Dissertation on

"Automated Traffic Light Control And Violation Detection System"

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

Bachelor of Technology
in
Computer Science & Engineering

UE17CS490B – Capstone Project Phase - 2

Submitted by:

Ch Abhishek	**PES1201700194**
Mahesh H A	**PES1201701667**
Advaith K Vasisht	**PES1201700207**
Revanth Y	**PES1201700201**

Under the guidance of

Prof. Raghu B A
Associate Professor
PES University

January - May 2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

FACULTY OF ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled

'Automated Traffic Light Control And Violation Detection System'

is a bonafide work carried out by

Ch Abhishek	**PES1201700194**
Mahesh H A	**PES1201701667**
Advaith K Vasisht	**PES1201700207**
Revanth Y	**PES1201700201**

in partial fulfilment for the completion of eighth semester Capstone Project Phase - 2 (UE17CS490B) in the Program of Study - Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period Jan. 2021 – May 2021. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 7th semester academic requirements in respect of project work.

Signature	Signature	Signature
Prof. Raghu B A	**Dr. Shylaja S S**	**Dr. B K Keshava**n
Associate Professor	Chairperson	Dean of Faculty

External Viva

Name of the Examiners	**Signature with Date**
1. ______________________	______________________
2. ______________________	______________________

DECLARATION

We hereby declare that the Capstone Project Phase - 1 entitled **"Automated Traffic Light Control And Violation Detection System"** has been carried out by us under the guidance of **Prof. Raghu B A**, Associate Professor and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester January – May 2021. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

PES1201700194	**Ch Abhishek**	
PES1201701667	**Mahesh H A**	Mahesh
PES1201700207	**Advaith K Vasisht**	
PES1201700201	**Revanth Y**	

ACKNOWLEDGEMENT

I would like to express my gratitude to **Prof. Raghu B A**, Associate Professor, Department of Computer Science and Engineering, PES University, for his continuous guidance, assistance, and encouragement throughout the development of this UE17CS490B - Capstone Project Phase – 2.

I am grateful to the project coordinators, Prof. Silviya Nancy J and Prof. Sunitha R, for organizing, managing, and helping with the entire process.

I take this opportunity to thank Dr. Shylaja S S, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department. I would like to thank Dr. B.K. Keshavan, Dean of Faculty, PES University for his help.

I am deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University for providing to me various opportunities and enlightenment every step of the way. Finally, this project could not have been completed without the continual support and encouragement I have received from my family and friends.

ABSTRACT

With the advent of computational technology, the growth of electronic systems as well as methods is great. With this being said, old problems which can be solved and have been now need to be optimised. This computational energy can be applied to modern transportation system management especially in the road transport sector, which can be made better with the use of image recognition and machine learning algorithms.

The existing systems are first compared and analysed for their drawbacks and advantages in the implementation of that system. A documented list of existing methods to control and manage the road transport sector is made. The problems in the current systems are identified and are noted down as modules. We then analyse the given component in road traffic management and try to optimise it.

In this project, the analysis is performed and a possible solution to a few of those modules are explored and implemented. The exploration takes place in road traffic light and signal management and the solution follows something along the lines of calculating density of vehicles to control the light signal. The second part of the project tries to detect violations of traffic rules without the requirement of an active person trying to monitor it. All of the above mentioned problems have been automated. The whole point of this project is to optimise so the necessity being that we reduce workload on human resources and increase efficiency.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER – 1

INTRODUCTION

The density of vehicles on roads is rapidly increasing which has given rise to problems like accidents, traffic and bottlenecks. As we all know, traffic lights can only be controlled by advanced technologies as it requires a lot of surveillance and monitoring to control the traffic. Due to the large number of these traffic signals it is impossible to monitor all of them. Hence we need to make an automated traffic light system that enables the flow of traffic as and when required by it. Road safety is a major concern by traffic as many rides fail to maintain all the traffic rules and they continue to violate them. Due to this the policeman is facing various problems as they cannot monitor every single violation. To help solve this problem, we need an automated system which can at the very least help the policeman and the government to curb this problem. We propose to build a system which can monitor and report potential violations.

The basic methodology of our project is that we use image processing techniques for all the image recognition and measurement, the density of vehicles, speed of vehicles can be computed. Capturing of images of moving vehicles can also be done.

Our proposed system aims to achieve the following features,
To automate and control the traffic-lights system by setting a timer.

- To allocate time for each signal based on density.
- To follow a specific scheduling order for traffic lights.
- Detection of helmet when violation occurs.
- To detect the vehicles whenever they cross the line.
- To detect the number plate when a potential violation is incurred.

CHAPTER - 2

PROBLEM DEFINITION

People travel to their workplace, go to school and also for some other events daily which gives rise to the increase in traffic on a daily basis. For this reason, Many emergency vehicles get delayed and are made to wait for a long time on roads. In many metropolitan cities, the increase in traffic is a major concern these days. Most of the traffic light control systems are operating on a timing mechanism i.e. for a given duration of time which controls the lights and changes it for every fixed duration. The present day traffic system is in a loop system where all the traffic lights are pre timed. Hence to solve these problems it is better to build a new traffic control system i.e. an automated and intelligent control system.

As we all know that the traffic is increasing rapidly. For this reason people must always wait for longer durations in peak hours to reach their destinations. Because of this, people are getting impatient and violating the traffic signals.

Some People due to their negligence, do not wear their helmets which can cost their lives. So there is a need for a proper system to detect these traffic rule violators and bring them to justice.

CHAPTER - 3

LITERATURE SURVEY

3.1 Image Processing Based Smart Traffic Control System [1]

3.1.1. ABSTRACT

As we all know that nowadays the problem of traffic in urban areas/cities has to be controlled and newer advanced technologies are to be introduced to improve the present traffic control system. With the usage of limited infrastructures, it is a difficult task to control the present-day traffic system. The simplest and only means of traffic control is by using a timer for each and every phase. There's also another way by which one can use electronic sensors for the detection of vehicles and to control the traffic accordingly. Here, the authors propose a system that uses image processing to control the traffic lights. Using the images captured, the system detects the vehicles. This proposed system does not use any electric sensors for detection. To capture the images, a camera is installed beside the traffic lights on roads. The main premise of this methodology is an empty road's image is captured which acts as a reference and then captures the images in a timely manner which are then compared with the earlier reference image. They will be matched sequentially so that we can control the traffic. Edge detection is being done by the **"Prewitt Edge Detection"** algorithm. The matching of images in percentage is done for some sequential durations of traffic light. Based on the percentage of matching, the control of the traffic system is being done.

3.1.2 INTRODUCTION:

Automation of traffic and monitoring systems are the need of hours these days for the management of usage of roads. One of them is by deploying various sensors to estimate the traffic and update all the traffic information sequentially to control the traffic. Another method is by deploying magnetic loop detectors in many technologies which is a difficult task due to their heavy maintenance and installation problems. With image recognition and measurement, the density of vehicles, speed of vehicles can be computed. Capturing of images of moving vehicles can also be done.

The proposed solution explains the following.

1. Finding the vehicles in the pictures captured by camera.
2. Automatically validate and control the traffic light.
3. To display the red light whenever the road is empty i.e. according to the reference image.
4. To display the green light whenever there is traffic on roads.
5. The duration of traffic lights needs to be controlled and adjusted accordingly.

3.1.3 COMPONENTS:

- Software Modules
- Hardware Modules
- Interfaces

1. Software Modules:

Image processing is done using MATLAB version 7.8

2. Hardware Modules:

- Web camera is used as **Image sensors**
- A personal computer for image processing tasks.

3. Interfaces:

Parallel ports of the PC are used as an interface between the hardware and software prototypes/modules.

3.1.4 METHODOLOGY

The steps involved in the methodology are,

- Capturing / Acquiring the image
- Conversion of given RGB to a better gray version.
- Enhancing the image for better clarity and quality
- Matching of images with the reference image using the given detection of edge algorithms.

The procedure involved in the methodology,

1. Phase 1:

The first step in this phase is to capture an image with a web camera. Then take another image when the road is empty i.e. no traffic to keep as a reference image.
This reference image when there is no traffic is taken at a location where the camera can perfectly capture and is stored or saved in the program for future reference. The conversion of the given RGB to a better gray is done on this empty road reference image. Image enhancement is done on the reference image by gamma correction.

In the end, Using “Prewitt Edge Detection” operator edge detection is done on the reference image.

2. Phase 2:

- In this phase, different images of the road will be captured real-time by the camera sequentially.
- All the captured images will then be sent for conversion from given level to gray.
- Image enhancement by gamma correction to all the images that are captured.
- Using prewitt edge detection operators, all the real-time captured images are subjected to edge detection.

3. Phase 3:

Edge detection procedure is done on both the real-time images and reference images. Based on the matching percentage, we calculate and analyze the lights of traffic signals.

0 - 10 percent → 90 seconds green light.

10 - 50 percent → 60 seconds green light.

50 - 70 percent → 30 seconds green light.

70 - 90 percent → 20 seconds green light.

90 to 100 percent → 60 seconds red light.

IMAGE ENHANCEMENT:

All the captured images that are acquired in RGB are converted into gray with the help of image enhancement. Each captured image must be in contrast with the background. In order to achieve this, a proper approach level must be known when the conversion in binary is being done. In this instance, all the image enhancement techniques come into picture.

DETECTION OF EDGES AND MATCHING OF IMAGES:

Step 1. Detection of edges:

There are many functionalities that can be extracted from an image such as points, lines and edges. But here they have used edges which can detect the changes in the images. Edges are the most important feature in every image and are essential to differentiate the distinct regions. Basically, between two different regions, the edges are the borders that are used in detection for matching.

"Prewitt Edge Detection" operator is used for detection of edge purposes.

Methodology:

To calculate the approximations of derivatives, Prewitt's operator is being used. It uses square kernels that wind with the captured image to find the edges.

For instance, Let Gx & Gy be two images that are captured at different points which contain both vertical and horizontal approximations which can be evaluated,

- Methods of detection of edges are used to find all picture elements in the image that will map the objects' edges.
- The achieved result is an image where all the picture components are detected.
- There are certain operators that are being used for this purpose such as prewitt, laplacian and sobel operators.

Gradient based edge detection algorithm is used to detect whether an edge is present in the image or not by gazing at the first derivative of the image for maximum and minimum.

Step 2: Matching of images:

Image matching is done based on edges i.e. edge based matching where two edges or representatives are put together. Each edge is evaluated and also compared against each and every edges of the other images that are captured. With the help of the matching of images, traffic light durations will be set accordingly.

3.1.5 CONCLUSION:

The study carried out by many concluded that DIP and edge detection is a good solution and advanced approach to set the traffic light states. The main inputs from this study tells us that it can reduce the congestion and bottlenecks in the traffic system and also circumvent the precious time that is wasted when green signal is ON in a road that is empty. By using all the actual real-time images as an input, this method is consistent in its own way of vehicle detection and also much better than the already existed systems. But the main drawback of this system is that it lacks one hundred percent accuracy and still needs a better improvement for it to stand out to its competitors.

3.2 An Approach On Road Traffic Analysis Using Image Processing [2]

3.2.1 ABSTRACT:

Style of transport for short distances is generally through roads in which as a result of the problem traffic overfilling. The traffic shortcoming area unit increasing attributable to varying movement,restricted usage of this infrastructure to present scenario, there is a necessity to attend longer waiting before the signals. This ends up in wastage of the existing system and it becomes slow. With this drawback, authors designed a paper that was able to facilitate the Traffic flow observation and analysis. The article presents a degree application of cv image method for implementing traffic flow observation and analysis. Here image process and pattern recognition methods are strategic and adapted for constraints of road traffic problems. These things offer well-desired abilities of the system to appear at the road, to initialize automatic vehicle search, speed calculation, recognize vehicle-plate number.

3.2.2 INTRODUCTION:

Observance and analyzing flow of traffic supported visual pic techniques, traffic analysis and observation in an extremely periodic observation increases and it helps to manage effectively.complex needs to cv modules and resolutions. Here the problem is introducing track automatically. Traffic analysis ends up in various forms: traffic congestion, speed violations, accidents, helmet detection and any against law actions which are done by road users. Automation approaches to those types of jobs are encouraged by most of the analysts and researchers. Here problem throughout the text are mainly implementing the dip techniques that are suitable for inspection ,analyses. The primary focus here uses the existing algo's and further develop that to achieve the goal and model of the system for traffic analysis is created.

The approach here focuses on ways in which image method, identifying patterns, cv algo's will be helpful for inspection,observation. Foremost necessary thing here is to change existing methods that suit period road look model systems so then implementation involves.Proposal of the theme was predicated on static video feed in addition to that devices connected to WAN. Possibility of system cover vehicles here are vehicle speed activity, plate number identification of vehicles and last but not least the street jam detection. Here Image process method tasks use the following techniques: segmentations, recognitions, image filtering, correction, identification, object modelling. “Technical responsibilities are frame extraction, video transmittal sequence, motion shooting, video sequence transmittal.” The traffic work analysis is achieved by generating the frames of traffic video data in MATLAB.

3.2.3 PROBLEM DESCRIPTION:

Recognition and processing the real time flow of vehicles needs lots of advanced mathematical, algorithmic and programming issues. Varied articles have thought of specific questions related to scene modelling, object pure mathematics accounting, image contours process. A need of data tactics and algorithms required in this digital technology world. The problematic here involves careful road observation because it is chosen in our analysis and has an order of different independent steps supposed to resolve tasks reasonably associated with every dissimilar one. These steps involve firstly input that can be image-feed or video-feed that is sent to the system, generating frames in an order and then applying masking on the images which involves filtering and linking, labelling, recognition of moving vehicle and then speed calculation.

The information has to be compelled to tend within the kind of video stream, so processed to find ascertained vehicles in each frame. The image events like image-segmentation then background-filtering follows detecting of edges keep in it well organized, vehicle-contour enclosed the examining frames. ensuing a confined centre of area given by space which is generated by counter , so for finding object speed and for trace happens in fed-frames which are generated by videos. Then there is a tag contour, which supposed to mark and confirm the variety of flow of objects within the zone which we focused on and then find congestion of the path. Subsequently there is a chance of noticing the class of vehicle, by exploiting its features and some attributes.Car buses,trucks,other vehicles. Finally we have to find and recognize car plates where violation occurs.

3.2.4 PROJECT SCOPE:

- Here the system monitors at intersections to find congestion then it also forecasts the flow that supports to control and avoid the congestion.
- Projected project firstly video-feed is captured with the help of the camera.That feed is sent for processing which is done by matlab then in case input is video generate frames and processes for finding the quantity of vehicles there within the captured feed and then at last count fed is sent to the server.
- Passing vehicles count at traffic intersections along the name of the signal where that count persists , which helps to know the density of vehicles at the intersections.

3.2.5 TECHNIQUES:

- Segmentation: It partitions an image into small objects. Normally, a lot of correct segmentation leads to successful recognition. And also, this technique involves background elimination like trees, stationary objects etc. Segmentation technique helps to focus on the extract the objects of our interest.
- Recognition:

  Recognition is the process of identifying object and feature extraction. Example: labelling objects has vehicles etc which follows the vehicle properties. Pattern-Matching recognition techniques used for identifying or observing patterns.

  And additionally, Author did some analysis to add that in that he supporting image process could be a higher technique to manage the phase transition of the traffic signal.

3.2.6 PROPOSED METHODOLOGY:

The Author's projected methodology is enforced by below modules.

- Network module: This component majorly concerns the info about the masking the path and image filtering and other noise data which exists.
- Traffic generation module: This component concerns the info associated with in what way traffic will generate and the way of traffic density raised. It will be helpful for notifying people of the best traffic possible route to their destinations and also informing people that at particular junction traffic is more/less.
- Congestion Component: Here the component concerns the data info related to the congestion that happens at the road. And also it determines the management of traffic when a dead-end block occurs at the road.
- Traffic density optimisation module: it's the optimisation method that helps for object or image optimisation.

3.2.7 PROS/CONS:

Lane masking will help in fine tuning the vehicle detection and will improve the quality of information processed. Proposed system may take a long time to process the image due to background elimination.

3.2.8 CONCLUSION:

Automation helps to avoid the delay that happens at the signal and improves the movement of vehicles. The green light period at the empty roads will be avoided. Traffic signals lights will be controlled efficiently because of automation.

3.3 Traffic Light Management and Violation Detection Using Image Processing [3]

3.3.1 ABSTRACT:

Currently because of the population rise the transportation demands are raised. Due to the increased population there is a need to improve our existing transportation system. Higher population leads to heavy usage of vehicles on roads. Old-style traffic controller provides a time-slot for every path of the road. To reduce wasted time and fuel waste on the road, an efficient and smart traffic control tactic is needed. Alternative innovative way is to keep the proximity sensors on the road path. The traffic data is provided by the sensors. Based on the captured sensor data, control-aition happens at the signals. This project suggests an innovative mode of traffic control. A camera is fitted along with traffic lights that is used to record images of live paths of traffic. Based on the recorded images dip helps to ease the job of controlling the traffic signals and knowing the traffic density. Projected structure helps to avoid the time which is being wasted at empty roads and helps to better usage of fuel, time in an effective manner.

3.3.2 INTRODUCTION:

Nowadays the rise of expansion of land leads to a lot of transportation necessities. It leads to a heavy traffic congestion going to happen in most of the areas. This leads to high time usage and fuel requirements because of the congestion which is happening.

The congestion which can happen in many ways like time, uncertain situations like road maintenance and accidents, climatic conditions etc. Naïve approach to solve the problem of congestion is building flyovers and constructing additional roads. But there is a problem in that approach, occupying land and a lot of economic ground resources activities involved and it takes time for collapsing the existing and creating new constructions. Because of these activities we have to use the present available resources to improve traffic management.

A lot of continuous monitoring of vehicle density on road can be the solution for congestion which is occurring can be reduced by managing the time duration at signals based on vehicle density. The advantage of having a smart transportation system which helps to reduce capital cost and other service costs. It helps the planning of transportation for any further developments.

An intellectual traffic system management largely has the benefit of using visual processing techniques.

3.3.3 EXISTING SYSTEM:

The disadvantage of controlling the traffic manually needs a lot of manpower and their attention towards the public and involvement of police should be needed. Nowadays the increasing traffic leads to more involvement of manpower and more the resources required which they use to control. Here strength of traffic police controlling is directly proportional to the traffic density. In some popular areas where people movement is high, controlling the traffic manually is very difficult.

"Autonomous traffic control systems are a type of embedded system. It helps to control the four set of traffic lights at traffic islands." it's not flexible because we're keeping the time period fixed for green,red,orange lights to work. These time periods are chosen according to the normal and peak time traffic. We have to keep in mind that the traffic density varies proportionally because of population increase and almost all people started using vehicles.

"Another possible way to control the traffic light is vehicle-actuated control". Here basically there is a frequent attempt to set the green light timings.A detector is placed at some distance from the stop line and it's a basic type of vehicle-actuated control. Here the controller does the job of adjusting green light time based on the data that is received from the controller. Detector acts like a proximity sensor. Like Normal existing traffic systems this also has accuracy problems. The Drawback of this control system is it only pays attention to vehicles when there is a green light and it also does not compare the total number of vehicles that are present at the green signal without knowing the total number of vehicles stand by on the red light signal. They work correctly when real traffic flows happen usually. "Traffic flow is assumed when the unit extension of the green signal is selected."

3.3.4 PROPOSED SYSTEM:

Here the system tried to improve the transportation system efficiently with existing resources rather than constructing new ones. It controls the traffic light signals based on the current road traffic situations and the possible violations which occur in that scenario.

The system takes the advantage of dip for controlling traffic on the road. The methodology idea of the system categories the vehicles and density of vehicle in each category and then traffic signals are controlled on basis of available traffic data.

The software section they use here is an open cv platform—it was like acquiring the live-road traffic feed,some image enhancements and some type of algorithms like svm to categorize and identify the vehicles. Here some components are necessary to support the system and to make it available all the time like source and processing power,camera,sensors, should be working properly and make sure there are no malfunctions involved and alignment of the devices should be placed properly. Some kind of periodic checks are needed. Here the approach is that camera capture the live feed of traffic data in digital format and it sent to cpu then cpu do some operations like categorizing vehicles,background elimination,Image enhancement, Identifying vehicles and also traffic signal generation based on the traffic density and storing the images where the violations are occurred. Matrix array is used to represent the digital images here the matrix contains the intensity of values. To find the adequate info from acquired images we have to do some operations like image enhancement and edge detection.For categorizing the objects from input it may be video or image svm is used. For the training supervised learning is used after that system the desired object from image and opencv is used for feature identification and detections. It contains the bulk set of software algorithms.

3.3.5 METHODOLOGY:

Here first collects the traffic feed from cameras that feed is converted into grayscale and then apply some defined thresholding function then generate images from that grayscale feed so there a scope for svm plays a role to categorize and identify the vehicles,finding vehicle density etc. At last stage system controls the signal lights based on the traffic density and it also implements some innovative features like road is facing a heavy congestion and if road is having a high density at path and another path having low density it shifts the time period slot of low density path to the high density path so it reduces the unwanted wasting time at the junctions.

3.3.6 CONCLUSION:

The vehicle categorization by using image processing and svm will be achieved efficiently. The system also helps to reduce the amount of time being wanted at the empty roads and intelligent systems support any further new traffic violations created in future. There is a scope of creating some controlling strategies for vip’s and emergency vehicles like hospitals,fire etc and the system also helps traffic policemen for saving the images of violations and the number plate of vehicles.

3.4 Automated Traffic Rules Violation and Licence Plate Detection [4]

3.4.1 INTRODUCTION:

This section details the papers read to gain information on background, data used, and the current methodologies being used for detecting violations and analysing number plates. There are three algorithms that are being used,

1. Edge detection to encompass boundaries.
2. Hough transform, feature extraction technique used to analyse images.
3. The last algorithm is used to identify the patterns in the extracted license plate image features.

3.4.2 PROPOSED SYSTEM:

Arduino is used to capture the frames. This project uses Automatic Number Plate Recognition (ANPR) techniques for plate and character identification.

3.4.3 METHODOLOGY(3 parts):

1. The Edge Detection algorithm performs mathematical operations on each pixel and draws an edge.

2. The Hough transform, converts the pixel values into a smaller matrix with easily processable attributes.
3. The last algorithm identifies patterns based on a training model and extracts the image features.

Part -1

The edges obtained from the images are dampened by the following few effects:

- Due to a limited depth of field, a focal blur occurred.
- Light sources with a positive radius created a penumbral blur due to the shadows.
- Shading at a smooth object.

Part - 2

- This technique was originally meant for line detection. We use it for its extraction capability.
- The linear Hough transform algorithm a 2-D array a.k.a the accumulator , to detect the presence of a line following the locus of a point given by by $r = xcos\theta + ysin\theta$.
- The Hough transform checks if there is enough evidence of a line being present at every pixel (x, y) and its neighbourhood.

Part - 3

For pattern recognition, the k-NN formula is used for classification and regression. In k-NN classification, the output will be a category feature into which the node will be classified. An object is put into a class by the majority, with the category assigned based on its k nearest neighbors.

3.4.4 CONCLUSION:

In this paper, the automatic number plate recognition is performed on a regular vehicle licence plate. The system uses image processing techniques for identifying the vehicle from the database stored in the computer.

CHAPTER - 4

PROJECT REQUIREMENTS SPECIFICATION

4.1 PROJECT SCOPE:

MOTIVATION:

The number of vehicles on roads are increasing day by day and it has given rise to problems like accidents, traffic and bottlenecks. As we all know, traffic lights can only be controlled by advanced technologies as it requires a lot of surveillance and monitoring to control the traffic. Due to the large number of these traffic signals it is impossible to monitor all of them. Hence we need to make an automated traffic light system that enables the flow of traffic as and when required by it. Road safety is a major concern by traffic as many rides fail to maintain all the traffic rules and they continue to violate them. Due to this the policeman is facing various problems as they cannot monitor every single violation.

4.2 PRODUCT FEATURES

- Traffic Light control is automated.
- Density based time allocation to signal.
- Specific Scheduling orders for lights are followed.
- Helmet Detection.
- Traffic Line violation.
- Number Plate Extraction.

4.3 USER CLASSES AND CHARACTERISTICS:

Policeman:

To be able to review remotely the potential violations that have occurred. To be able to get the details of the licence plate of those violating vehicles. He/She may then confirm the violation of the rules and then impose a fine or discard the report. This prevents this user class from using a constant monitoring system and to use an asynchronous monitoring system.

Public:

Reduction in traffic during peak hours and smooth flow of intersection traffic. Prevention of jams and reduction of traffic violations. This improves the throughput of individual users and may reduce pollution due to lesser usage of vehicles.

4.4 Operating Environment

- Microcontroller(Arduino or Raspberry Pi)
- Any Linux based OS
- Python-3
- Most recent version of Anaconda and its set of python libraries which include Scikit as well as OpenCV.

4.5 General Constraints, Assumptions and Dependencies:

- Constraints: Camera Quality, Availability of hardware processors
- Assumptions: Cameras at every junction and with enough lighting, proper connections between the surveillance centre and the cameras, visually identifiable violations.
- Dependencies: Adequate Lighting, Electricity, Regular Maintenance of cameras.

4.5.1 RISKS:

- Inaccuracy during measurement.
- Physical damage to hardware.
- Software flaws in open source libraries.
- Interruption in power supply

4.6. FUNCTIONAL REQUIREMENTS:

- Image Recognition: Taking the camera feed as input and splitting it into frames in order to recognize objects based on a few parameters. The output is the object frame.
- Feature Extraction: The frames received are fed as input and matched to the respective attribute and those attributes are sent as features for further processing.
- Density Calculation: The features extracted are counted and density is calculated based on the count and sent to the next phase.
- Controlling traffic lights based on density: The received density is compared with a forecasting model to determine the status of the light and the new status is sent to the light.
- Violation detection: The features extracted are taken as the input given by the feature extraction module and checks if a violation has occurred and returns a boolean value.
- Number Plate Extraction: In case of violation signal sent by the violation detection module, the image is received and ANPR is used to recognise the number plate and the image and the number is sent for further action.

4.6.1 External Interface Requirements

User Interfaces

- TKinter with

- Time
- Date
- Type of violation
- Photographic evidence of Violation
- Number plate Photograph.
- Violated or Not Violated Button

Hardware Requirements

- Camera with at least 12 MP.
- A microcontroller or a processor capable of running programs and an OS.
- Wires for the light control or a display indicating the light arrangement.
- Camera Feed either wired or wirelessly.

Software Requirements

- Microcontroller(Arduino or Raspberry Pi)
- Any Linux based OS or Raspbian
- Python-3
- Most recent version of Anaconda and its set of python libraries which include Scikit as well as OpenCV.

4.7 NON-FUNCTIONAL REQUIREMENTS

4.7.1 Performance Requirement

- The accuracy required is above 80%.
- Number plate recognition must exist. If not then just send evidence.
- 100% uptime and constant monitoring.

4.7.2 Safety Requirements

- Street lights and reliable hardware. Someone to review the violation.
- Protection for cameras.

4.8 SECURITY REQUIREMENTS:

- Security: All the data is stored by the Govt. agency. This data is free to use under RTI

CHAPTER - 5

HIGH LEVEL DESIGN

5.1 Design Considerations

5.1.1 Design Goals

- Our goal is to automate the existing system.
- We will maximise the efficiency with the help of automation.
- Image processing has been chosen because of the large availability of existing cameras which reduces the cost of the project as well and gives moderately accurate results.
- Since it is automated, we aim to provide full-time surveillance.
- This method helps the state police to focus on other important problems and also enforces the existing traffic laws to ensure better conduct by the public.

5.1.2 Architecture Choices

Papers	Pros	Cons
Image Processing Based Smart Traffic Control	Image processing fares better than other techniques to regulate traffic.	Fails in some cases, Not 100% accurate and may face technical difficulties
Traffic Light Management and Violation Detection Using Image Processing	DIP and uses image enhancement techniques and SVM to improve accuracy.	Not 100% accurately timed light control and possibility of better ML algorithms

Automated Traffic Rules Violation and Licence Plate Detection	Uses k-NN and Edge detection by converting outlines to B&W for easier processing	More advanced techniques of DIP exist to extract features and text matching is not 100% accurate
An Approach on Traffic Analysis using Image Processing	Lane masking will help in fine tuning the vehicle detection and will improve the quality of information processed	May take a long time to process the image due to background elimination

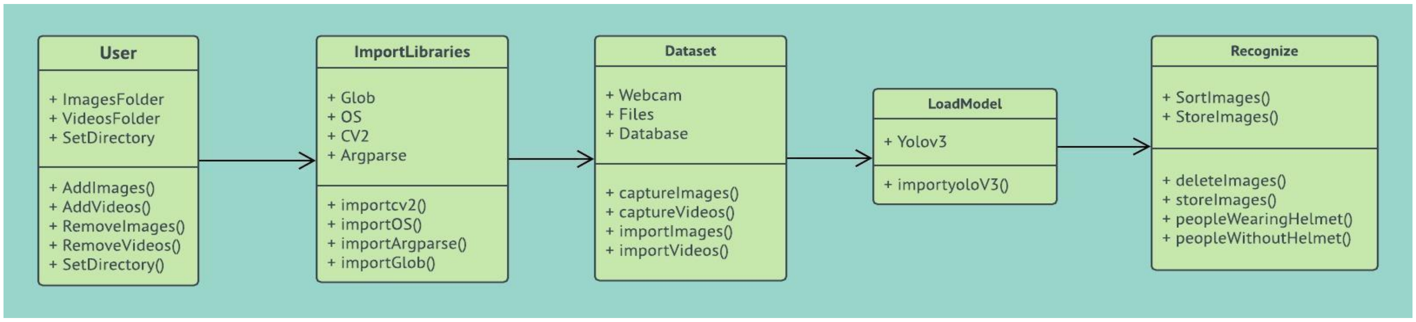
5.1.3 Constraints, Assumptions and Dependencies:

- Constraints: Camera Quality, Availability of hardware processors
- Assumptions: Cameras at every junction and with enough lighting, proper connections between the surveillance centre and the cameras, visually identifiable violations.
- Dependencies: Adequate Lighting, Electricity, Regular Maintenance of cameras.

CHAPTER - 6

LOW LEVEL DESIGN

6.1. Class Diagram:



6.1.1. Helmet Detection

6.1.1.1. Class USER:

In this scenario, Traffic policemen are the users.

6.1.1.2. Data members of USER:

ImagesFolder/VideosFolder: They have access to the videos/images that are captured during traffic.

AddImages/AddVideos: The Traffic Police can feed a particular city/junction images to the helmet detection model.

RemoveImages/RemoveVideos: The Traffic Police can remove particular images/videos from the helmet detection model.

6.1.1.3. Class Import Libraries:

In this class, a set of required libraries and modules are being imported.

6.1.1.4. Data members and Methods of USER:

Glob: This module is used to retrieve files/path names matching a specified pattern.

OS: This module is used in Python that will provide functionalities for interaction with the operating system. OS module is one of the standard utility modules in python.

Argparse: This python module is used to generate help and issues messages and errors too whenever the user provides invalid arguments in the program.

(Below modulus has to be added in line violation not in helmet detection expt cv2)

Tkinter: This module provides a fast and easy way to create a Graphic-user interface for python applications.

OpenCV: This module mainly focuses on image/video processing and analysis , it helps to solve computer vision problems.

ImageIo: This module provides a way to read/write the wide range of image data.

PIL: This module adds additional functionality like opening, manipulating, and saving many different image file formats.

6.1.1.5. Class Load Model:

Yolov3: This algorithm is helpful in creating layers of convolutional and stack together as a whole.

6.1.1.6. Class Recognize:

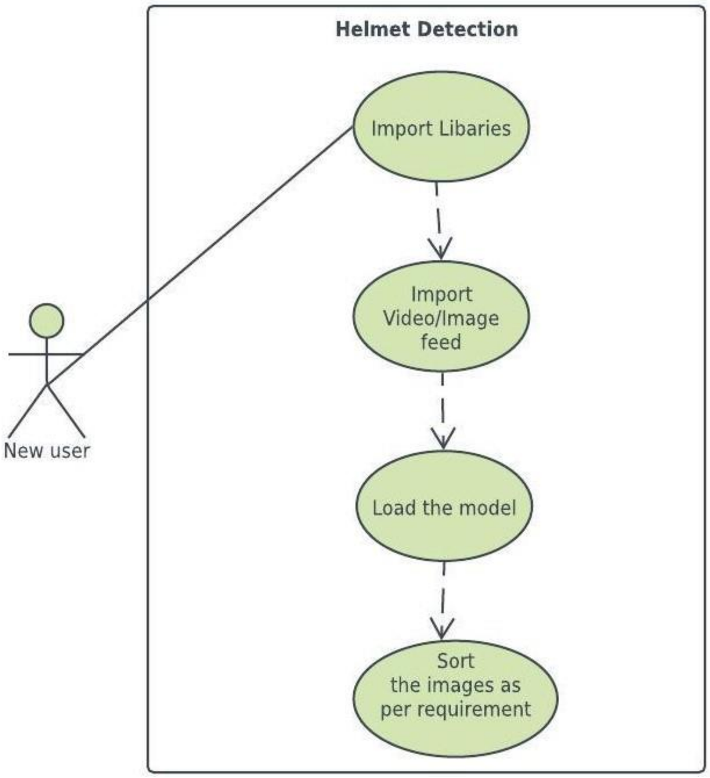
Description: In this class, images are sorted and stored.

6.1.1.7. Methods of Class Recognize:

peopleWearingHelmet(): Images of people wearing helmets are detected.

peopleWithoutHelmet(): Images of people who are not wearing helmets are detected.

6.2. Use Case Diagram:



Considering Helmet Detection, the use case diagram consists of the following use case items,

i) Importing the Libraries: All the necessary modules and libraries are to be imported for the model to work.

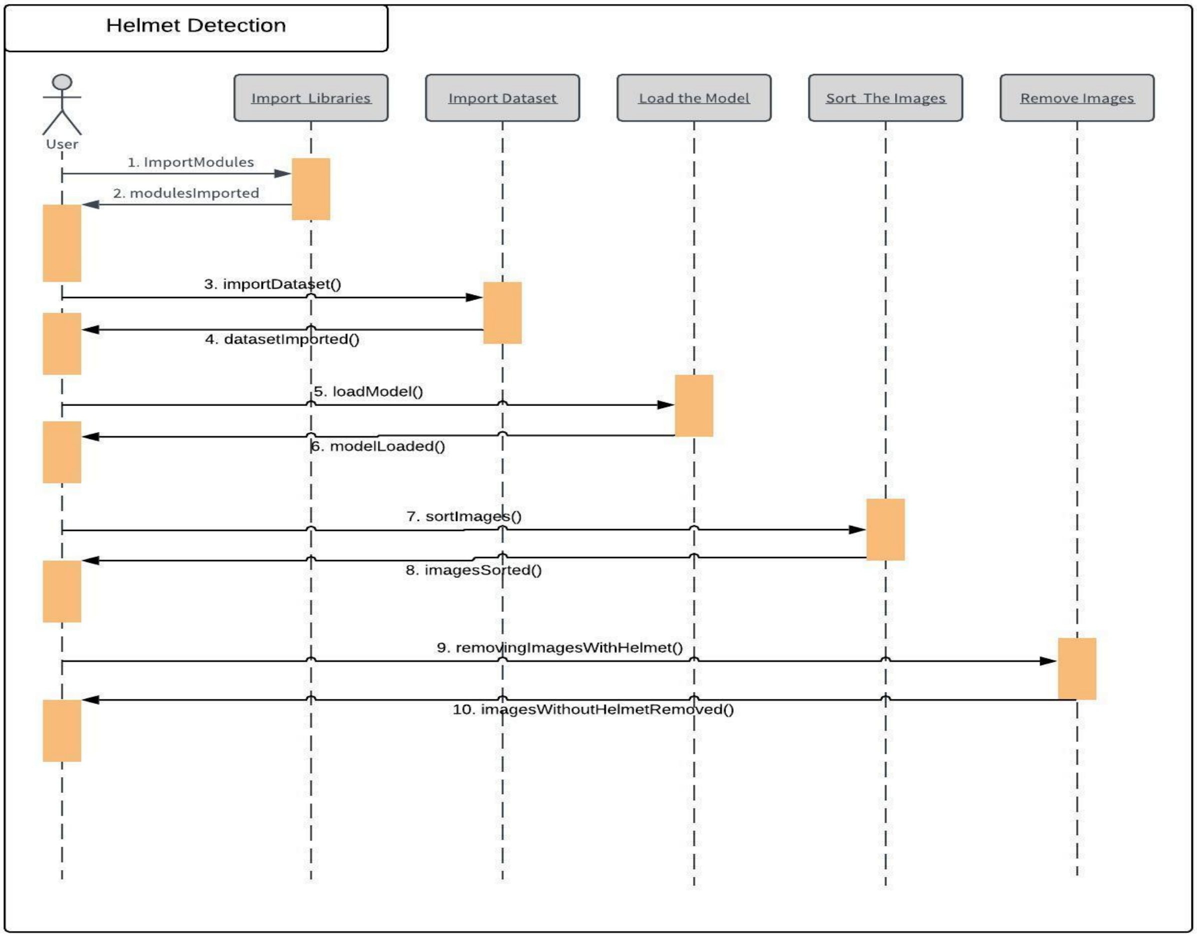
ii) Importing the Video/Image Feed: All the recorded videos/images of the traffic are taken as input by importing it.

iii) Loading the model: Helmet detection model is being loaded to detect the helmets.

iv) Sort the images: We sort the images as per the requirement so that the images of people with helmets are removed and without helmets are being detected.

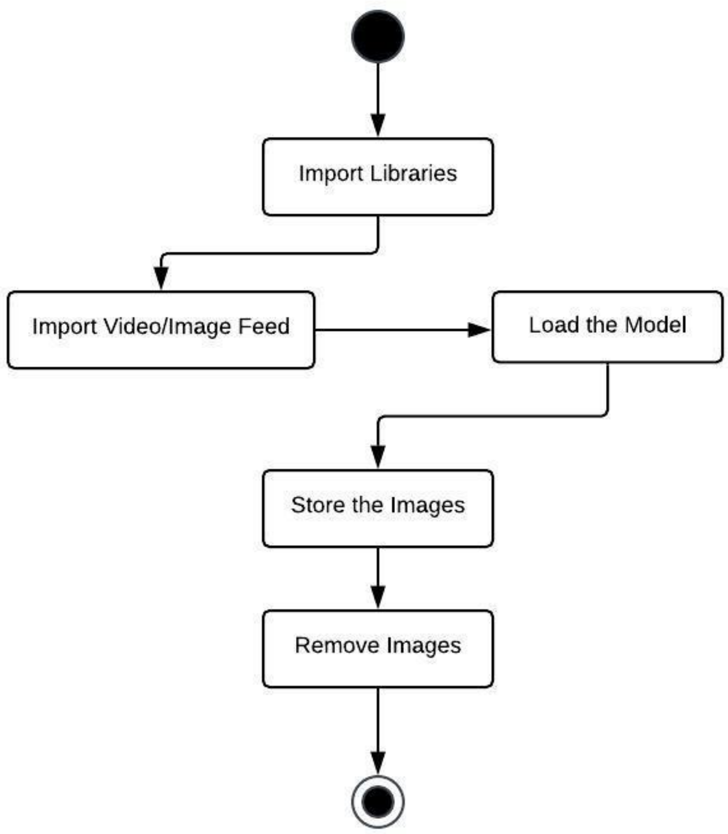
6.3. Sequence Diagram:

Below is the sequence diagram for Helmet Detection that consists of all the classes that are described in above paragraph

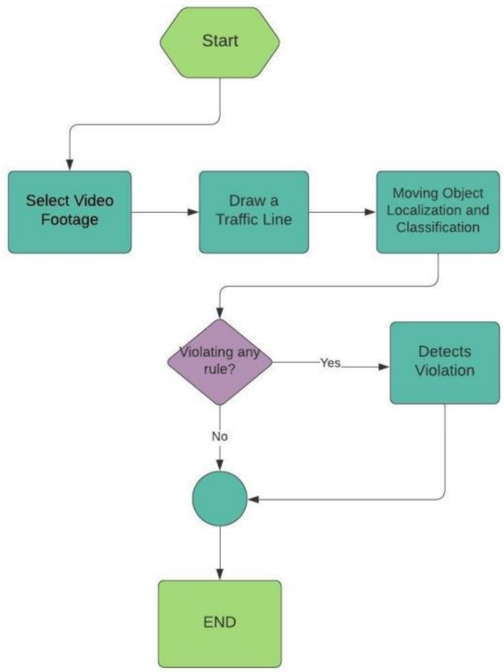


6.4. Activity Diagram:

Below is the activity diagram for helmet detection and line violation.



6.5. Activity/Flow chart for Line Violation:



The following are the steps involved in line violation,

1.The user needs to select the video footage

2.the user needs to draw a traffic violation line at the signal junction.

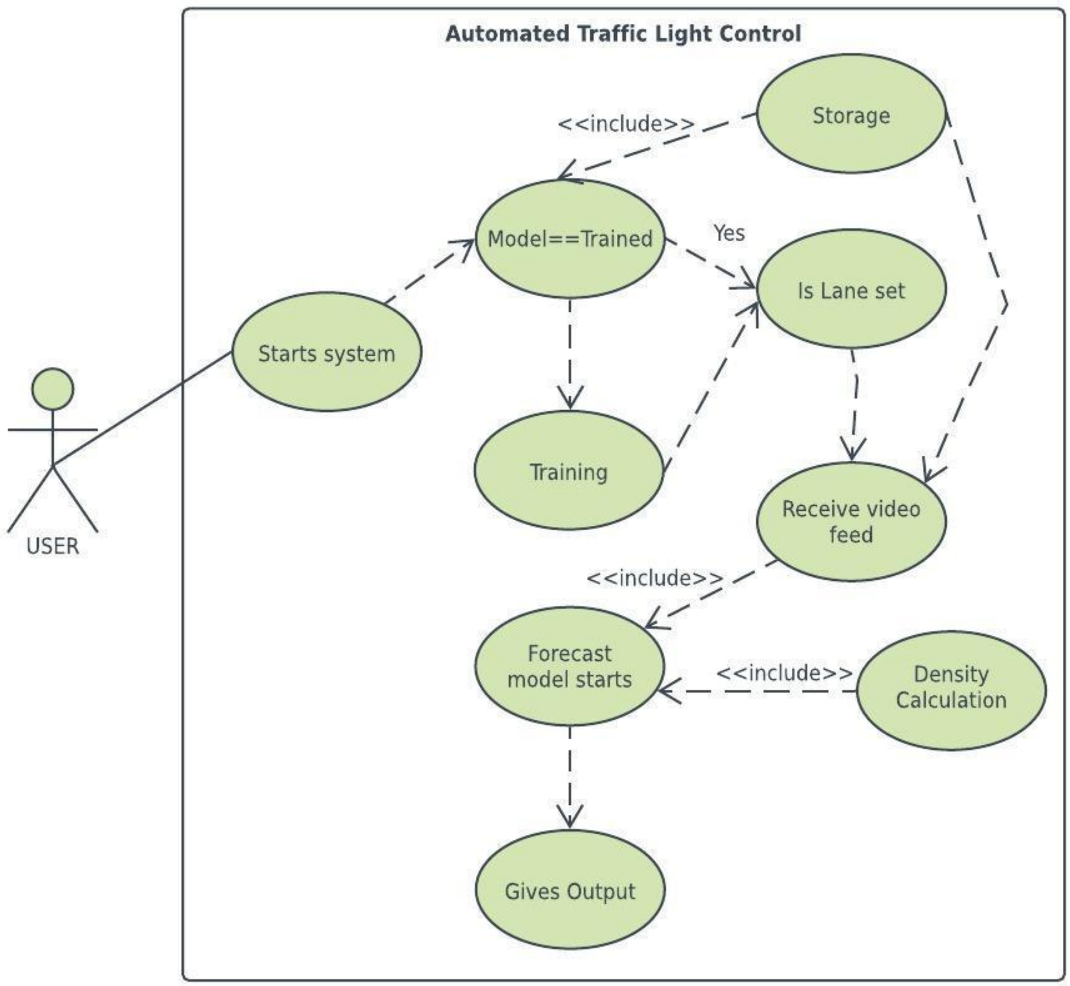
3.Now, the model will try to detect the moving objects and also classify the vehicles which cross the line drawn in the previous step.

4.Vehicles which cross the line during red light is considered a violation.

5.Now, the pictures of these violated vehicles are stored and further sent to license plate detection.

6.6. Use Case Diagram for Automated Traffic Light Control:

User as in the traffic policemen starts the system. This model is being trained to provide accurate results. Video feed is sent as input which again will be divided into image frames and used as an input to the model. We calculate the density and then forecast the model to output the desired results.



CHAPTER - 7

SYSTEM DESIGN

7.1 TRAFFIC LIGHT AUTOMATION

We attempt to design an architecture which will try to automate the traffic lights without manual control. This proposed system is a 5 step methodology as follows :-

1.Image Cleaning

The training images are obtained from a CCTV camera or a camera present at an intersection or a junction. These images are the result of the frames generated by the video stream. From each of these training images, we extract the Region of Interest (ROI) which holds the object we are interested in. We select the ROI in such a way that all the unwanted parts are weeded out and only the object of interest is included.

2.Feature Extraction

The ROI that we extracted in the earlier step is now divided into a grid of XxY pixels within each cell. The number of cells vary based on the ROI which varies based on different lanes and intersections.

3.Training the model

Each of the cells extracted is fed into Prewitt's Edge Detection algorithm and then, the features are sent to the SVM model to predict if the cell has the object of interest or it does not have. The model is trained and saved.

4.Density Estimation

For the traffic density prediction, we take the live images that are captured from the cameras at the junction and then, extract the ROI and create a grid with each cell of dimensions X x Y pixels. Then we pass each of these cells to the trained SVM model as a test set and obtain the predictions. The SVM predicts if it contains traffic or empty roads for each of the cells. With this method, we classify the whole grid and can estimate the traffic density by getting the number of cells having traffic divided by the total number of cells expressed as a percentage.

5.Traffic Light Signalling

After obtaining the density, the density across all the signals are compared and ranked. Based on the rank, we allocate the time to the signal. The time allocation is done as follows based on a ARIMA forecasting model. The forecasted density is then taken and a decision is made based on the following.

Density(%)	Green Light Duration(sec)
0-30	15
30-50	20
50--70	30
70--90	45
90--100	60

The scheduling of the traffic lights is done in a Round-Robin fashion. After each round, the forecast is done again and the process is repeated. Each signal, after allowing a green light, shall not be allowed to get another green signal until it is time for its turn..

7.2 TRAFFIC VIOLATION DETECTION:

In the proposed method we attempt to design a model for identifying vehicles which violates the traffic rules such as detection of non helmet riders, line violations during red light.
Following are the detailed methodology for the list of violations:

First we select a video footage, we split the video into grids and use object detection model where person,vehicles are identified and then we find for the following violations:

7.2.1 Detection of non-helmet riders using YOLO algorithm :

The input frame which is received by the object detection model is used to detect 2 classes i.e., “vehicle(2 wheeler)”, “Person”. At the output we get the probability value along with confidence of detection of image and required class detection.

With their image number and class name in order the detected objects are extracted and stored as separate images so for example if the extracted object is a bike it will be saved as bike-1,bike2, etc.. similarly for the person as well and these details are stored in a dictionary which can be used for further processing later.

- **Helmet Detection :**

The Person’s image is given as input to the helmet detection model, once the person-bike pair is obtained. The person’s image is cropped to one fourth of the image because while testing for the helmet detection model some false detections are observed. This helps us in removing wrong results or false detected cases where a driver is holding his helmet in hand while riding or keeping it on the bike while riding instead of wearing.
If the rider is wearing the helmet then further processing is not required, but if he is not wearing the helmet then we need to crop the image and process it to the helmet detection model where a bounding box around the helmet along with the detection probability is displayed.

- **Licence Plate Detection:**

If the helmet is not found then we proceed to this step where the image of the vehicle is given to the license plate detection model. First we train the model by taking the set of images without the rider wearing the helmet and then using a labelling tool we annotate the license plate in those images i.e., by creating a bounding box around the license plate in those images so that the model can learn. Then for detecting the license plates these images are used to build the training model.

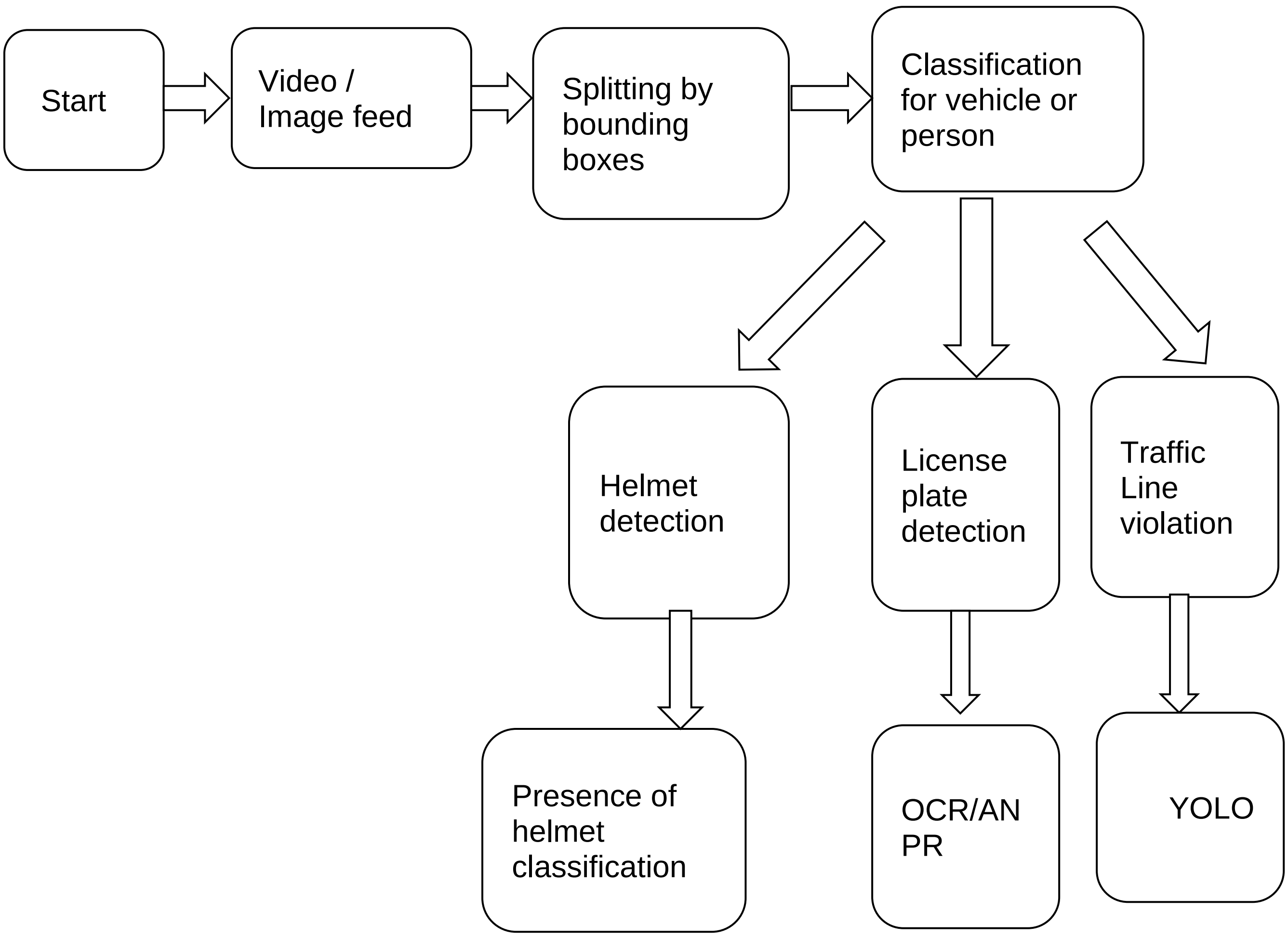
In the training model a bounding box is created across the license plate for the given image. The json file stores information such as coordinates of the bounding box, name of the class and confidence detection. These extracted images are stored so that we can use the coordinates stored in the file which are used to extract the license plate image

Sometimes, for a single bike image, there might be multiple bounding boxes detected. In that case, a threshold of half of one is set for confidence of detection. Confidence greater than threshold is chosen while reading details of the bounding box in the json file.

- **Line violation :**

This model is implemented by drawing the traffic line on the road in the video footage of the user. This line indicates that traffic light is red in state.Violation occurs when vehicles cross the traffic line. These vehicles are detected using an object detection algo named YOLO. Once the vehicles are detected violation cases are checked. If the traffic light is red in state and if any vehicle passes the line drawn violation happens. The bounding box around the vehicle becomes red after the violation is detected. And other detected objects will be having a green bounding box.

Flowchart for a complete traffic violation detection system:



CHAPTER 8:

METHODOLOGY

8.1. Traffic Light Automation:

1.Image Cleaning

The training images are obtained from a CCTV camera or a camera present at an intersection or a junction. These images are the result of the frames generated by the video stream. From each of these training images, we extract the Region of Interest (ROI) which holds the object we are interested in. We select the ROI in such a way that all the unwanted parts are weeded out and only the object of interest is included. The ROI that we extracted is now divided into a grid of XxY pixels within each cell. The number of cells vary based on the ROI which varies based on different lanes and intersections. Each of these cells are stored in a temporary directory and are sorted based on the presence of a vehicle or not. The minimum amount of images required is around 10,000 based on an estimate for acceptable accuracy. The sorted images are stored in separate directories inside the temporary directories and are separated as containing vehicles or not containing vehicles for supervised learning. This separation is done manually.

2.Feature Extraction

Each of the cells extracted is fed into the ROI analyser which uses an edge detection algorithm which then saves the result of the analysis on each image in a serialised format a.k.a pickle format and then saves it locally. The ROI's at this stage can be deleted as we have already extracted the required data. The analysis conducted uses the Histogram of Oriented Gradients(HOG) and Local Binary Patterns(LBP). These two algorithms use the gradient difference present in different pixels to find out the edge. The HOG is meant to apply a transformation of 1-0-1 row matrix to the existing grid of the image and gradient magnitude and direction is obtained. This is then stored for each pixel in the grid and a line can be plotted if necessary by calculating the common gradient directions. The LBP works in a similar manner except it calculates the value of each pixel to be the concatenated binary output in a clockwise direction around the central pixel. These binary values are obtained by comparing the central pixel value with that of its neighbours. These two features are then saved in a pickle file.

3.Training the model

The saved features from the previous step are now loaded back into the next phase. Here, we use an SVM(Support Vector Machines). We have chosen SVM due to its ability to distinctly separate the data since our data requires only binary classification. Here we use a linear kernel as we have separated the images into a binary based system. Using a non parametric model will not only increase the time taken but also reduce the accuracy as it grows with the dataset size. The model is trained on the features extracted from images in the previous step and is again saved as a pickle file.

4.Density Estimation

For the traffic density prediction, we take the live images that are captured from the cameras at the junction and then, extract the ROI and create a grid with each cell of dimensions X x Y pixels. Then we pass each of these cells to the trained SVM model as a test set and obtain the predictions. The SVM predicts if it contains traffic or empty roads for each of the cells. With this method, we classify the whole grid and can estimate the traffic density by getting the number of cells having traffic divided by the total number of cells expressed as a percentage.

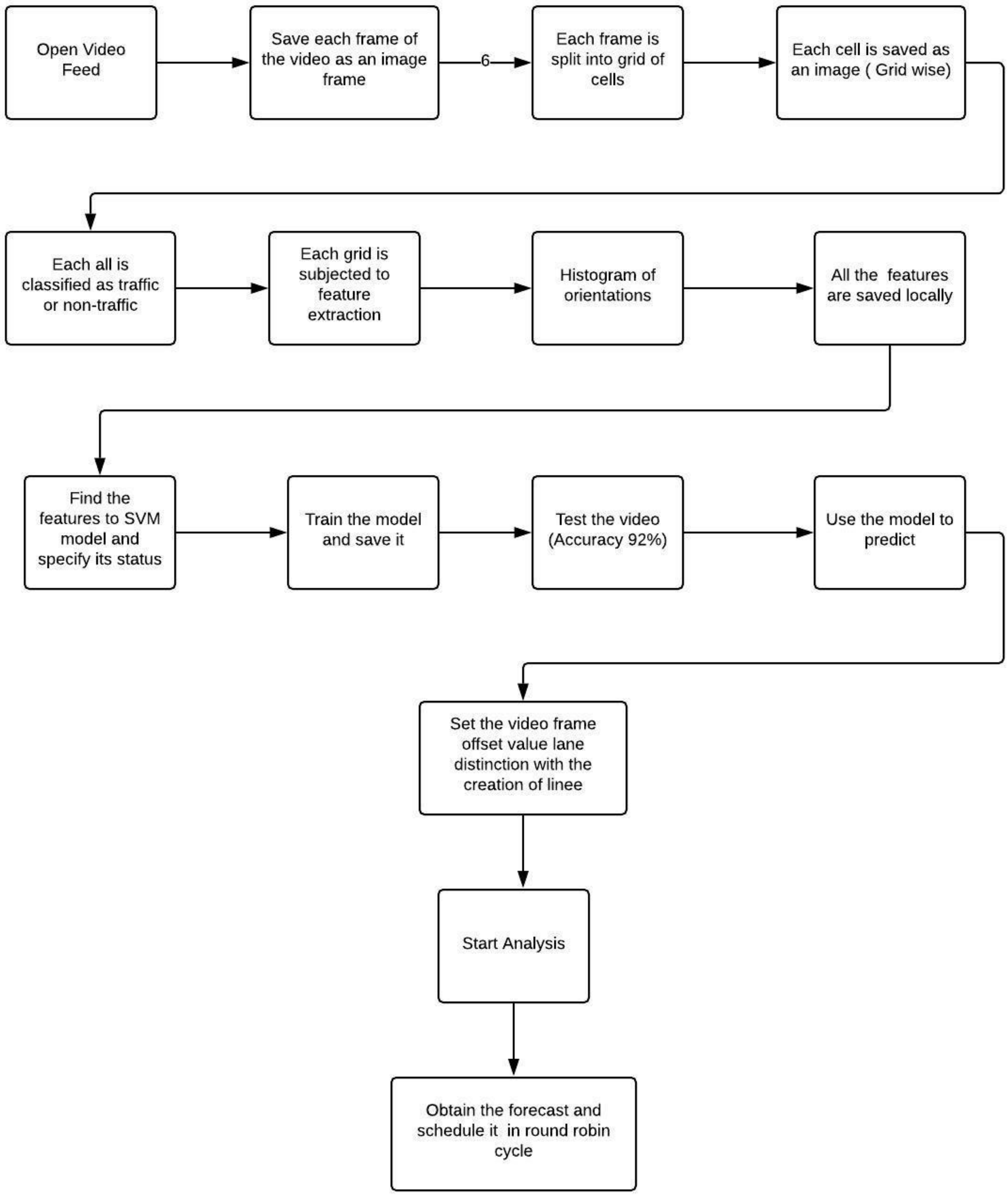
5.Traffic Light Signalling

After obtaining the density, the density across all the signals are compared and ranked. Based on the rank, we allocate the time to the signal. The time allocation is done as follows based on a ARIMA(Auto Regressive Integrated Moving Average) forecasting model. We have not chosen LSTM(Long Short Term Memory) as the predicted densities follow a time series pattern. Since this is short term remembrance data and a low hardware compatible project, we will not use more models to train and test. We have chosen to forecast the time to reduce the time spent per lane overall in the junction. The forecasted density is then taken and a decision is made based on the following.

Density(%)	Green Light Duration(sec)
0-30	15
30-50	20
50--70	30
70--90	45
90--100	60

The scheduling of the traffic lights is done in a Round-Robin fashion. After each round, the forecast is done again and the process is repeated. Each signal, after allowing a green light, shall not be allowed to get another green signal until it is time for its turn.

Below is the detailed Workflow/Activity Diagram that explains each and every step of implementation in a much detailed way


Open Video Feed
Save each frame of the video as an image frame
6
Each frame is split into grid of cells
Each cell is saved as an image (Grid wise)
Each all is classified as traffic or non-traffic
Each grid is subjected to feature extraction
Histogram of orientations
All the features are saved locally
Find the features to SVM model and specify its status
Train the model and save it
Test the video (Accuracy 92%)
Use the model to predict
Set the video frame offset value lane distinction with the creation of linee
Start Analysis
Obtain the forecast and schedule it in round robin cycle

8.2. Line Violation:

8.2.1. Vehicle Classification:

Using the CCTV camera or camera present at a junction, all the vehicles are being detected. With the help of an object detection model i.e., called YOLO (You Look Only Once) algorithm, all the vehicles and the bounding boxes will be put into their own corresponding classes. This algorithm has a better accuracy than many other existing algorithms and it is also helpful in detection of objects.

Features:

1. Bounding Box Predictions:

Using the YOLOv3 algorithm, we need to classify all the bounding boxes and all the calculations must be done as a single network so that everything is classified into a complete detection. This YOLOv3 algorithm works similar to logistic regression in such a way that all the calculations are 1, then all the bounding boxes will become an overlap over the object. It will predict only 1 bounding box prior for one ground truth object and any error in this would incur for both classification as well as detection loss. With this there might be many other bounding boxes that have their objectness score pretty much over than the required threshold value but the value will be less than the required best object. All these errors will not occur for the loss of classification but for the loss of detection.

2. Class Prediction:

Independent logistic classifiers are used in YOLOv3 algorithm where it is being used for separate classes. All these processes are must for this to make a multi-label classification. Using this multi-label classification, each and every box will predict all the classes of the corresponding bounding box.

3. Predictions across scales:

To support detection at varying scales YOLOv3 predicts boxes at 3 different scales. For the predictions across scales, all the features will be extracted from each and every scale using the features of pyramid networks. Using the YOLOv3 algorithm, we can use the feature of predicting across scales in a much better manner. Using the dimension clustors, all the bounding boxes are being divided to 3 scales. These scales are then being counted as 3 bounding boxes per scale which in the end makes a total of 9 bounding boxes.

4. Feature Extractor:

YOLOv3 uses a new network- Darknet-53. Darknet-53 has 53 convolutional layers. This YOLO algorithm is a better performer and more powerful than many traditional approaches. The most important feature i.e. feature extraction is easier and convenient in this algorithm.

8.2.2. Violation Detection:

The vehicles are detected using the YOLOv3 model. After the detection of all the vehicles, they moved to license plate recognition so that everyone is caught and details are sent to the traffic police. A straight line called the traffic violation line is drawn on the road near the signal in the CCTV footage or the camera footage from the junction. The crossing of this line during red traffic lights is treated as a violation. So if anyone crosses that line, all the vehicles that cross will be treated as violated and all these images of violated vehicles will be sent for license plate recognition. For the violation to occur, vehicles need to cross the line drawn. Whenever a vehicle crosses that line it is considered a violation and a red bounding box is drawn onto each vehicle. Other vehicles will be marked a s green i.e. not violated.

8.2.3. Implementation:

OpenCV is a python module that is also an open source module that is
helpful in computer vision which also contains machine learning techniques. This is used in this project for image processing purposes. To implement the classifier of vehicles, we use TensorFlow

8.2.3.1. Graphical User Interface (GUI)

The GUI i.e., graphical user interface contains all the features that are needed for the model. Editing is not required in this module for management as everything can be used to infer. If we need to play with the video we can do it by just opening it.
Firstly, to start the model, the user should select/click on “Open” to select the CCTV video footage and then add it. The user can also add videos from the stored files.

Whenever we open the CCTV video footage that is stored, the existing system will open it as a preview video so that we could get a glance of it. This preview video will have all the frames of the video. This preview video is being used in order to specify the roads and draw a violation traffic line on the road near traffic signals. The traffic line violation that is the line drawn by the user will be acting as a traffic signal line. To enable the line drawing feature, we need to select the ‘Region of interest’ item from the ‘Analyze’ option. After that administrator will need to select two points to draw a line that specifies the traffic signal.

In the violation detection system, we select the region of interest and it starts the process. When we draw the line violation line, it will be displayed on the console. After the line is drawn, this violation detection system will be started immediately. At first the weights will be loaded. Afterwards the detection system will play its role and complete the detection of objects process. GUI i.e. Graphical User interface will display the output frame to frame subsequently.

The output will be shown by the violation detection system till the last frame of the footage. In the background an ‘output.mp4’ will be generated. The file will be in the ‘output’ folder of ‘Resources’. To terminate the process we should click ‘q’.

So, finally after the processing of the existing one CCTV video footage, the user can also insert another video footage from the file manager. If the work is complete the user can quit using the ‘Exit’ item from File option.

8.3. License Plate Recognition:

8.3.1. Overview:

1) Read in an image into python using opencv
2) Apply filtering, edge detection use contour search to find plates
3) Extracting number plate text using ocr with easy ocr

8.3.2. Implementation Steps:

1) Install and import dependencies

Easyocr : EasyOCR is a dedicated python package/module that consists of the feature that conversion of image to text can be done.

Imutils : These are the functions used in python which play a major role in defining and aligning the image processing functionalities that rae rotation of the image, translation of the image, resizing of the image, displaying all the matplot library will be more easier with OpenCV and python 3. This can also be used for skeletonization of the code as well.

2) Read in Image, Grayscale and Blur

```
img = cv2.imread('image4.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.imshow(cv2.cvtColor(gray, cv2.COLOR_BGR2RGB))
```

Initially the image is loaded using imread, next we use cvtColor to recolor the image and convert it into a grey image that is done by passing the color code converter cvtColor(gray, cv2.COLOR_BGR2GRAY). So here we change the image color which is initially in BGR to GREY.

3) Apply filter and find edges for localization

This is for Noise Reduction & Edge Detection:

```
bfilter = cv2.bilateralFilter(gray, 11, 17, 17)
edged = cv2.Canny(bfilter, 30, 200)
plt.imshow(cv2.cvtColor(edged, cv2.COLOR_BGR2RGB))
```

Here we are applying filtering to remove noise from the image and edge detection to find the edges in the image. For filtering we use a bilateral filter method to pass the image and pass the properties which allow us to specify how intensely we want the noise reduction.
Then from that we use a canny algorithm to perform edge detection which has a set of parameters which can be tuned to get the ideal edges.

4) Find Contours and Apply Mask

```
keypoints = cv2.findContours(edged.copy(), cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
```

```
contours = imutils.grab_contours(keypoints)
contours = sorted(contours, key=cv2.contourArea, reverse=True)[:10]
```

We do contour detection that is detecting the polygons within the lines. Ideally the shapes in the images. In our case we identify rectangles as our number plate is in that shape.

First step we do is to find contours by using the method findContours to which we pass the edge image. And here we are approximating the required contour value by using cv2.CHAIN_APPROX_SIMPLE.

And then we store the contours by using function grab_contours and we sort the top 10 contours on the basis of contour area. Next we loop through these contours and find whether they are rectangle or square shaped. And by doing that we get the coordinates to the number plate.

```
mask = np.zeros(gray.shape, np.uint8)
new_image = cv2.drawContours(mask, [location], 0,255,-1)
new_image = cv2.bitwise_and(img, img, mask=mask)
```

Next we mask and isolate the section of the number plate using the coordinates. Here first we masked the entire grey image. And next we highlight only the number plate using drawContours function where it is given the location of the number plate as parameter to it. And in the final step we bitwise_and to put up the number plate on to the masked image.

Next we isolate the segment which contains the number plate by cropping that part. So here we find the coordinates of every section where our image is in black and crop the image.

```
(x,y) = np.where(mask==255)
(x1, y1) = (np.min(x), np.min(y))
(x2, y2) = (np.max(x), np.max(y))
cropped_image = gray[x1:x2+1, y1:y2+1]
```

5. Use Easy OCR To Read Text

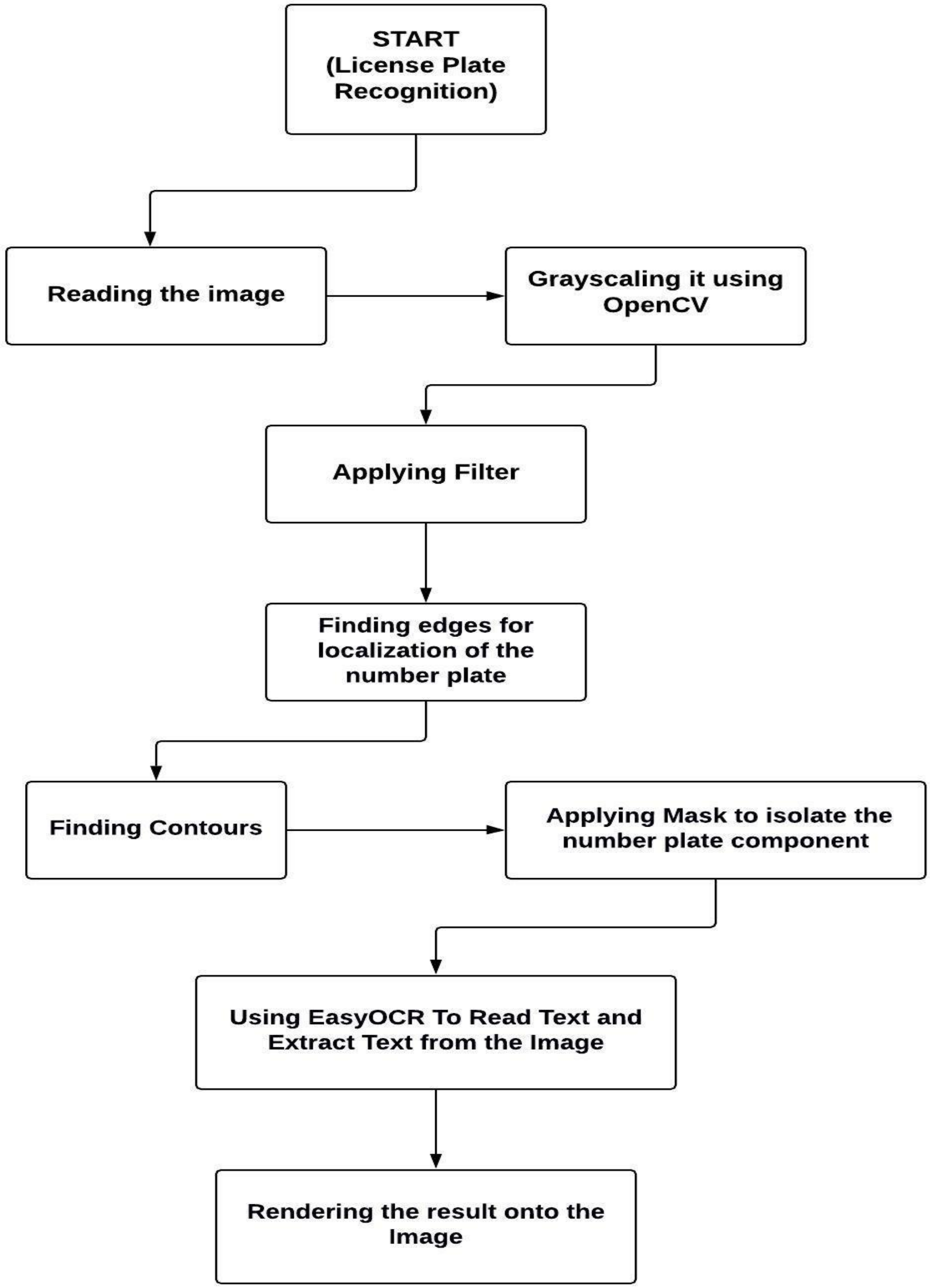
```
reader = easyocr.Reader(['en'])
result = reader.readtext(cropped_image)
```

In the above step we use easy ocr to read the text from the image.

6. Render Result

In this part we take our number plate and overlay our detection on the original image.

Below is the complete flowchart of the working of license plate recognition:


START
(License Plate
Recognition)
Reading the image
Grayscaling it using
OpenCV
Applying Filter
Finding edges for
localization of the
number plate
Finding Contours
Applying Mask to isolate the
number plate component
Using EasyOCR To Read Text and
Extract Text from the Image
Rendering the result onto the
Image

8.4. Detection of non-helmet riders (Helmet Detection) :

8.4.1 Libraries used overview:

Glob: This module is used to retrieve files/path names matching a specified pattern.

OS: This module is used in Python that will provide functionalities for interaction with the operating system. OS module is one of the standard utility modules in python.

Argparse: This python module is used to generate help and issues messages and errors too whenever the user provides invalid arguments in the program.

OpenCV: This module mainly focuses on image/video processing and analysis, it helps to solve computer vision problems.

Shutil : This module in python is useful in automation of the process i.e. copying and removing of all the files and directories present.

8.4.2. Implementation Steps:

1. Load names of classes:

From obj.names and store it in the list called classes.

```
classesFile = r"C:\Users\revan\Documents\yolov3-Helmet-Detection\obj.names";
classes = None
with open(classesFile, 'rt') as f:
classes = f.read().rstrip('\n').split('\n')
```

2. Load the config and weights:

load configuration and weights from the directories and storing respective.

```
modelConfiguration ="C:\Users\revan\Documents\yolov3-Helmet-Detection\yolov3-obj.cfg";
modelWeights r"C:\Users\revan\Documents\yolov3-Helmet-Detection\yolov3-
obj_2400.weights"
```

3. Preparing network by the weights and config which are loaded by above code:

```
net = cv.dnn.readNetFromDarknet(modelConfiguration, modelWeights)
net.setPreferableBackend(cv.dnn.DNN_BACKEND_OPENCV)
net.setPreferableTarget(cv.dnn.DNN_TARGET_CPU)
```

4. Defined Functions :

This below functions helps to get names of the output layers for the network net

Retrieving the names of all the output layers:

```
def getOutputsNames(net):
layersNames = net.getLayerNames()
```

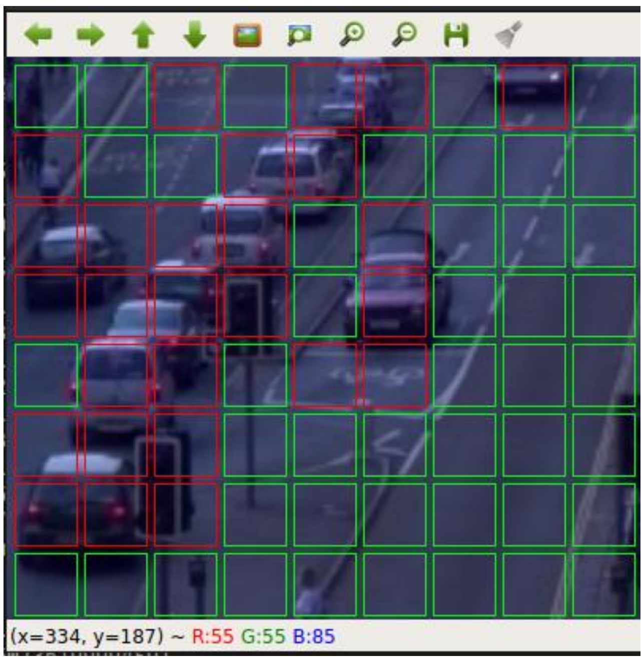
The drawPredit function helps to draw the bounding box for the person who is wearing the helmet and this function is called by post process function. We define Post Process function below removes the bounding boxes with low confidence using non-maxima suppression.It

scans through all the bounding boxes output from the network and keeps only the ones with high confidence scores. All the assigning of the class label of box with the class of the highest score.

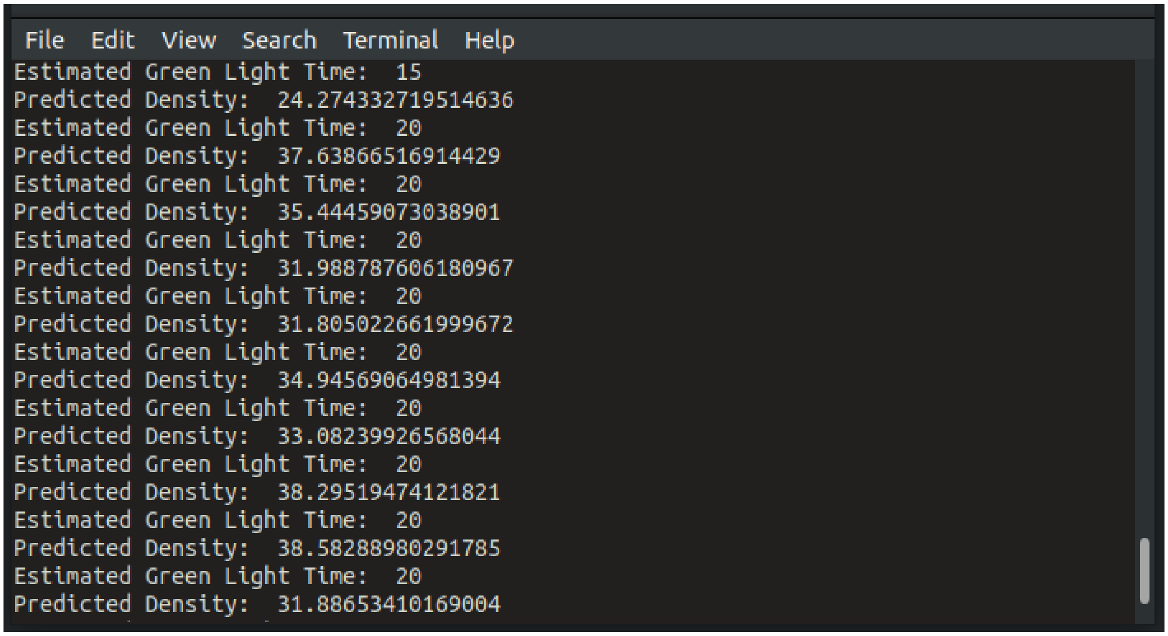
CHAPTER - 9

RESULTS AND DISCUSSION

9.1. TRAFFIC LIGHT AUTOMATION :



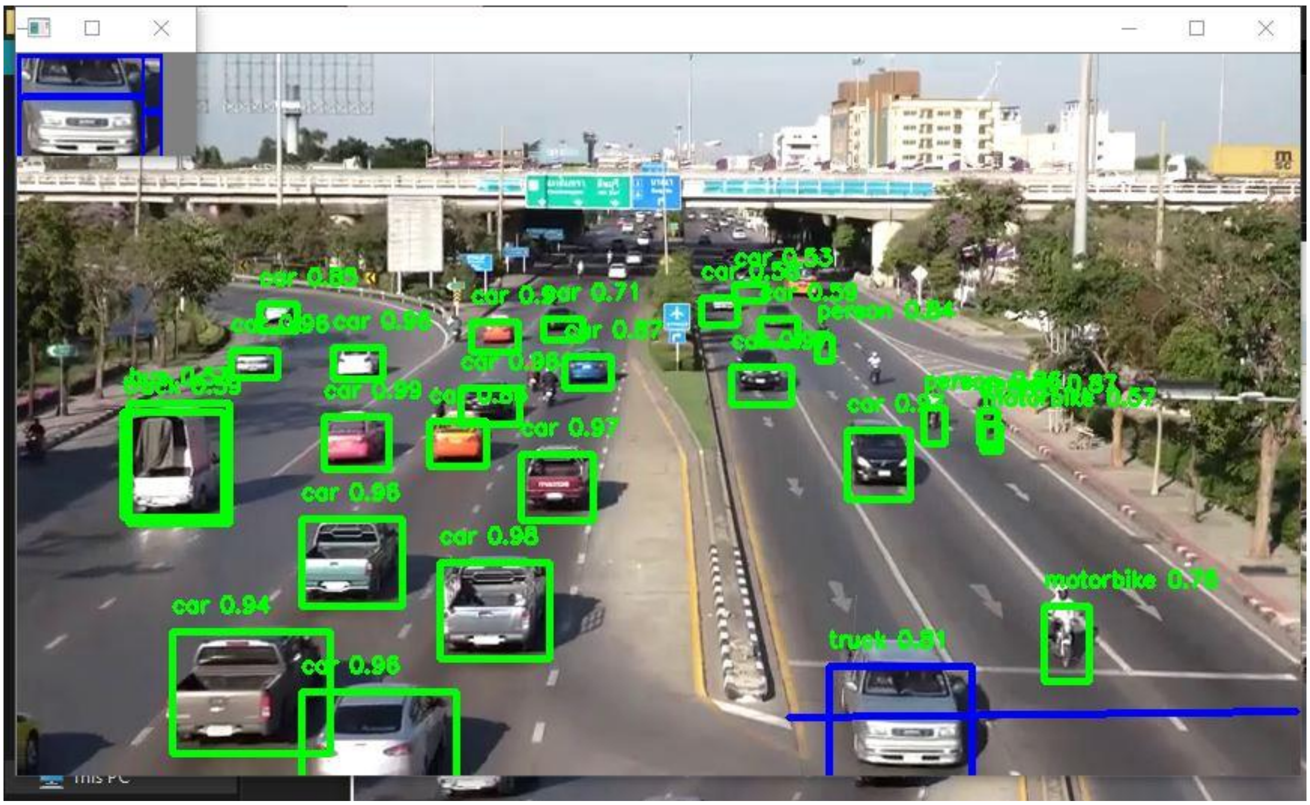
This is how the result appears after the grid split and after the model has classified the grid.
The red coloured grids are supposed to be grids containing traffic vehicles and the green coloured grids are ones that do not contain.



The model gives us an accuracy of 90-100 range now. For each frame, the density is stored and then forecasted. And after each time interval, the predicted density and green light time is displayed.

9.2. Line Violation:

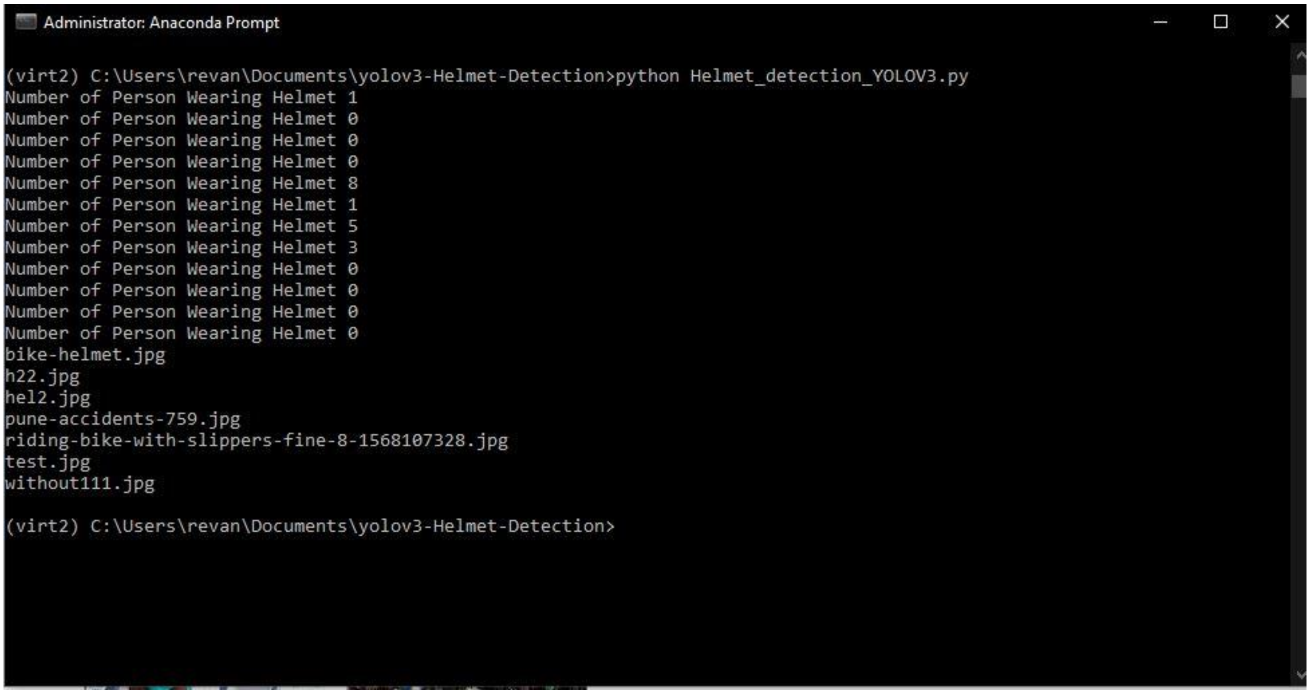
In the below image the moving objects are detected. An object detection model YOLOv3 is used to classify these moving objects into respective classes.After detecting the vehicles, violation cases are checked. A straight line called the traffic violation line is drawn on the road near the signal in the CCTV footage or the camera footage from the junction. The crossing of this line during red traffic lights is treated as a violation.



If any vehicle passes the traffic light in red state, violation happens. And the image of the vehicle is captured and stored. And these images are sent as an input to the License Plate Detection Model.

9.3. Helmet Detection:

The image below gives information about the number of persons wearing helmets and the captured images of people not wearing any helmet.

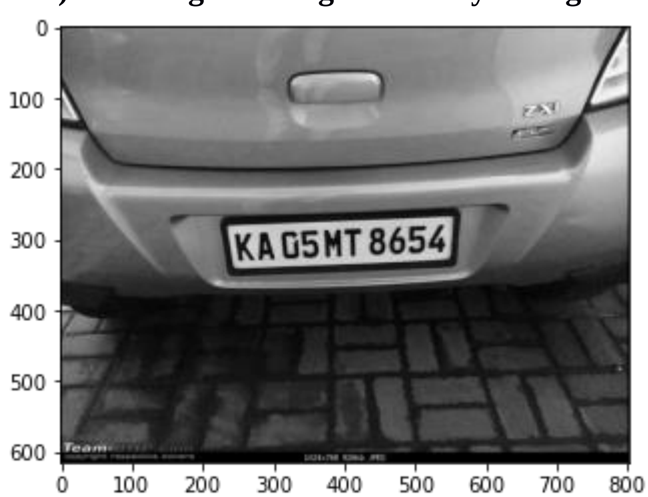


In the below image as we can see people wearing helmets are detected and classified in helmet class with the confidence value shown.

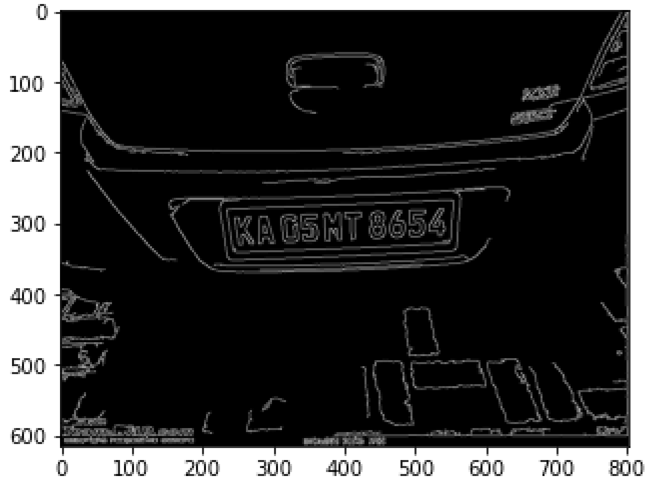
Riders who are not wearing helmets are violating the rule and these violated vehicles captured images are stored under a detected images folder. And these images are sent as an input to the License Plate Detection Model.

9.4. License Plate Recognition: It is achieved by following the 5 steps given below

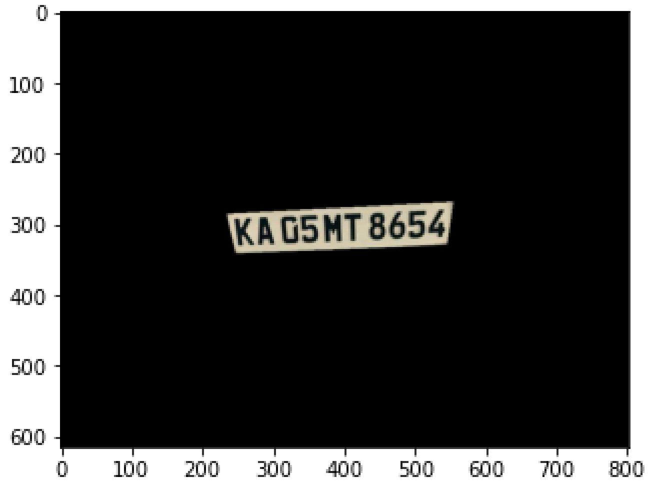
1) Reading the Image and Grayscaling it using opencv



2. Applying filter and finding edges for localization of the number plate



3. Finding Contours and Applying Mask to isolate the number plate component



4. Using EasyOCR To Read Text and Extract Text from the Image :

5. Rendering the result onto the Image

CHAPTER – 10

CONCLUSION & FUTURE WORK

In order to overcome the problems faced by the traffic police and the public regarding the traffic control system, we have come up with a better solution that uses image processing techniques, scheduling algorithms which is possibly optimal and better than many other existing systems. We have built a system that detects the potential traffic violations and automates the traffic light signal controller. By using all the actual real time images as an input, this method is consistent in its own ways of vehicle detection and also much better than already existed systems.

The project can be improved in a lot of aspects and more work can be put into place such as a report can be generated to the vehicles which violate the traffic rules with detailed information about the rule violated, the place and the video footage for reference. And we should make sure the website should be fault tolerant.

BIBLIOGRAPHY/REFERENCES

[1] Pandit, V., Doshi, J., Mehta, D., Mhatre, A. and Janardhan, A., 2014. Smart traffic control system using image processing. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, *3*(1), pp.2278-6856

[2] Khanke, P. and Kulkarni, P.S., 2014. A technique on road traffic analysis using image processing. *International Journal of Engineering Research and Technology*, *3*, pp.2769-2772.

[3] Soman, R. and Radhakrishnan, K., 2018. Traffic Light Control and Violation Detection Using Image Processing. *Traffic*, *8*(4).

[4] Narkhede, A., Nikam, V., Soni, A., Sathe, A. and Chiddarwar, G.G., Automatic Traffic Rule Violation Detection and Number Plate Recognition

[5] Zaatouri, K. and Ezzedine, T., 2018, December. A Self-Adaptive Traffic Light Control System Based on YOLO. In *2018 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC)* (pp. 16-19). IEEE.

[6] Saha, Satadal & Basu, Subhadip & Nasipuri, Mita & Basu, Dipak. (2009). Development of an automated Red Light Violation Detection System (RLVDS) for Indian vehicles.

APPENDIX A

Abbreviations:

1. WAN : Wireless Area Networks
2. RTI : Right To Information
3. YOLO : You Only Look Once
4. k-NN : ‘k’ Nearest Neighbours
5. SVM : Support Vector Machines
6. ANPR : Automatic Number Plate Recognition
7. DIP : Digital Image Processing
8. HOG : Histogram of Oriented Gradients
9. LBP : Local Binary Patterns
10. ARIMA : Auto Regressive Integrated Moving Average
11. LSTM : Long Short Term Memory