A

Major Project

On

ADVANCED CONTROL OF TRAFFIC LIGHT SYSTEM

(Submitted in partial fulfilment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

S Shruthi (177R1A0554)

C. Sairam (177R1A0505)

Anshika Gupta (177R1A0502)

Under the Guidance of

Mr. K. KARUNAKAR

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under 2(f) & 12(B) of the UGC Act. 1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401

2017-2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**ADVANCED CONTROL OF TRAFFIC LIGHT SYSTEM**" being submitted by **S Shruthi(177R1A0554)**, **C. Sairam(177R1A0505)** & **Anshika Gupta(177R1A0502)** in partial fulfilment of the requirements for the award of the degree of B. Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad , is a record bonafide work carried out by him/her under our guidance and supervision during the year 2020-2021.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of degree or diploma.

Mr. K. Karunakar
Assistant Professor
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
Head of the Department

EXTERNAL EXAMINAR

Submitted by viva voce Examination held on ___________________________________

ACKNOWLEDGEMENT

Apart from the efforts of us , the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We take this opportunity to express profound gratitude and deep regard to our guide.

Mr. K. Karunakar, Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to project review committee (PRC) Coordinators **Mr. J. Narasimha Rao**, **Mr. B. P. Deepak Kumar**, **Ms. V. Sandhya** for their cordial support , valuable information and guidance ,which helped us in completing this task through various stages.

We are also thankful to **Dr . K. Srujan Raju**, Head of Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr . A. Raji Reddy** , Director for being cooperative throughout the course of this project. We would like to express our sincere gratitude to Sri. **Ch. Gopal Reddy**, Chairman for providing excellent infrastructure and nice atmosphere throughout the course of this project.

The guidance and support received from all the members of CMR Technical Campus who contributed to the completion of project . We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed . We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project .

S SHRUTHI(177R1A0554)

C. SAIRAM(177R1A0505)

ANSHIKA GUPTA(177R1A0502)

ABSTRACT

This project titled as "Advanced Control of Traffic Light System" is an application based on Artificial Intelligence. This project aims to design a traffic light controller based on Computer Vision that can adapt to the current traffic situation. It uses live video feed from the CCTV cameras at traffic junctions for real-time traffic density calculation by detecting the vehicles at the signal and setting the green signal time accordingly. We then set the timers of these traffic signals according to vehicle density in each direction and hence the system becomes adaptive. This helps to optimize the green signal times, and traffic is cleared at a much faster rate than a static system, thus reducing the unwanted delays, congestion, and waiting time, which in turn will reduce the fuel consumption and pollution. Hence this project is really beneficial when it comes to dealing with heavy traffic in major cities.

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
Figure 3.1	Project Architecture	10
Figure 3.2	YOLO Architecture	11
Figure 3.3	Use Case Diagram	14
Figure 3.4	Class Diagram	15
Figure 3.5	Sequence Diagram	16
Figure 3.6	Activity Diagram	17

LIST OF SCREENSHOTS

SCREENSHOT NO.	SCREENSHOT NAME	PAGE NO
Screen 5.1.1	Signal timers cycle img1	27
Screen 5.1.2	Signal timers cycle img1	27
Screen 5.2.1	Simulation Screenshot1	28
Screen 5.2.2	Simulation Screenshot2	28

TABLE OF CONTENTS

ABSTRACT	**i**
LIST OF FIGURES	**ii**
LIST OF SCREENSHOTS	**iii**
1. INTRODUCTION	**1**
1.1 PROJECT SCOPE	2
1.2 PROJECT PURPOSE	2
1.3 PROJECT FEATURES	2
2. SYSTEM ANALYSIS	**3**
2.1 PROBLEM DEFINITION	4
2.2 EXISTING SYSTEM	4
2.2.1 LIMITATIONS OF EXISTING SYSTEM	5
2.3 PROPOSED SYSTEM	5
2.3.1 ADVANTAGES OF PROPOSED SYSTEM	6
2.4 FEASIBILITY STUDY	6
2.4.1 ECONOMIC FEASIBILITY	6
2.4.2 TECHNICAL FEASIBILITY	7
2.4.3 SOCIAL FEASIBILITY	7
2.5 HARDWARE AND SOFTWARE REQUIREMENTS	8
2.5.1 HARDWARE REQUIREMENTS	8
2.5.2 SOFTWARE REQUIREMENTS	8
3. ARCHITECTURE	**9**
3.1 PROJECT ARCHITECTURE	10
3.2 MODULES DESCRIPTION	10
3.2.1 VECHICLE DETECTION MODULE	10
3.2.2 SIGNAL SWITCHING ALGORITHM	12
3.2.2.1 WORKING OF THE ALGORITHM	13
3.2.3 SIMULATION MODULE	13
3.3 USE CASE DIAGRAM	14

	3.4 CLASS DIAGRAM	15
	3.5 SEQUENCE DIAGRAM	16
	3.6 ACTIVITY DIAGRAM	17
4.	**IMPLEMENTATION**	**18**
	4.1 INTRODUCTION	19
	4.2 SAMPLE CODE	19
5.	**SCREENSHOTS**	**26**
	5.1 SIGNAL TIMERS	27
	5.2 SIMULATION	28
6.	**TESTING**	**29**
	6.1 INTRODUCTION TO TESTING	**30**
	6.2 TYPES OF TESTING	30
	6.2.1 UNIT TESTING	30
	6.2.2 INTEGRATION TESTING	30
	6.2.3 FUNCTIONAL TESTING	31
	6.3 TEST CASES	31
	6.3.1. CURRENT SYSTEM VS PROPOSED SYSTEM PERFORMANCE RESULT TABLE	32
7.	**CONCLUSION AND FUTURE SCOPE**	**33**
	7.1 PROJECT CONCLUSION	34
	7.2 FUTURE SCOPE	34
8.	**BIBLIOGRAPHY**	**35**
	8.1 REFERENCES	36
	8.2 WEBSITES	37

1.INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

This project titled as “Advanced Control of Traffic Light System is an Artificial Intelligence based application. This project aims to design a traffic light controller based on Computer Vision that can adapt to the current traffic situation. It uses live video feed from the CCTV cameras at traffic junctions for real-time traffic density calculation by detecting the vehicles at the signal and setting the green signal time accordingly.

1.2 PROJECT PURPOSE

The congestion of urban traffic is becoming one of the critical issues with increasing population and automobiles in cities. Although it seems to pervade everywhere, megacities are the ones most affected by it. And it's ever increasing nature makes it imperative to know the road traffic density in real-time for better signal control and effective traffic management. The current traffic management systems that are in place are generally static, which means that they do not adjust according to the needs of the traffic flow. Our proposed system aims to design a traffic light controller based on Computer Vision that can adapt to the current traffic situation.

1.3 PROJECT FEATURES

Our proposed system aims to design a traffic light controller based on Computer Vision that can adapt to the current traffic situation. We have used object detection techniques like YOLO in order to detect the number of vehicles for each direction. We then set the timers of these traffic signals according to vehicle density in each direction and hence the system becomes adaptive. This helps to optimize the green signal times, and traffic is cleared at a much faster rate than a static system, thus reducing the unwanted delays, congestion, and waiting time, which in turn will reduce the fuel consumption and pollution.

2.SYSTEM ANALYSIS

2.SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analysed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

A detailed study of the process must be made by various techniques like Object detection which helps to detect vehicles, methods to get optimized signal time etc. The data collected by - sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

2.2 EXISTING SYSTEM

The current traffic management systems that are in place are generally static, which means that they do not adjust according to the needs of the traffic flow. There are three traffic management systems used mostly and they are:

❑ **Manual Controlling**: Manual controlling is the same as the name suggests requires manpower to control the traffic. The traffic police are allotted for a required area to control traffic. The traffic police carry signboards , sign light and whistle to control the traffic.

❑ **Automatic Controlling**: Automatically traffic light is controlled by timers and electrical sensors. In traffic light, a constant numerical value loaded in the timer. The timer automatically gets ON and OFF based on the timer value.

❑ **Electronic Sensors**: Another advanced method is placing some loop detectors or proximity sensors on the road . This sensor gives data about the traffic on the road . According to the sensor data the traffic signals are handled.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

- The manual controlling system requires a large number of manpower.
- Conventional traffic lights uses a timer for every phase, which is fixed and does not adapt according to the real-time traffic on that road
- Electronic sensors i.e, proximity sensors or loop detectors, accuracy and coverage are often in conflict because the collection of high-quality information is usually based on sophisticated and expensive technology and thus limited budget will reduce the number of facilities.

2.3 PROPOSED SYSTEM

The traffic has no specific pattern that is followed , and the static signal timers pose a huge problem to the already critical condition of congestion. Therefore, implementing a system which aims to reduce chances of such scenarios by automatically computing the optimal green signal time based on the current traffic at the signal will ensure that the direction with more traffic is allotted green signal for longer duration of time as compared to the direction with lesser traffic .

This system can override the older system of hard coded lights which cause unwanted delays reducing congestion and waiting time which will reduce the number of accidents and fuel consumption which will in turn help in controlling air pollution.

The main objective of this project is to design a traffic light controller based on Computer Vision that can adapt to the current traffic situation. Our proposed system aims to use live video feed from the CCTV cameras at traffic junctions for real-time traffic density calculation by detecting the vehicles at the signal and set the green signal time accordingly. The vehicles are classified as car, bike, bus/truck ,or rickshaw to obtain a more accurate estimate of the green signal time.

2.3.1 ADVANTAGES OF PROPOSED SYSTEM

- Real time traffic light switching according to the current traffic density .
- Virtually no new hardware to be installed.
- Less expensive than sensors.
- Autonomous: No need of manpower.

2.4 FEASIBILITY STUDY

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company.

Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 ECONOMIC FEASIBILTY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require. The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. As the cameras are preinstalled at signals in many major cities, timers are available and also there are surveillance centres are also present which means these centres can be used as for running the algorithm to calculate signal time. Hence, the system can be developed because of its economic feasibility.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 SOCIAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioural aspects are considered carefully and conclude that the project is socially feasible.

2.5 HARDWARE AND SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- Processor : Intel Dual Core@ CPU 2.90GHz.
- Hard disk : 20GB and above.
- RAM : 4GB and above.
- Monitor : 5 inches or above

Camera feed and traffic lights already being installed in real time.

2.5.2 SOFTWARE REQUIREMENTS

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements.

- Operating system : Windows 7 or above, Mac OS, Linux
- Languages : Python
- Backend : Artificial Intelligence (superset of deep learning and machine learning)
- IDE : Visual Studio C++ Build Tools, or Git using Git CMD

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for using artificial intelligence , starting from input images to setting optimal green signal time.

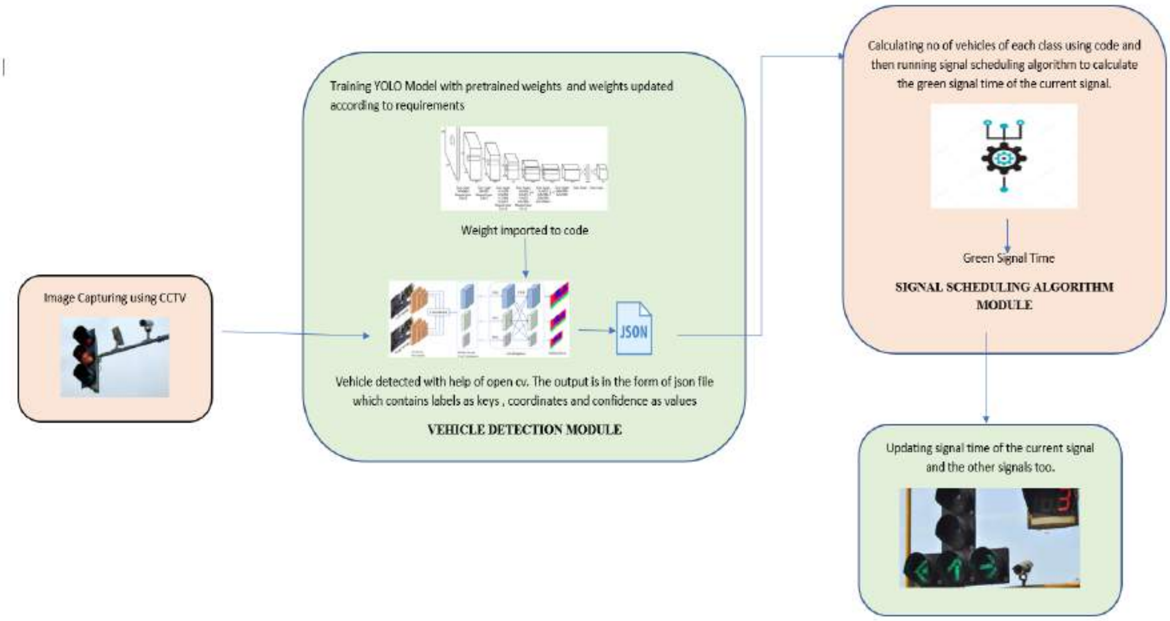


Figure 3.1 : Project Architecture of Advanced Control of Traffic Light System

3.2 MODULES DESCRIPTION

In our project we have three modules namely:

- Vehicle Detection Module
- Signal Switching Algorithm
- Simulation Module

3.2.1 VEHICLE DETECTION MODULE

This module deals with detecting the number of vehicles of each class at the present signal by using the image captured at that signal

- The proposed system uses YOLO (You only look once) for vehicle detection, which provides the desired accuracy and processing time. A custom YOLO model was trained for vehicle detection, which can detect vehicles of different classes like cars, bikes, heavy vehicles (buses and trucks), and rickshaws.

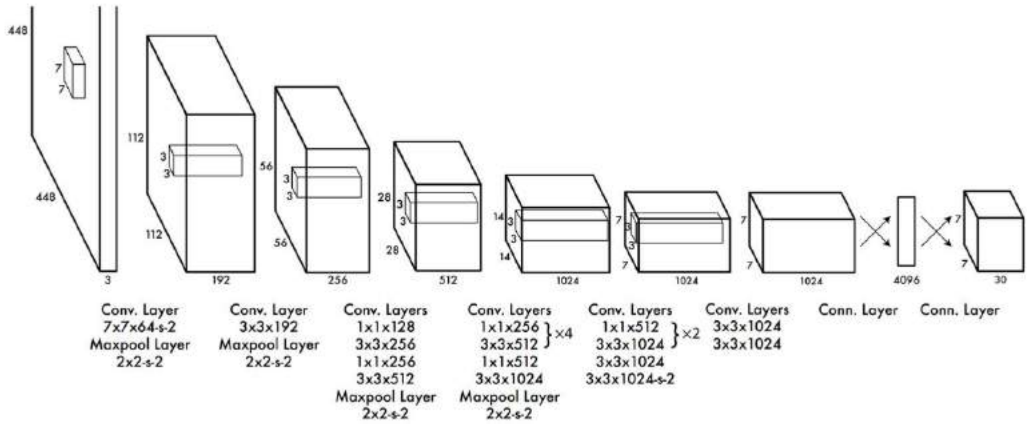


Figure 3.2 : Architecture of YOLO (You Only Look Once) Model

- The dataset for training the model was prepared by scraping images from google and labelling them manually using LabelIMG, a graphical image annotation tool.
- Then the model was trained using the pre-trained weights downloaded from the YOLO website. The configuration of the .cfg file used for training was changed in accordance with the specifications of our model. The number of output neurons in the last layer was set equal to the number of classes the model is supposed to detect by changing the 'classes' variable. In our system, this was 4 viz. Car, Bike, Bus/Truck, and Rickshaw. The number of filters also needs to be changed by the formula 5*(5+number of classes), i.e., 45 in our case.
- After making these configuration changes, the model was trained until the loss was significantly less and no longer seemed to reduce. This marked the end of the training, and the weights were now updated according to our requirements.
- These weights were then imported in code and used for vehicle detection with the help of OpenCV library. A threshold is set as the minimum confidence required for successful detection. After the model is loaded and an image is fed to the model, it gives the result in a JSON format i.e., in the form of key-value pairs, in which labels are keys, and their confidence and coordinates are values. Again, OpenCV can be used to draw the bounding boxes on the images from the labels and coordinates received.

3.2.2 SIGNAL SWITCHING ALGORITHM

The Signal Switching Algorithm sets the green signal timer according to traffic density returned by the vehicle detection module, and updates the red signal timers of other signals accordingly. It also switches between the signals cyclically according to the timers.

The following factors were considered while developing the algorithm:

1. The processing time of the algorithm to calculate traffic density and then the green light duration – this decides at what time the image needs to be acquired
2. Number of lanes
3. Total count of vehicles of each class like cars, trucks, etc.
4. The average speed of each class of vehicle when the green light starts i.e. the average time required to cross the signal by each class of vehicle .
5. The minimum and maximum time limit for the green light duration - to prevent starvation.

The formula for calculating Green Signal Time is:

$$GST = \frac{\sum_{vehicleClass} (NoOfVehicles_{vehicleClass} * AverageTime_{vehicleClass})}{(NoOfLanes + 1)}$$

Where:

- GST : Green Signal Time
- NoOfVehicles: This is the number of vehicles of each class at the signal as detected by vehicle detection module
- AverageTime: This is the average time the vehicles of that class take to cross an intersection.
- NoOfLanes : This is the number of lanes at the intersection.

3.2.2.1 WORKING OF THE ALGORITHM:

When the algorithm is first run, the default time is set for the first signal for the first cycle and the times for all the other signals. A separate thread is started which handles the detection of vehicles for each direction and the main thread handles the timer of the current signal.

When the green light of the current signal reaches 0 seconds, the detection threads take the snapshot of the next direction. The result is then parsed and the timer of the next green signal is set .

All this happens in the background while the main thread is counting down the timer of the current green signal. Once the green timer of the current signal becomes zero , the next signal becomes green for the amount of time set by the algorithm and this process continues.

The signals switch in a cyclic fashion and not according to the densest direction first. This is in accordance with the current system where the signals turn green one after the other in a fixed pattern and does not need the people to alter their ways or cause any confusion. The order of signals is also the same as the current system, and the yellow signals have been accounted for as well.

3.2.3 SIMULATION MODULE

A simulation was developed from scratch using Pygame to simulate real-life traffic. It assists in visualizing the system and comparing it with the existing static system. It contains a 4-way intersection with 4 traffic signals. Each signal has a timer on top of it, which shows the time remaining for the signal to switch from green to yellow, yellow to red, or red to green. Each signal also has the number of vehicles that have crossed the intersection displayed beside it. Vehicles such as cars, bikes, buses, trucks, and rickshaws come in from all directions. In order to make the simulation more realistic, some of the vehicles in the rightmost lane turn to cross the intersection. Whether a vehicle will turn or not is also set using random numbers when the vehicle is generated. It also contains a timer that displays the time elapsed since the start of the simulation.

3.3 USE CASE DIAGRAM

In the use case diagram we have basically two actors who are the CCTV Camera and the System . The CCTV Camera captures image at signal and sends the captured image to trained YOLO model .Whereas the System is responsible for training the YOLO model, calculating weights, detecting vehicles using openCv , Sending the vehicle detection data to the server , Running the Signal Switching Algorithm , Calculating green signal time using algorithm and then Updating the signal timers.

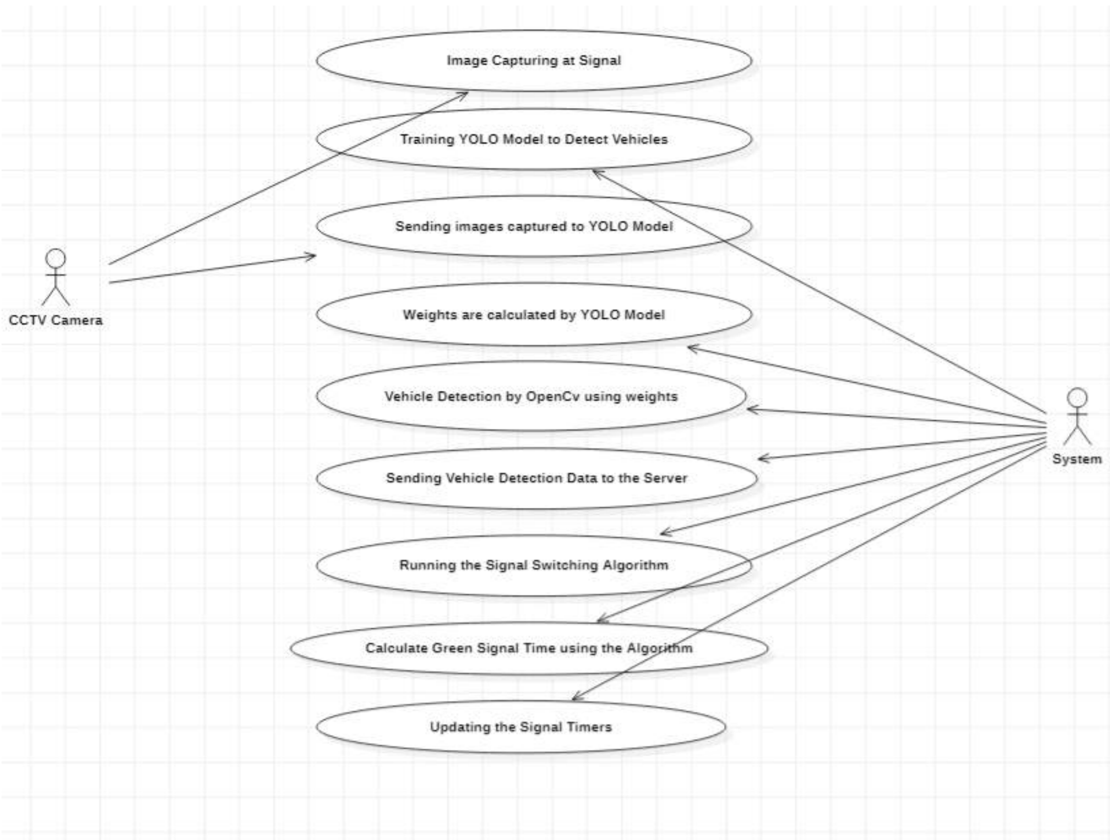


Figure 3.3: Use Case Diagram for Advanced Control of Traffic Light System

3.4 CLASS DIAGRAM

Class Diagram is a collection of classes and objects.

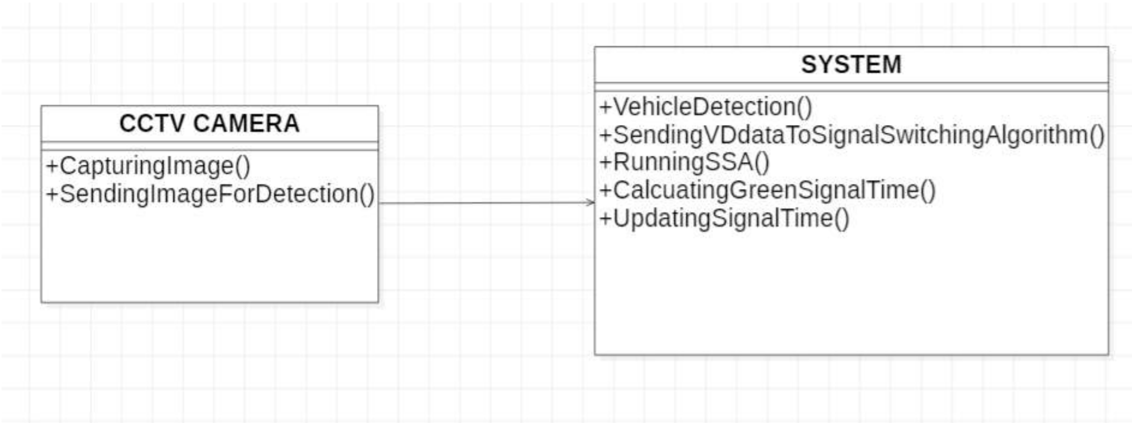


Figure 3.4: Class Diagram for Advanced Control of Traffic Light System

3.5 SEQUENCE DIAGRAM

Sequence diagram shows the sequence of actions that are followed up from the beginning to the end.

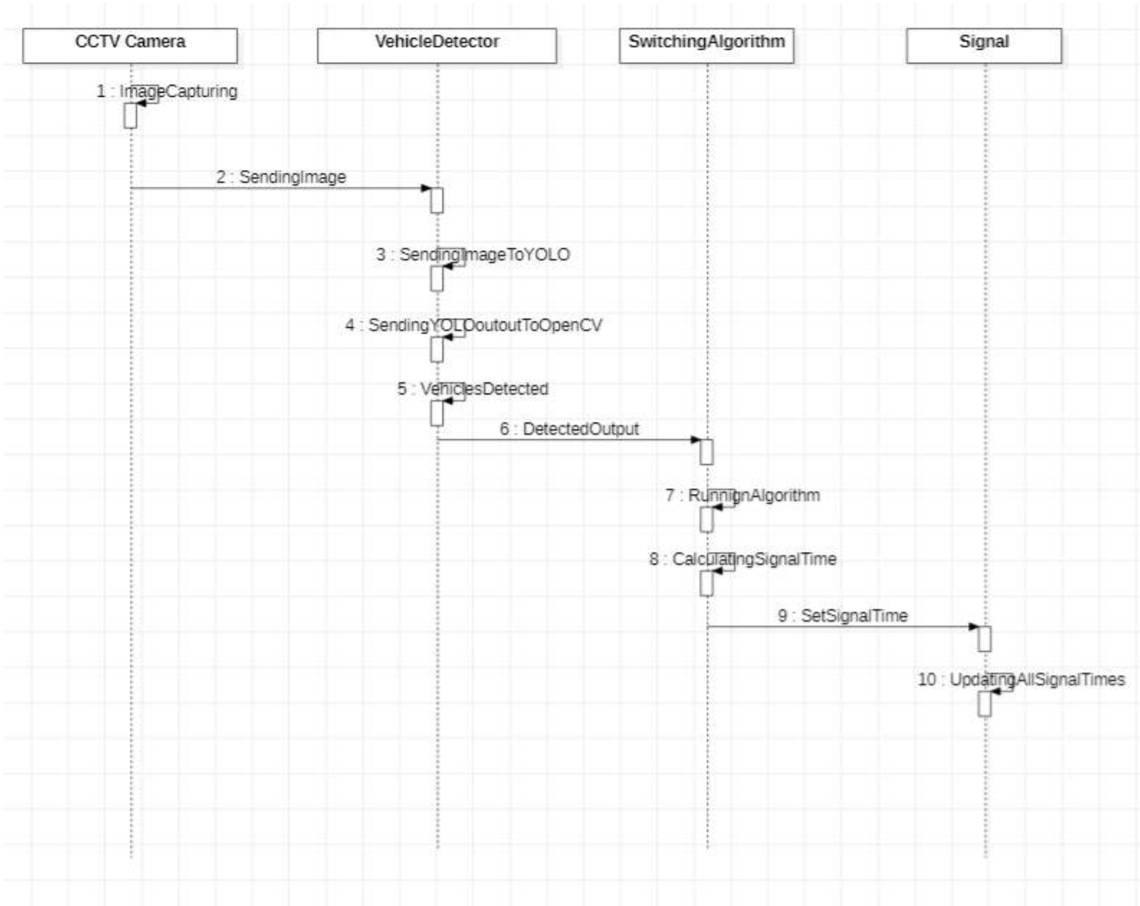


Figure 3.5 : Sequence Diagram for Advanced Control of Traffic Light System

3.6 ACTIVITY DIAGRAM

It describes about flow of activity states.

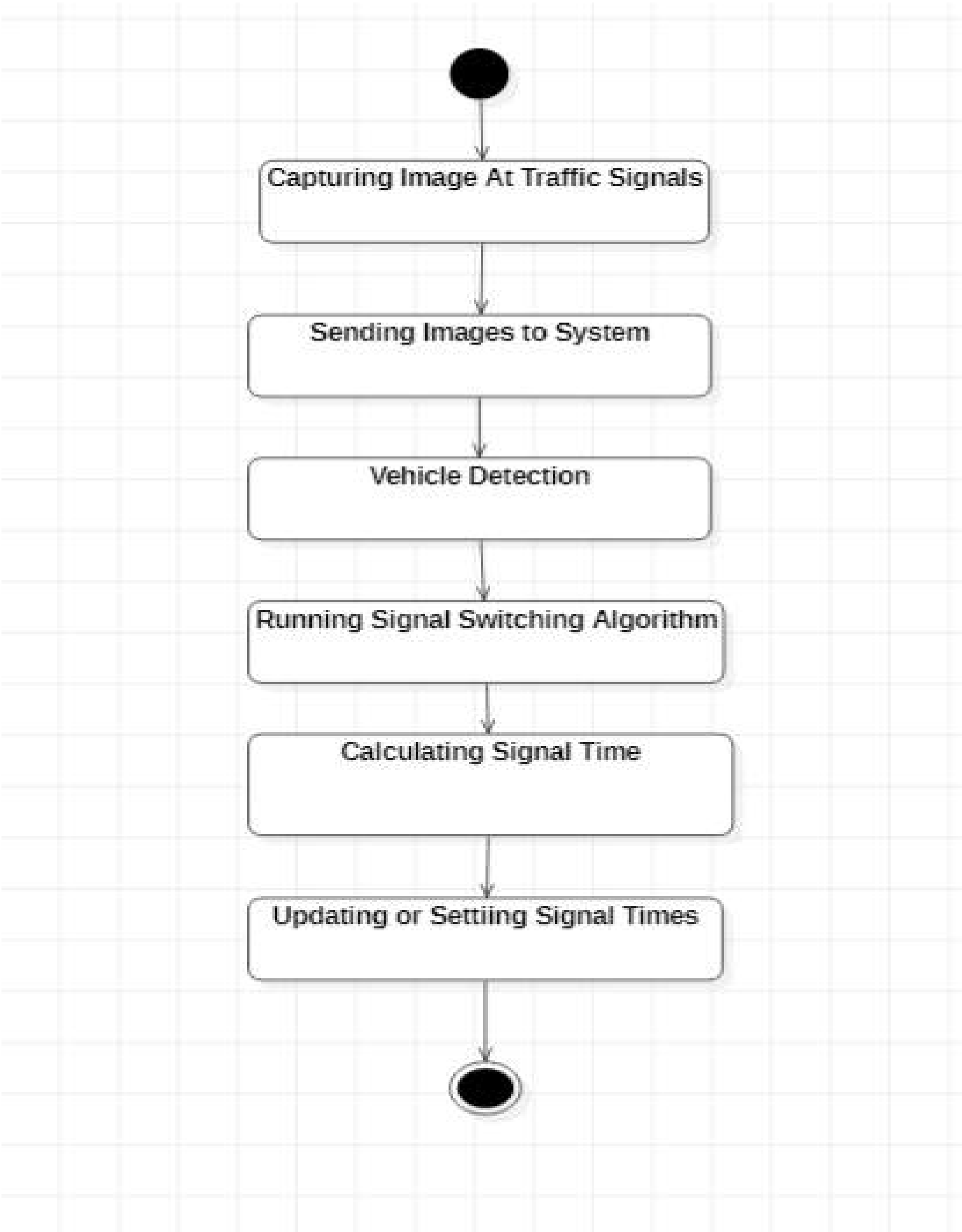


Figure 3.6 : Activity diagram for Advanced Control of Traffic Light System

4.IMPLEMENTATION

4. IMPLEMENTATION

4.1 INTRODUCTION

For implementing the algorithm, we have used python as a programming language. Python is an interpreter based high-level programming language. Python is a versatile language. It is mostly used for Data Science and Software Development. Python has gained popularity due to its ease of use and code readability. As a result, Python is widely used for Data Analysis, Natural Language Processing, and Computer Vision. Python comes with various graphical and statistical packages like Matplotlib, Numpy, SciPy and more advanced packages for Deep Learning such as TensorFlow, CV 2,Pygame etc. For the purpose of data mining, wrangling, visualizations and developing predictive models, we utilize Python. This makes Python a very flexible programming language.

4.2 SAMPLE CODE

simulation.py

```
# *** IMAGE XY COOD IS TOP LEFT
import random
import math
import time
import threading
from vehicle_detection import detection
import pygame
import sys
import os
# options={
#    'model':'./cfg/yolo.cfg',     #specifying the path of model
```

```
#    'load':'./bin/yolov2.weights',   #weights


speeds = {'car':2.25, 'bus':1.8, 'truck':1.8, 'rickshaw':2, 'bike':2.5}  # average speeds of vehicle

# Coordinates of start

x = {'right':[0,0,0], 'down':[755,727,697], 'left':[1400,1400,1400], 'up':[602,627,657]}

y = {'right':[348,370,398], 'down':[0,0,0], 'left':[498,466,436], 'up':[800,800,800]}


vehicles = {'right': {0:[], 1:[], 2:[], 'crossed':0}, 'down': {0:[], 1:[], 2:[], 'crossed':0}, 'left':
{0:[], 1:[], 2:[], 'crossed':0}, 'up': {0:[], 1:[], 2:[], 'crossed':0}}

    signals.append(ts3)

    ts4 = TrafficSignal(defaultRed, defaultYellow, defaultGreen, defaultMinimum,
defaultMaximum)

    signals.append(ts4)

    repeat()

# Set time according to formula

def setTime():

    global noOfCars, noOfBikes, noOfBuses, noOfTrucks, noOfRickshaws, noOfLanes

    global carTime, busTime, truckTime, rickshawTime, bikeTime

    os.system("say detecting vehicles, "+directionNumbers[(currentGreen+1)%noOfSignals])

#    detection_result=detection(currentGreen,tfnet)

#    greenTime = math.ceil(((noOfCars*carTime) + (noOfRickshaws*rickshawTime) +
(noOfBuses*busTime) + (noOfBikes*bikeTime))/(noOfLanes+1))

#    if(greenTime<defaultMinimum):

#       greenTime = defaultMinimum

#    elif(greenTime>defaultMaximum):
```

```
#       greenTime = defaultMaximum

    # greenTime = 
len(vehicles[currentGreen][0])+len(vehicles[currentGreen][1])+len(vehicles[currentGreen][2]
)

    # noOfVehicles = 
len(vehicles[directionNumbers[nextGreen]][1])+len(vehicles[directionNumbers[nextGreen]][
2])-vehicles[directionNumbers[nextGreen]]['crossed']

    # print("no. of vehicles = ",noOfVehicles)

    noOfCars, noOfBuses, noOfTrucks, noOfRickshaws, noOfBikes = 0,0,0,0,0

    for j in range(len(vehicles[directionNumbers[nextGreen]][0])):

        vehicle = vehicles[directionNumbers[nextGreen]][0][j]

        if(vehicle.crossed==0):

            vclass = vehicle.vehicleClass

            # print(vclass)

            noOfBikes += 1

  time.sleep(1)

    currentYellow = 1   # set yellow signal on

    vehicleCountTexts[currentGreen] = "0"

    # reset stop coordinates of lanes and vehicles

    for i in range(0,3):

        stops[directionNumbers[currentGreen]][i] = 
defaultStop[directionNumbers[currentGreen]]

        for vehicle in vehicles[directionNumbers[currentGreen]][i]:

            vehicle.stop = defaultStop[directionNumbers[currentGreen]]

    while(signals[currentGreen].yellow>0):  # while the timer of current yellow signal is not 
zero

        printStatus()
```

```
        updateValues()

        time.sleep(1)

    currentYellow = 0   # set yellow signal off

    # reset all signal times of current signal to default times

    signals[currentGreen].green = defaultGreen

    signals[currentGreen].yellow = defaultYellow

    signals[currentGreen].red = defaultRed

    currentGreen = nextGreen # set next signal as green signal

    nextGreen = (currentGreen+1)%noOfSignals    # set next green signal

    signals[nextGreen].red = signals[currentGreen].yellow+signals[currentGreen].green    # set
the red time of next to next signal as (yellow time + green time) of next signal

    repeat()

# Print the signal timers on cmd

def printStatus():

        for i in range(0, noOfSignals):

                if(i==currentGreen):

                        if(currentYellow==0):

                                print(" GREEN TS",i+1,"-> r:",signals[i].red,"
y:",signals[i].yellow," g:",signals[i].green)

                        else:

                                print("YELLOW TS",i+1,"-> r:",signals[i].red,"
y:",signals[i].yellow," g:",signals[i].green)

                else:

                        print("   RED TS",i+1,"-> r:",signals[i].red," y:",signals[i].yellow,"
g:",signals[i].green)

        print()
```

```
# Update values of the signal timers after every second
def updateValues():
    for i in range(0, noOfSignals):
        if(i==currentGreen):
            if(currentYellow==0):
                signals[i].green-=1
                signals[i].totalGreenTime+=1
   else:
            signals[i].red-=1


def simulationTime():
    global timeElapsed, simTime
    while(True):
        timeElapsed += 1
        time.sleep(1)
        if(timeElapsed==simTime):
            totalVehicles = 0
            print('Lane-wise Vehicle Counts')
            for i in range(noOfSignals):
                print('Lane',i+1,':',vehicles[directionNumbers[i]]['crossed'])
                totalVehicles += vehicles[directionNumbers[i]]['crossed']
            print('Total vehicles passed: ',totalVehicles)
            print('Total time passed: ',timeElapsed)
            print('No. of vehicles passed per unit time: ',(float(totalVehicles)/float(timeElapsed)))
```

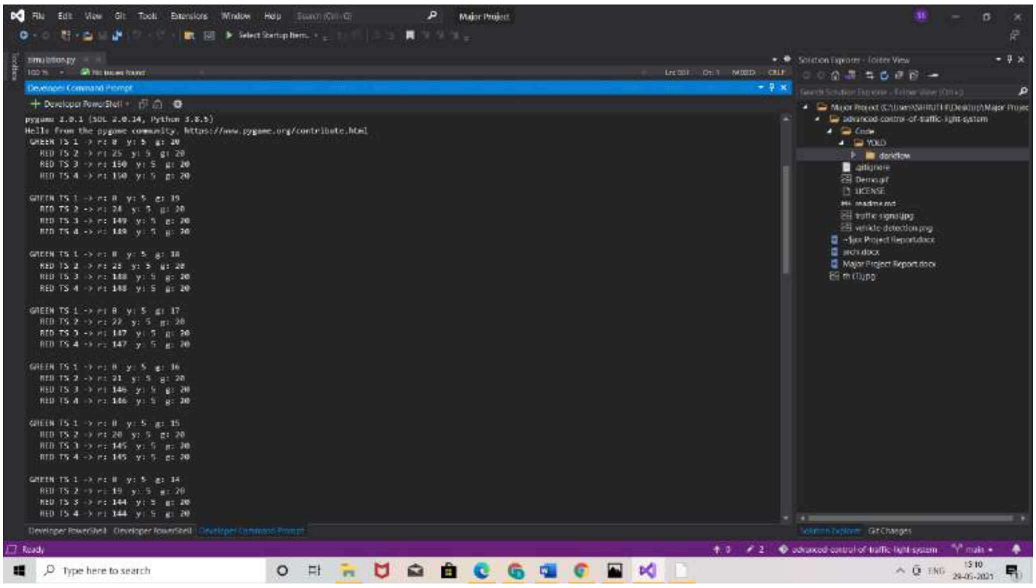
```
            os._exit(1)

class Main:

    thread4 = threading.Thread(name="simulationTime",target=simulationTime, args=())

    thread4.daemon = True

    thread4.start()

    thread2 = threading.Thread(name="initialization",target=initialize, args=())    # 
initialization

    thread2.daemon = True

    # Loading signal images and font

    redSignal = pygame.image.load('images/signals/red.png')

    yellowSignal = pygame.image.load('images/signals/yellow.png')

    greenSignal = pygame.image.load('images/signals/green.png')

    font = pygame.font.Font(None, 30)

    thread3 = threading.Thread(name="generateVehicles",target=generateVehicles, args=())    
# Generating vehicles

    thread3.daemon = True

    thread3.start()

    while True:

        for event in pygame.event.get():

            if event.type == pygame.QUIT:

                sys.exit()

        screen.blit(background,(0,0))   # display background in simulation

   for i in range(0,noOfSignals):  # display signal and set timer according to current status: 
green, yello, or red

            if(i==currentGreen):
```

```
            if(currentYellow==1):
                if(signals[i].yellow==0):
                    signals[i].signalText = "STOP"
                else:
                    signals[i].signalText = signals[i].yellow
                screen.blit(yellowSignal, signalCoods[i])
            else:
                if(signals[i].green==0):
                    signals[i].signalText = "SLOW"
                else:
                    signals[i].signalText = signals[i].green
                screen.blit(greenSignal, signalCoods[i])
        # display the vehicles
        for vehicle in simulation:
    screen.blit(vehicle.currentImage, [vehicle.x, vehicle.y])
            # vehicle.render(screen)
            vehicle.move()
        pygame.display.update()
Main()
```

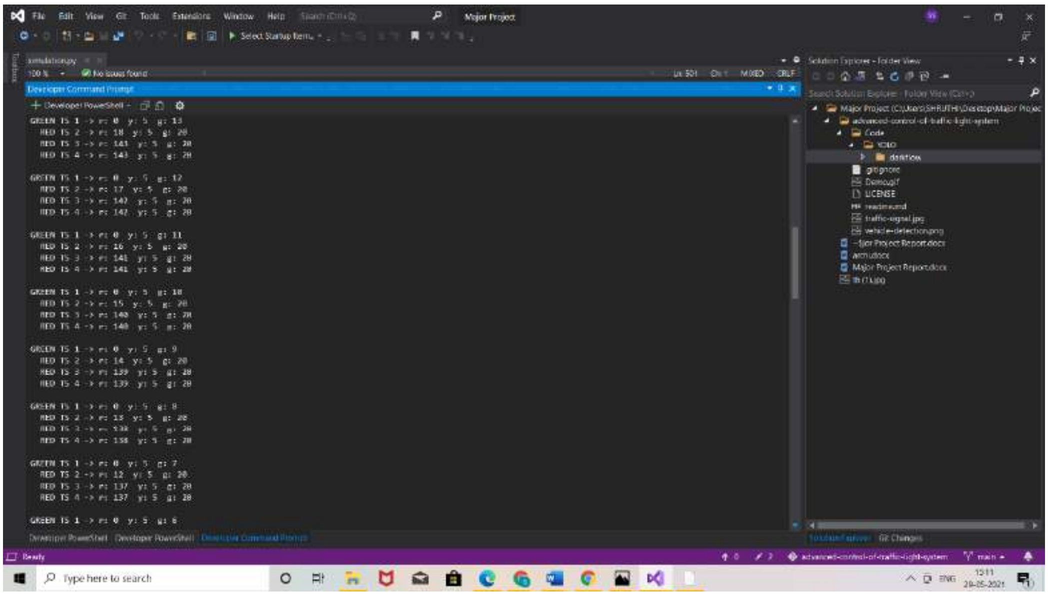
5.SCREENSHOTS

5. SCREENSHOTS

5.1 SIGNAL TIMERS



Screen 5.1.1: Signal timers cycle img1 for Advanced Control of Traffic light System



Screen 5.1.2: Signal timers cycle img2 for Advanced control of Traffic Light System

5.2 SIMULATION

Screen 5.2.1: Simulation screenshot1 for Advanced Control of Traffic Light System

Screen 5.2.2: Simulation screenshot2 for Advanced Control of Traffic Light System

6. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes.

6.3 TEST CASES

A Test Case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application.

We compare the total number of vehicles that cross the intersection over a period of 5 minutes in the current system and the proposed advanced system with same distribution of traffic, over a time of one hour with 12 simulation of 5 minutes each and different distributions of traffic.

The results obtained are tabulated in the form of number of vehicles lane-wise and total number of vehicles passed.

Current System					
Simulation No.	Lane 1	Lane 2	Lane 3	Lane 4	Total
1	67	74	51	18	210
2	78	73	47	19	217
3	80	73	33	29	215
4	76	71	39	27	213
5	77	66	44	26	213
6	74	72	37	21	204
7	65	73	36	18	192
8	60	68	33	28	189
9	49	83	36	28	196
10	57	70	46	25	198
11	53	70	39	34	196
12	55	70	29	38	192
					2435

Proposed Adaptive System					
Simulation No.	Lane 1	Lane 2	Lane 3	Lane 4	Total
1	111	86	42	31	270
2	105	83	38	28	254
3	100	96	36	20	252
4	96	75	56	22	249
5	93	89	42	24	248
6	77	97	37	30	241
7	76	82	48	30	236
8	71	92	48	30	241
9	85	98	48	31	262
10	79	92	37	30	238
11	110	105	24	17	256
12	76	87	43	44	250
					2997

Table 6.2.1: Current System vs Proposed System performance result table for Advanced Control of Traffic Light System

As it can be seen with all conditions alike the advanced system was able to pass 2997 vehicles ,but the current static system could pass only 2435 vehicles in one hour , which means 562 more vehicles. Thus , the proposed advanced system has an improved performance by about 23 percent.

7.CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

Our proposed system can override the older system of hard coded lights which cause unwanted delays reducing congestion and waiting time which will reduce the number of accidents and fuel consumption which will in turn help in controlling air pollution.

It sets the green signal time according to the traffic density at the signal and ensures that the direction with more traffic congestion is allotted green signal for longer duration of time as compared to the direction with lesser traffic.

Thus the proposed advanced system improves the performance by 23% over the current system in terms or number of vehicles crossing an intersection. This is a significant improvement and this adaptive system can be integrated with CCTV cameras in major cities in order to facilitate better management of traffic.

7.2 FUTURE SCOPE

The proposed system can be extended further by enabling more features and functionalities which would improve the way this works and also eradicate all the current problems completely. Some of the aspects that can be considered for future improvements that can be carried out in this project are:

Identifying vehicles with red lights: These vehicles may be a police vehicle, ambulance, fire engines, etc.

Accident or breakdown detection : This can be a feature so as to locate the accident place and this data can be used by ambulances or other services which would helps victims of the accident.

8. BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

1. N.Kham and C. Nwe “Implementation of modern traffic light control system” ,International journal of scientific and research publications, Vol. 4, Issue 6, Jun. 2014.
2. Isa, N.Shaari ,A. Fayee and N.Azlin ,”Portable wireless traffic light system”, International journal of research in engineering and technology, Vol. 3, Issue 2, pp. 242-247, Feb 2014.
3. K. Vidhya , and A.Banu “Density based traffic signal system”, International journal of innovative research in science, engineering and technology , Vol.3 ,Issue 3,pp. 2218-2223, March 2014.
4. "Adaptive traffic light timing controller," IIT KANPUR NERD MAGAZINE, Siddharth Srivastava, Subhadeep Chakraborty, Raj Kamal, Rahil, and Minocha.
5. Automatic Traffic Density Estimation and Vehicle Classification for Traffic Surveillance Systems Using Neural Networks, mcajournal, Vol. 14, No3, p187.
6. Khekare, G.S.; Sakhare, A.V., "A smart city framework for intelligent traffic system using VANET," Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013 International Multi-Conference on , vol., no., pp.302,305, 22-23 March 2013.
7. Badura, S.; Lieskovsky, A., "Intelligent Traffic System: Cooperation of MANET and Image Processing," Integrated Intelligent Computing (ICIIC), 2010 First International Conference on , vol., no., pp.119,123, 5-7 Aug. 2010.
8. Yang Xiao; Yu-ming Xie; Lingyun Lu; Shuang Gao, "The traffic data compression and decompression for intelligent traffic systems based on two-dimensional wavelet.
9. Redmon, Joseph (2016). "You only look once: Unified, real-time object detection". Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. arXiv:1506.02640. Bibcode:2015arXiv150602640R.
10. O. Chinyere, O. Francisca, and O. Amano , “Design and simulation of an intelligent traffic control system”, International journal of advances in engineering & technology, Vol.1 ,Issue 5,pp. 47-57, Nov 2011.

8.2 WEBSITES

1. https://www.pygame.org/wiki/about is a good place to start.
2. https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006
3. https://www.analyticsvidhya.com/blog/2019/03/opencv-functions-computer-vision-python/
4. https://realpython.com/pygame-a-primer/
5. http://www.mathworks.in/products/image/
6. http://www.tomtom.com/engb/traffic-index/ranking/
7. https://medium.com/@ODSC/overview-of-the-yolo-object-detection-a-review.
8. https://github.com/tzutalin/labelImg is where you can get it.
9. https://www.cityoffirvine.org/signal-operations-maintenance/traffic-signal-synchronization.
10. http://www.mathworks.in/help/images/ref/imsubtract.html
11. https://www.edureka.co/blog/python-opencv-tutorial/
12. https://en.wikipedia.org/wiki/Object_detection
13. https://medium.com/@ODSC/overview-of-the-yolo-object-detection-algorithm-7b52a745d3e0

International Journal of All Research Education and Scientific Methods (IJARESM), ISSN: 2455-6211
Volume 9, Issue 5, May -2021, Impact Factor: 7.429, Available online at: www.ijaresm.com

Advanced Control of Traffic Light System

K Karunakar[1], S Shruthi[2], C Sairam[3], Anshika Gupta[4]

[1]Assistant Professor, CMR Technical Campus,
[2,3,4]CMR Technical Campus

---*****************---

ABSTRACT

Nowadays, traffic congestion is a major issue. Although it appears to be present everywhere, it has the greatest impact on megacities. Its ever-increasing nature necessitates real-time knowledge of road traffic density for better signal control and effective traffic management. Congestion in traffic can be caused by a variety of factors, including insufficient capacity and significant Red Light delays, and so on. The delay of the respective light is hard coded and not dependent on traffic. As a result, there is a need for traffic control simulation and optimization in order to better accommodate.

Keywords: Traffic Congestion, Switching Algorithm, YOLO.

INTRODUCTION

With an increasing population and automobiles in cities, urban traffic congestion is becoming one of the most pressing issues. Not only do traffic delays add to the time and frustration, but they also add to the cost of living. Drivers, but it also raises fuel consumption, transportation expenses, and pollution levels. Megacities are the ones who are most afflicted by it, despite the fact that it appears to be present everywhere. It's also ever-increasing natural resources. As a result, we're implementing a system that aims to reduce the likelihood of such scenarios by automatically computing the optimal green signal time based on current traffic at the signal. And this will ensure that the direction with the most traffic receives a green signal for a longer period of time in comparison to the less congested route. To obtain a more accurate estimate of the green signal time, vehicles are classified as car, bike, bus/truck, or rickshaw. We used object detection techniques such as YOLO to determine the number of vehicles in each direction. The timers of these traffic signals are then set based on vehicle density in each direction, and the system becomes adaptive. This helps to optimise the green signal times, and traffic is cleared at a much faster rate than a static system, reducing unwanted delays, congestion, and waiting time, and thus lowering the overall cost.

Overview

The traffic follows no specific pattern, and the static signal timers worsen the critical condition of congestion. As a result, implementing a system that aims to reduce the likelihood of such scenarios by Based on the current traffic at the signal, the optimal green signal time is calculated automatically. AND will ensure that the direction with the most traffic receives a green signal for a longer period of time than the direction with the least traffic. This system can override the traditional method of hard-coded lights, which causes unnecessary delays, lowering traffic and waiting times, reducing the frequency of accidents and fuel usage, and so helping to limit air pollution.

MATHEMATICAL FORMULATION

$$GST = \frac{\sum_{vehicleClass} (NoOfVehicles_{vehicleClass} * AverageTime_{vehicleClass})}{(NoOfLanes + 1)}.$$

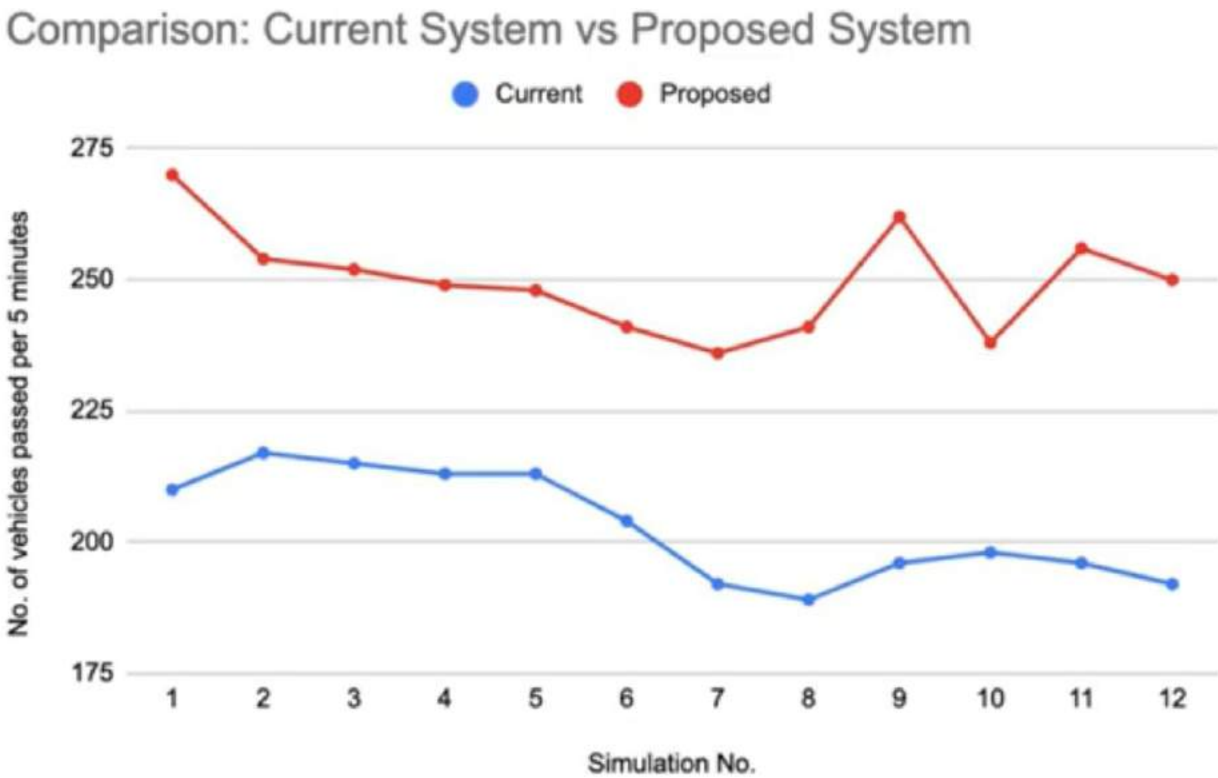
Where;

- GST is green signal time
- noOfVehiclesOfClass is the number of vehicles of each class detected by the vehicle detection module at the signal,
- averageTimeOfClass is the average time it takes for vehicles of that class to cross a junction, and

- noOfLanes is the number of lanes at the intersection.

ANALYSIS

The result can be better visualized in the graph below the blue line is the current static system when the red line is a proposed adaptive system the adaptive system on average lost 48 more vehicles to pass every 5 minutes as compared to the static system.



CONCLUSION

Our recommended method can override the older system of hard-coded lights, which causes serious delays, minimizing traffic and waiting time, reducing the number of accidents and fuel consumption, and so contributing to decrease air pollution. It adjusts the green signal timing based on the traffic density at the signal, ensuring that the direction with one of the most traffic is given a green signal for a prolonged period than the one with the least traffic. As a result, the suggested adaptive system outperforms the current approach by 23% in terms of the number of vehicles crossing an intersection. This is a huge advancement, and this adaptable technology may be combined with CCTV cameras in major cities to help with traffic control.

REFERENCES

[1] Tom Tom World Traffic Index [online], 2019. http://www.tomtom.com/en gb/traffic-index/ranking/
[2] . N.Kham and C. Nwe “Implementation of modern traffic light control system” ,International journal of scientific and research publications, Vol. 4, Issue 6, Jun. 2014.
[3] Isa, N.Shaari ,A. Fayee and N.Azlin ,”Portable wireless traffic light system”, International journal of research in engineering and technology, Vol. 3, Issue 2, pp. 242-247, Feb 2014.
[4] "Adaptive traffic light timing controller," IIT KANPUR NERD MAGAZINE, Siddharth Srivastava, Subhadeep Chakraborty, Raj Kamal, Rahil, and Minocha.
[5] YOLO Object Detection Algorithm Overview, 2018, [online] https://medium.com/@ODSC/overview-of-the-yolo-object-detection-a-review.
[6] LabelImg Annotation Tool, [online], 2015. https://github.com/tzutalin/labelImg is where you can get it.
[7] Pygame Library, [online], 2019. https://www.pygame.org/wiki/about is a good place to start.

[8] Traffic Signal Synchronization Traffic signal synchronisation can be found at https://www.cityoffirvine.org/signal-operations-maintenance/traffic-signal-synchronization.
[9] https://www.analyticsvidhya.com/blog/2019/03/opencv-functions-computer-visionpython/
[10] https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detectionexplained-492dc9230006

IJARESM

ISSN: 2455-6211, New Delhi, India

International Journal of All Research Education & Scientific Methods

An ISO & UGC Certified Peer-Reviewed Multi-disciplinary Journal

Certificate of Publication

K Karunakar

Assistant Professor, CMR Technical Campus

TITLE OF PAPER

Advanced Control of Traffic Light System

has been published in

IJARESM, Impact Factor: 7.429, Volume 9 Issue 5, May- 2021

Paper Id: IJARESM/May21

Date: 31-05-2021

CERTIFIED ISO 9001:2008 JOURNAL

Website: www.ijaresm.com
Email: editor.ijaresm@gmail.com

IJARESM Auth. Signatory INDIA

Authorized Signatory

IJARESM

ISSN: 2455-6211, New Delhi, India

International Journal of All Research Education & Scientific Methods

An ISO & UGC Certified Peer-Reviewed Multi-disciplinary Journal

Certificate of Publication

S Shruthi

CMR Technical Campus

TITLE OF PAPER

Advanced Control of Traffic Light System

has been published in

IJARESM, Impact Factor: 7.429, Volume 9 Issue 5, May- 2021

Paper Id: IJARESM/May21

Date: 31-05-2021

Website: www.ijaresm.com
Email: editor.ijaresm@gmail.com

Authorized Signatory

IJARESM

ISSN: 2455-6211, New Delhi, India

International Journal of All Research Education & Scientific Methods

An ISO & UGC Certified Peer-Reviewed Multi-disciplinary Journal

Certificate of Publication

C Sairam

CMR Technical Campus

TITLE OF PAPER

Advanced Control of Traffic Light System

has been published in

IJARESM, Impact Factor: 7.429, Volume 9 Issue 5, May- 2021

Paper Id: IJARESM/May21

Date: 31-05-2021

Website: www.ijaresm.com
Email: editor.ijaresm@gmail.com

Authorized Signatory

IJARESM

ISSN: 2455-6211, New Delhi, India

International Journal of All Research Education & Scientific Methods

An ISO & UGC Certified Peer-Reviewed Multi-disciplinary Journal

Certificate of Publication

Anshika Gupta

CMR Technical Campus

TITLE OF PAPER

Advanced Control of Traffic Light System

has been published in

IJARESM, Impact Factor: 7.429, Volume 9 Issue 5, May- 2021

Paper Id: IJARESM/May21

Date: 31-05-2021

Website: www.ijaresm.com
Email: editor.ijaresm@gmail.com

Authorized Signatory