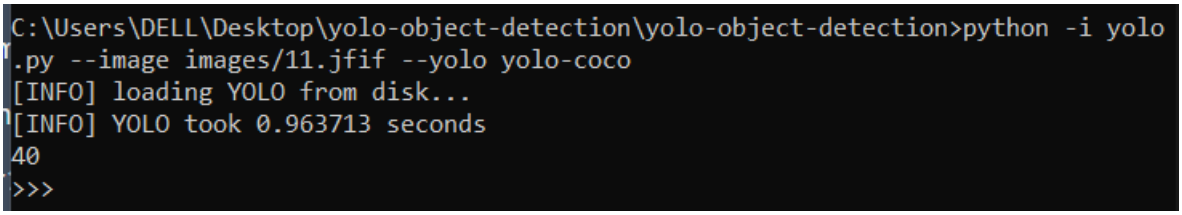

Figure 4.10: Count obtained from image

In dynamic switching the switching time will be assigned on the basis of how densely the road is jammed.

The model is designed keeping the point into consideration that the amount of traffic on all signal does not remains the same then how can threshold value remain the same, hence after few computational processes the model will learn by itself to set threshold value based on how heaped signal is.

The figures 4.8 and 4.9 shows how the camera capture and select image, how further process of detection and counting are taken forward to achieve the ultimate goal of performing real-time signal monitoring and according the signals are handled with the help of raspberry board.

Chapter 5

Result and Testing

5.1 Result

The reason why YOLO is preferres in this project is that Yolo can process many frames with less execution time as compared to other pretrained models. Yolo computes its prediction in terms of precision and recall, precision measures how accurate is your predictions and recall measures how good you find all the positives i.e. how correctly the objects are classified. This can be explained with the help of figures 5.1 and 5.2.

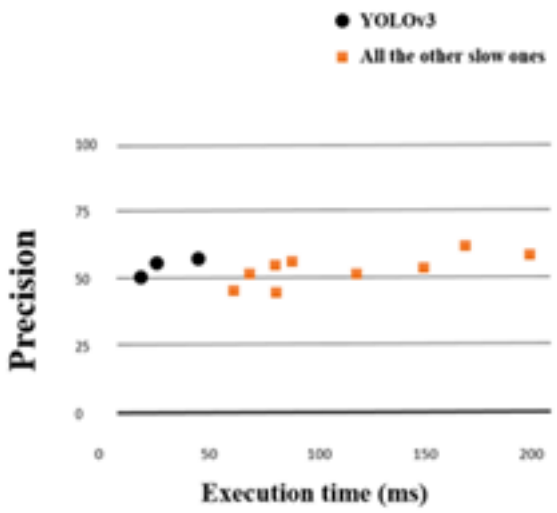

Figure 5.1: Yolo Execution Time Chart

To Test our project we formed two test cases to check working of model.

- Test Case 1:- If any particular lane is more populated by vehicles than the other lanes at a junction when compared with the threshold values, then system will take real time data and the system will work smartly by allotting slightly more switching time to the densely populated signal.
- Test Case 2:- If the population of vehicles is less than the threshold value then traditional traffic signal system will continue.

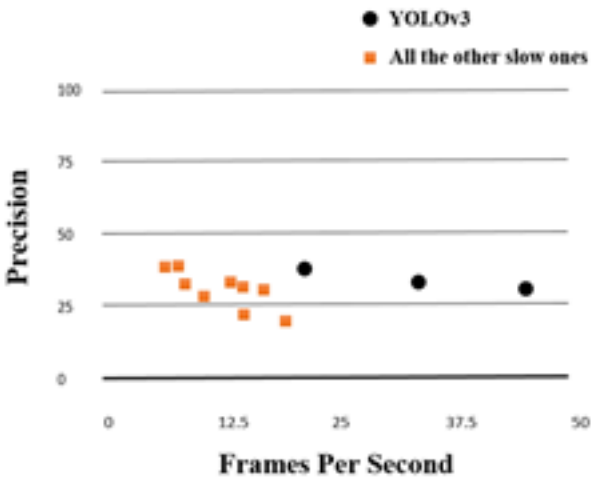

Figure 5.2: Yolo Frames per Second Chart

5.2 Testing

Testing is the process of evaluating a system or its components with the intent to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

Black-box testing is a method of software testing that examines the functionality of an application based on the specifications. It is also known as Specifications based testing. The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.Independent Testing Team usually performs this type of testing during the software testing life cycle.This method of test can be applied to each and every level of software testing such as unit, integration, system and acceptance testing.

In traffic management project, higher testing preference is given to end user experience rather than internal code implementation. In this project, testing at the end user side is more important i.e. how system functions and show its effectiveness after being implanted on signals. Thus Blackbox testing is the best suited testing method for managing Traffic signals. blackbox checks the functionality of system without checking internal code structure, implementation detail or internal paths of the software. This testing is purely based only on software requirement and specifications. It only check whether correct input (i.e. count of vehicles) is provided to the system and correspondingly the system provide correct output (i.e. turning led's green if count is greater then threshold).

Some of the most prominent and blackbox testing methods are as follows:

Functional Testing: This is a type of black-box testing that is based on the specifications of the software that is to be tested. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for.

Non-Functional Testing: This section is based upon testing an application from its non-functional attributes. Non-functional testing involves testing a software from the requirements which are nonfunctional in nature but important such as performance, security, user interface, etc.

Regression Testing: Whenever a change in a software application is made, it is quite possible that other areas within the application have been affected by this change. Regression testing is performed to verify that a fixed bug hasn't resulted in another functionality or business rule violation. The intent of regression testing is to ensure that a change, such as a bug fix should not result in another fault being uncovered in the application. Functional,

Non-Functional and Regression Testing are the testing methods because of which Blackbox Testing becomes best suited for this project, Since performance testing along with spotting a bug and fixing it, is the primary requirement of this project.

5.3 Project Timeline Chart

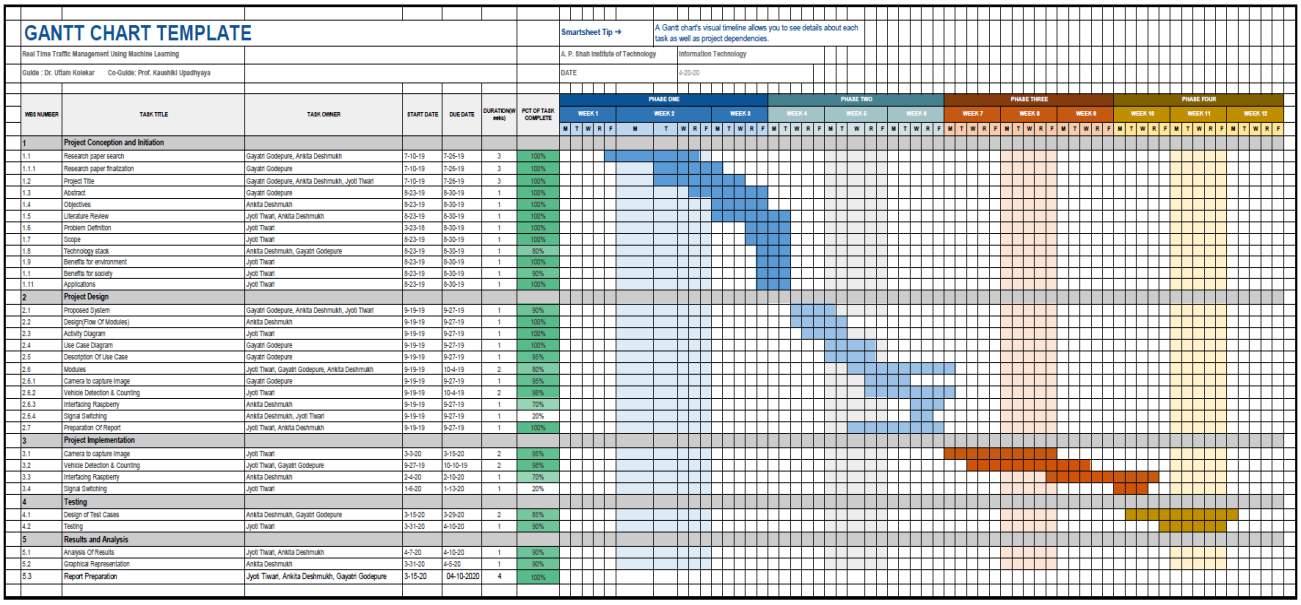

Figure 5.3: Project Timeline

Chapter 6

Conclusions and Future Scope

6.1 Conclusion

Real Time Traffic Signal Monitoring Handling system's aims to fix the problem of traffic which most of the cities in urban as well as rural areas are facing with the help of this project wherein the focus would be to minimize the vehicular congestion virtually without any installation of any kind of hardware. The setup requires camera and raspberry pi board as its hardware requirement and interfacing thereby forming portable medium. The model trained can be used for efficient traffic flow without creating much chaos on the road. The model may take comparatively more training time but the response time will be less. The model is prepared in such a way that it decide smart switching timing for the signal on all sides of the road so the no one has to wait for longer interval of time on the road and flow of traffic is smooth on the road.

6.2 Future Scope

In case of an ambulance stuck in traffic, prior information will be available at signal and accordingly the particular lane will be made free so that ambulance can pass easily even from highly congested areas. Also the aim is to perform pothole detection and report the quality of road to the administration regularly at predefined interval.

6.3 Benefits for Environment and Society

- Minimise vehicular emission on the road.
- Current traffic condition can be analysed to understand the cause of traffic jam and accordingly action required can be taken to reduce jam at signals.
- If traffic is managed in an efficient way then amount of noise and chaos created at signals can be reduced.
- Time saving process where in individuals would not have to wait for signal to turn green after some predefined interval.
- No human interaction required for signal switching and signal can be observed from any remote location.

Bibliography

[1] Indrabayu , Rizki Yusliana Bakti , Intan Sari Areni , A. Ais Prayogi "Vehicle Detection and Tracking using Gaussian Mixture Model and Kalman Filter" , 2016 International Conference on Computational In telligence and Cybernetics Makassar , Indonesia , 22-24 November 2016.

[2] Li Xun , Nan Kaikai , Liu Yao , Zuo Tao "A Real-Time Traffic De tection Method Based on Improved Kalman Filter", 2018 3rd Interna tional Conference on Robotics and Automation Engineering (ICRAE) Guangzhou , China , 17-19 November 2018.

[3] Safoora Maqbool , Mehwish Khan , Jawaria -Tahir , Abdul Jalil , Ahmad Ali , Javed Ahmad Vehicle Detection, Tracking and Counting" 2018 IEEE 3rd International Conference on Signal and Image Pro cessing (ICSIP) Shenzhen , China , 13-15 July 2018.

[4] R. Krishnamoorthy , Sethu Manickam "Automated Traffic Moni toring Using Image Vision" ,The 2nd International Conference on In ventive Communication and Computational Technologies (ICICCT 2018) Coimbatore , India , 20-21 April 2018.

[5] Jess Tyron G. Nodado , Hans Christian P. Morales , Ma Angelica P. Abugan , Jerick L. Olisea , Angelo C. Aralar , Pocholo James M. Loresco "Intelligent Traffic Light System Using Computer Vision with Android Monitoring and Control" , Proceedings of TENCON 2018 - 2018 IEEE Region 10 Conference Jeju.

[6] Sayan Mondal, Alan Yessenbayev, Jahya Burke, Nihar Wahal, "A Survey of Information Acquisition in Neural Object Detection Systems", 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

[7] Joseph Redmon, "YOLOv3: An Incremental Improvement" Ali Farhadi University of Washington.

Acknowledgement

We have great pleasure in presenting the report on **Real Time Traffic Management Using Machine Learning.** We take this opportunity to express our sincere thanks towards our guide **Dr. Uttam Kolekar** & Co-Guide **Prof. Kaushiki Upadhyaya** Department of IT, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Prof. Kiran B. Deshpande** Head of Department,IT, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We thank **Prof. Vishal S. Badgujar** BE project co-ordinator, Department of IT, APSIT for being encouraging throughout the course and for guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Jyoti Tiwari:
16104028:

Ankita Deshmukh:
16104031:

Gayatri Godepure:
17204002:

Publication

Paper entitled **"Real Time Traffic Management Using Machine Learning"** is presented and published at **"2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)"** by **"Jyoti Tiwari","Ankita Deshmukh", "Gayatri Godepure".**
https://ieeexplore.ieee.org/document/9077777