Mini Project On

Dynamic Traffic Signal
Using IMAGE processing and DEEP learning

BACHELOR OF TECHNOLOGY
in
ELECTRONICS & TELECOMMUNICATION ENGINEERING

Department of Electronics & Telecommunication Engineering
Sardar Patel Institute of Technology
Munshi Nagar, Andheri(W), Mumbai-400058
UNIVERSITY OF MUMBAI 2021-2022

By
Aditya Kulkarni : 2020200037
Sandeep Pillai : 2020200049
Rohan Rane : 2020200051

Under the guidance of
Dr. Reena Sonkusare

CERTIFICATE

This is to certify that the dissertation entitled **"Dynamic Traffic Signal Using IMAGE processing and DEEP learning"** has been completed successfully by **Aditya Kulkarni, Sandeep Pillai and Rohan Rane** under the guidance of **Dr. Reena Kumbhare** for the award of Degree of **Bachelor of Technology** in **Electronics & Telecommunication Engineering** from **University of Mumbai**.

Certified by

Dr. Reena Sonkusare
Project Guide and Head of Department

Dr. B. N. Chaudhari
Principal

Department of Electronics & Telecommunication Engineering
Sardar Patel Institute of Technology
Munshi Nagar, Andheri(W), Mumbai-400058
UNIVERSITY OF MUMBAI 2021-2022

DISSERTATION APPROVAL CERTIFICATE

This is to certify that the dissertation entitled **"Dynamic Traffic Signal Using IMAGE processing and DEEP learning"** by **Aditya Kulkarni, Sandeep Pillai and Rohan Rane** is approved for the award of Degree of **Bachelor of Technology** in **Electronics & Telecommunication Engineering** from **University of Mumbai**.

External Examiner **Internal Examiner**

(signature) **(signature)**

Name: **Name:**

Date: **Date:**

Seal of the Institute

Abstract

Traffic lights are the source of signaling devices for road junctions. Normal traffic lights have limitations as they use predefined hardware, which function according to the static hard coded program which does not have flexibility. As the number of vehicles increases, current systems fail. To manage the traffic flow we have come up with "Dynamic Traffic Signal". The most significant issue which is being looked at by the advanced world is the traffic blockage in the urban communities and towns. The system contains raspberry-pi and camera modules. A camera will be placed along with the traffic light. It will capture the live video and extract a frame from it. After that, the deep learning algorithm will count the number of vehicles. Based on different vehicle counts, the raspberry pi takes decisions and as a result it updates the traffic light durations.

Acknowledgement

We have great pleasure in presenting the report on **"Dynamic Traffic Signal Using IMAGE processing and DEEP learning"**. We take this opportunity to express our sincere thanks towards our mentor **Dr. Reena Sonkusare, Professor & Head of Department of Electronics & Telecommunication Engineering, S.P.I.T., Mumbai,** for providing the technical guidelines and the suggestions regarding this line of work. We would like to express our gratitude towards their constant encouragement, support and guidance throughout the development of the project.

We also thank all the staff of S.P.I.T., Mumbai for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all of my peers at S.P.I.T., Mumbai for their encouragement.

Aditya Kulkarni
Sandeep Pillai
Rohan Rane

Contents

Chapter 1

Introduction

With the modern world continuously becoming very fast-paced each and every person is always trying to make the most of his time. It is very much required that any person doesn't waste a lot of his crucial time on a petty activity like traveling. Along with this driving in streets with a lot of traffic has been scientifically proven to be the cause of very high mental strain and pressure. So it is a basic requirement in modern-day cities to have a dynamic model of the traffic signal to control the transportation in the area.

Coming to the present case of traffic lights across the cities, it can be noted that it is static and is always the same for any lane, even though the traffic in those lanes may not be the same. This causes some lanes to become empty while some lanes are too congested and this is a waste of time for people and also a waste of resources as a lot of fuel is wasted while not moving. Along with this in congested lanes the traffic light is for a very less duration, so many people tend to cut the traffic light and this may be the cause of various road accidents.

Coming to the most important point of why we require dynamic traffic lights is that there are many emergencies occurring and it is required for emergency vehicles like police, ambulance, and fire trucks to reach these situations in the proper time as this is the case of life or death.

For all these reasons we can say that dynamic traffic control is the best solution. In a dynamic traffic light the waiting time of a signal changes with respect to the number of cars in the lane and thus nearly all the lanes become equally congested or equally empty. Thus the average waiting time of all the people at the traffic light is reduced.

Chapter 2

Literature Review

The main objective of this Literature Review was to study and analyze a design and a system which would be most efficient and affordable with an ability to modify it whenever required to cope up with technology modernization.

2.1 Dynamic Traffic System Based On Real Time Detection Of Traffic Congestion

By Aditya Rao, Akshay Phadnis, Atul Patil, Tejal Rajput and Dr. Prof. Pravin Futane.

On 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)

- **Overview:** The paper proposes a dynamic traffic system that takes in present traffic footage and calculates the percentage congestion and based on this, allocates the timer to each signal.

- **Positive Aspects:** Uses image processing for Background subtraction and Edge Detection

- **Negative Aspects:** Uses Canny Edge Detection in which Weak and strong edges are combined which may cause errors.

2.2 Smart Traffic Light Switching/Traffic Density Calculation using Video Processing

By Anurag Kanungo, Ayush Sharma and Chetan Singla
On 2014 RAECS UIET Panjab University Chandigarh

- **Overview:** This paper presents a method to use live video feed from the cameras at traffic junctions for real time traffic density calculation using video and image processing.

- **Positive Aspect:** Used for 4 way junction detection which shows 35% improvement in congestion and allows Traffic light synchronization enabling free flow of traffic

- **Negative Aspect**: Shows poor results in low light conditions

2.3 Density and Time based Traffic Control System using Video Processing

By Tanvi Sable, Nehal Parate, Dharini Nadkar, Swapnil Shinde
On ITM Web of Conferences 32 (ICACC - 2020)

- **Overview:** This paper discusses the idea of a traffic signal system by detecting traffic density and adjusting the signal accordingly.

- **Positive Aspect:** Uses haar cascade algorithm providing high accuracy

- **Negative Aspect:** Uses single rotor camera which has high operation / maintenance cost

Chapter 3

3.1 Problem Statement

With the traffic increasing day by day in each and every city the static system of the present day is becoming time consuming and inconvenient. It comes to our notice when we see some lanes being completely full and still suffering in red light stoppage for a long time while some lanes are so empty that providing them a long green light time does not make any sense.

3.2 Objective

- To make the present static system of traffic lights more dynamic so that the waiting time on average for most of the vehicles is reduced by maximum.

- To prioritize the time aspect of people by counting the number of vehicles in each lane to vary the timing of the traffic lights with respect to the amount of vehicles.

3.3 Gap Analysis

Existing System	New System
Fixed Algorithm	Flexible algorithm
No traffic, but still need to wait	Set signal time according to traffic density
No exceptions for emergency cases	Emergency cases

Chapter 4

Theory

Port structure of Raspberry Pi 3 Model B

- **Status LED:**

  Current RPi models have the following LEDs:

  1) **PWR (red):** Indicates that power has been provided to the board. On A+ and later models it will flash if the voltage drops below 4.63V

  2) **LNK (green):** Indicates Ethernet (LAN) connection and activity: it is constantly ON when connected, and flashes on data transfers. Located on the PCB in model B, and on the Ethernet RJ-45 socket in later models.

  3) **100/1000 (yellow or green):** Indicates 100Mbit Ethernet link on 100Mbit models or 1000 Mbit link on Raspberry Pi 3B+ and 4. It's OFF for lower speed connections. Located on the PCB in model B, and on the Ethernet RJ-45 socket in later models.

  4) **FDX (green or orange):** Indicates Full Duplex Ethernet connection. Only exists in model B.

- **USB 2.0 Port:** There are 4 USB 2.0 ports in Raspberry Pi 3 model B which can be used to connect various input devices like keyboard, camera, mouse, etc for getting input to the raspberry pi.

- **10/100 Ethernet Port:** This can be used to provide internet connection to the raspberry pi using the ethernet cable.

- **Micro USB connector:** This port is used to connect the raspberry pi to a current source generally of above 1.2A for basic functioning of the raspberry pi.

- **HDMI Video/Audio Connector:** This can be used to connect a screen to view the processing of raspberry pi on a screen specifically monitor of a computer or a television with hdmi port.

- **CSI Camera Connector:** This is used to connect the raspberry pi legacy camera module to the raspberry pi to get video input from the camera.

- **Micro SD Card Slot (Under the board):** The whole raspberry pi requires a storage so that operating system can be installed in it, the codes written can be saved in it and finally various processing outputs can be processed in it. It is advisable to use an SD card of more than 8GB so that all functions can be carried out in the proper manner.

- **GPIO Header:** These are the 40 general purpose input output ports which can be used to manipulate various processes like lighting a led, etc.

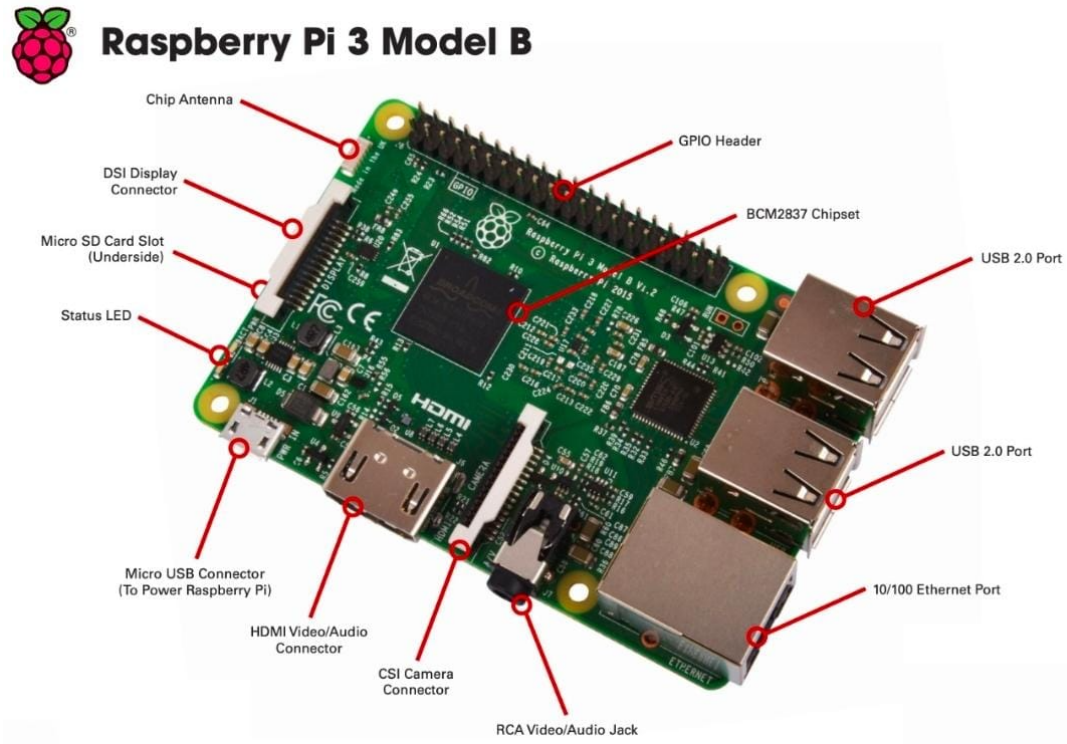


Fig 1: Ports Of Raspberry Pi 3B

Pin structure of Raspberry Pi 3 Model B

1. **GPIO:** These can be used to connect any device to the raspberry pi and as these are basically the general purpose input output pins, we have used these to connect the led traffic light module to our raspberry pi and make the pins work as output pins.

2. **GND:** Ground is provided so that the current is so that if excess current comes from the connected devices is properly handled and raspberry pi or any other device connected is not damaged.

3. **3.3V PWR:** This can be used to provide power to raspberry pi if the usb and micro usb connector are used by raspberry pi to connect some other devices and these ports are not available for providing the power.

4. **Reserved:** These pins are reserved for some other purposes by the raspberry pi.

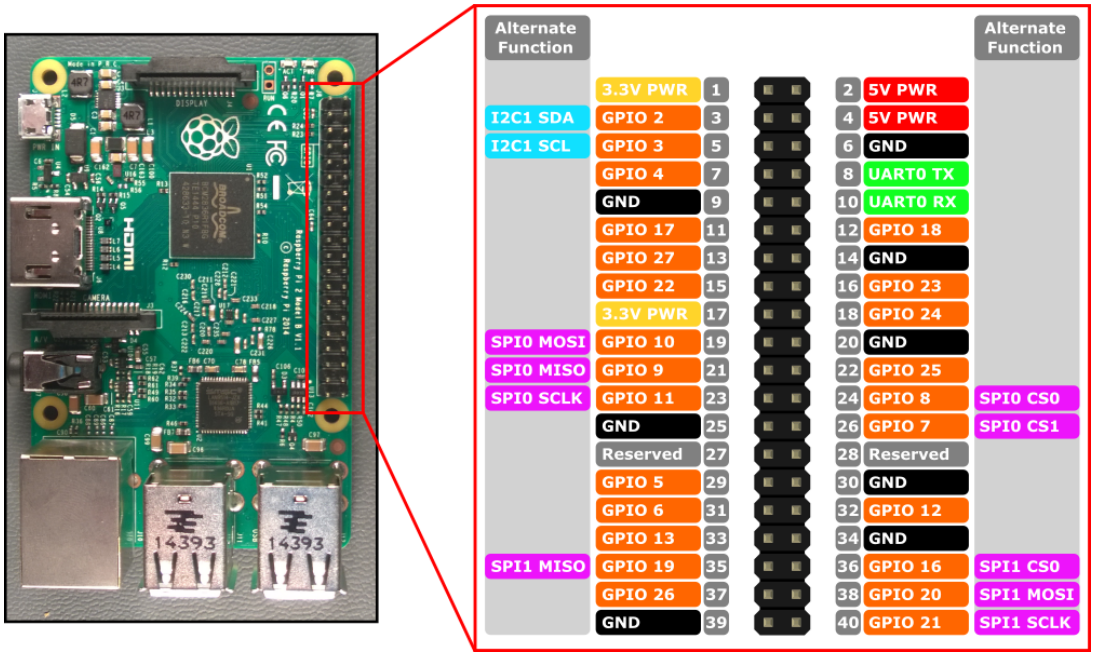


Fig2: Pin Diagram of GPIO Pins in Raspberry Pi 3B

Models used for vehicle detection:

Yolo

- YOLO is an abbreviation for the term ‘You Only Look Once’. This is an algorithm that detects and recognizes various objects in a picture (in real-time). Object detection in YOLO is done as a regression problem and provides the class probabilities of the detected images.

- The YOLO algorithm employs convolutional neural networks (CNN) to detect objects in real-time. As the name suggests, the algorithm requires only a single forward propagation through a neural network to detect objects.

- This means that prediction in the entire image is done in a single algorithm run. The CNN is used to predict various class probabilities and bounding boxes simultaneously.

How the YOLO algorithm works

YOLO algorithm works using the following three techniques:

- Residual blocks
- Bounding box regression
- Intersection Over Union (IOU)

Residual blocks

First, the image is divided into various grids. Each grid has a dimension of S x S. The following image shows how an input image is divided into grids.

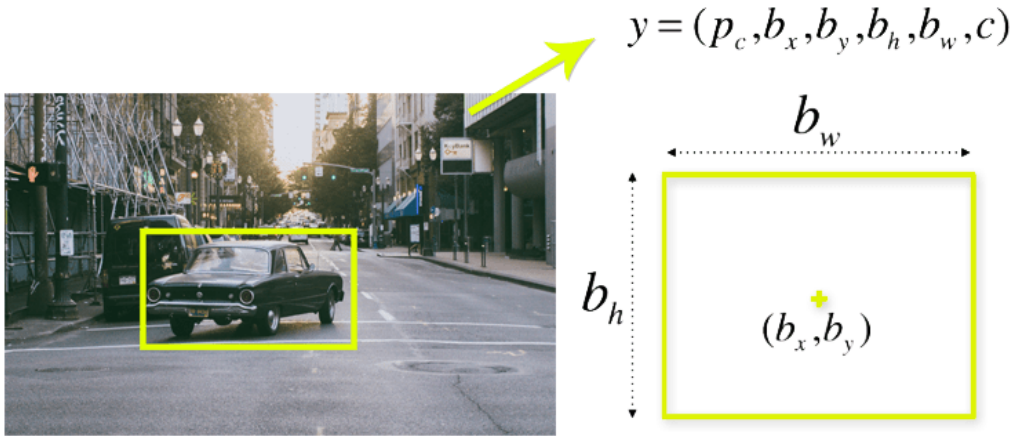
In the image above, there are many grid cells of equal dimension. Every grid cell will detect objects that appear within them. For example, if an object center appears within a certain grid cell, then this cell will be responsible for detecting it.

Bounding box regression

A bounding box is an outline that highlights an object in an image.
Every bounding box in the image consists of the following attributes:

- Width (bw)
- Height (bh)
- Class (for example, person, car, traffic light, etc.)- This is represented by the letter c.
- Bounding box center (bx,by)

The following image shows an example of a bounding box. The bounding box has been represented by a yellow outline.

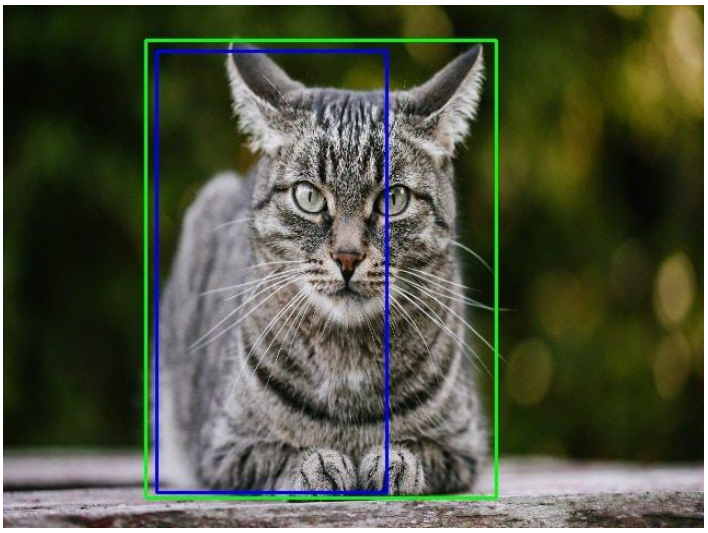


YOLO uses a single bounding box regression to predict the height, width, center, and class of objects. In the image above, represents the probability of an object appearing in the bounding box.

Intersection over union (IOU)

Intersection over union (IOU) is a phenomenon in object detection that describes how boxes overlap. YOLO uses IOU to provide an output box that surrounds the objects perfectly.

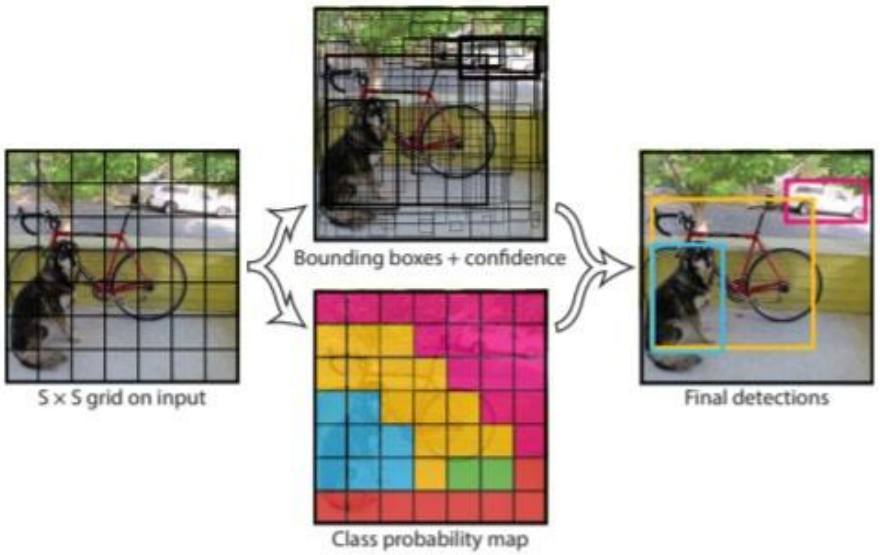
Each grid cell is responsible for predicting the bounding boxes and their confidence scores. The IOU is equal to 1 if the predicted bounding box is the same as the real box. This mechanism eliminates bounding boxes that are not equal to the real box. The following image provides a simple example of how IOU works.



In the image above, there are two bounding boxes, one in green and the other one in blue. The blue box is the predicted box while the green box is the real box. YOLO ensures that the two bounding boxes are equal.

Combination of the three techniques

The following image shows how the three techniques are applied to produce the final detection results.



- First, the image is divided into grid cells. Each grid cell forecasts B bounding boxes and provides their confidence scores. The cells predict the class probabilities to establish the class of each object.

- For example, we can notice at least three classes of objects: a car, a dog, and a bicycle. All the predictions are made simultaneously using a single convolutional neural network.

- Intersection over union ensures that the predicted bounding boxes are equal to the real boxes of the objects. This phenomenon eliminates unnecessary bounding boxes that do not meet the characteristics of the objects (like height and width). The final detection will consist of unique bounding boxes that fit the objects perfectly.

- For example, the car is surrounded by the pink bounding box while the bicycle is surrounded by the yellow bounding box. The dog has been highlighted using the blue bounding box.

Haar Cascade

Haar Cascade classifiers are an effective way for object detection. This method was proposed by Paul Viola and Michael Jones in their paper Rapid Object Detection using a Boosted Cascade of Simple Features .Haar Cascade is a machine learning-based approach where a lot of positive and negative images are used to train the classifier.

- **Positive images –** These images contain the images which we want our classifier to identify.
- **Negative Images –** Images of everything else, which do not contain the object we want to detect.

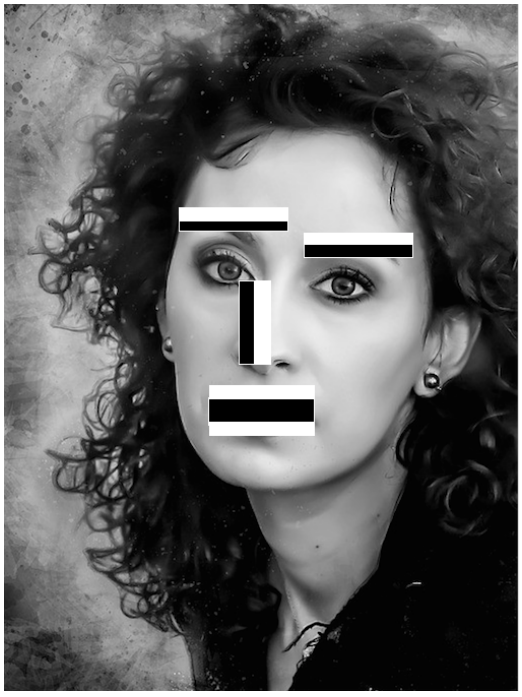
How the Haar Cascade Algorithm works

Haar Cascade algorithm works on a 4 step process:

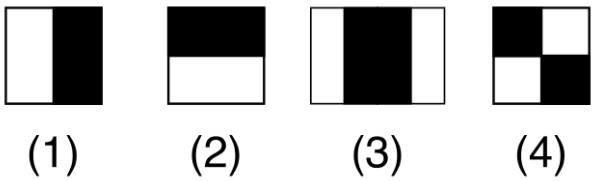
- Haar features
- Integral images
- Adaboost classifier
- Cascading

Haar Features

Haar-like features are digital image features used in object recognition. A Haar-like feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. This difference is then used to categorize subsections of an image.



Haar features applied on relevant parts of face

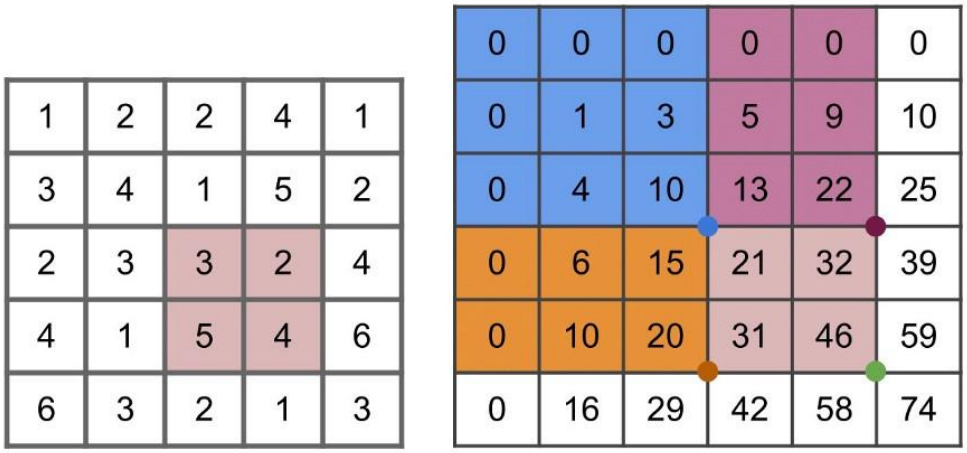


Square shape Kernels : Haar Cascade Features

We will use respective Haar features for detecting eyebrows, forehead, eyebrow from lighted pixels- darker pixels. For black and white images (ideal cases) pixel values are 0 or 1 but in real cases there are normalized pixel values.

Integral Images

An Integral Image is an intermediate representation of an image where the value for location (x, y) on the integral image equals the sum of the pixels above and to the left (inclusive) of the (x, y) location on the original image



1	2	2	4	1
3	4	1	5	2
2	3	3	2	4
4	1	5	4	6
6	3	2	1	3

Input Image

0	0	0	0	0	0
0	1	3	5	9	10
0	4	10	13	22	25
0	6	15	21	32	39
0	10	20	31	46	59
0	16	29	42	58	74

Integral Image

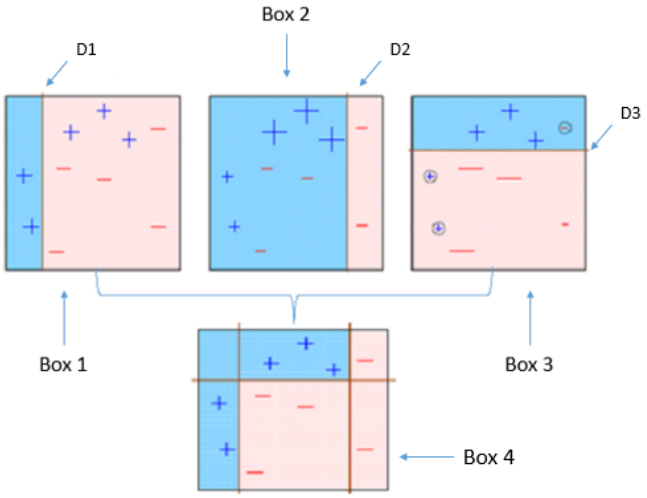
Integral Images is a data structure and algorithm feature used for generating the sum of values in a rectangular subset of a grid. The goal is reducing the number of computations needed to obtain the summations of pixel intensities within a window.

$$I(x,y) = \sum_{\substack{x' \le x \\ y' \le y}} i(x',y')$$

The computed value by the Haar features are applied to the the mathematical tool provided by the integral image algorithm to fasten this complex process of computing

Adaboost Classifiers

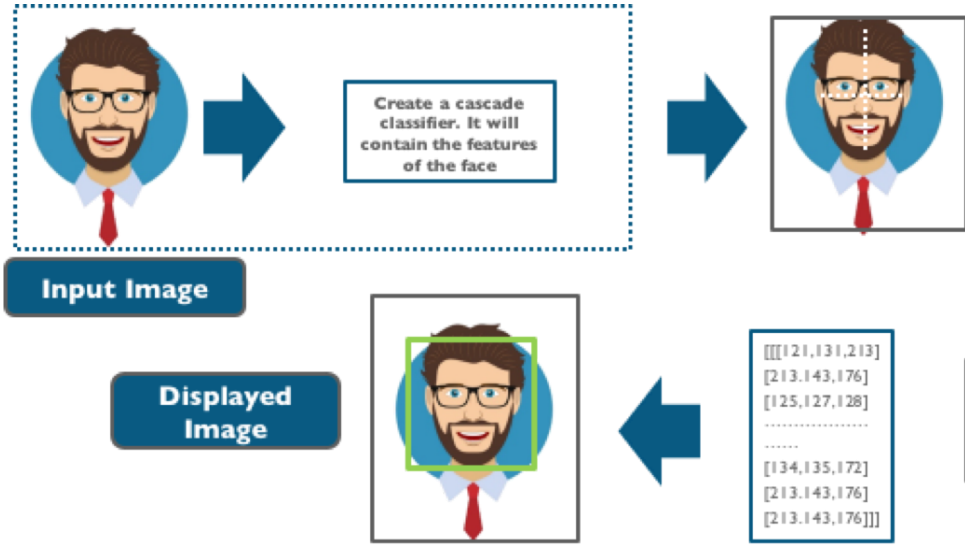
An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.



Adaboost Classifier

Cascading

Cascading is a particular case of ensemble learning based on the concatenation of several classifiers, using all information collected from the output from a given classifier as additional information for the next classifier in the cascade.



Libraries used:

OpenCV (cv2) : OpenCV (Open Source Computer Vision Library) is an open-source library that includes several hundreds of computer vision algorithms.

Turtle: Turtle is a Python library which is used to create graphics, pictures, and games. Turtle is similar to the virtual canvas in which we can draw pictures and attractive shapes. It provides an onscreen pen that we can use for drawing.

Random: Python Random module is an in-built module of Python which is used to generate random numbers

Glob: The glob module finds all the pathnames matching a specified pattern according to the rules used by the Unix shell, although results are returned in arbitrary order.

Time: Python's time module provides a function for getting local time from the number of seconds elapsed since the epoch called localtime()

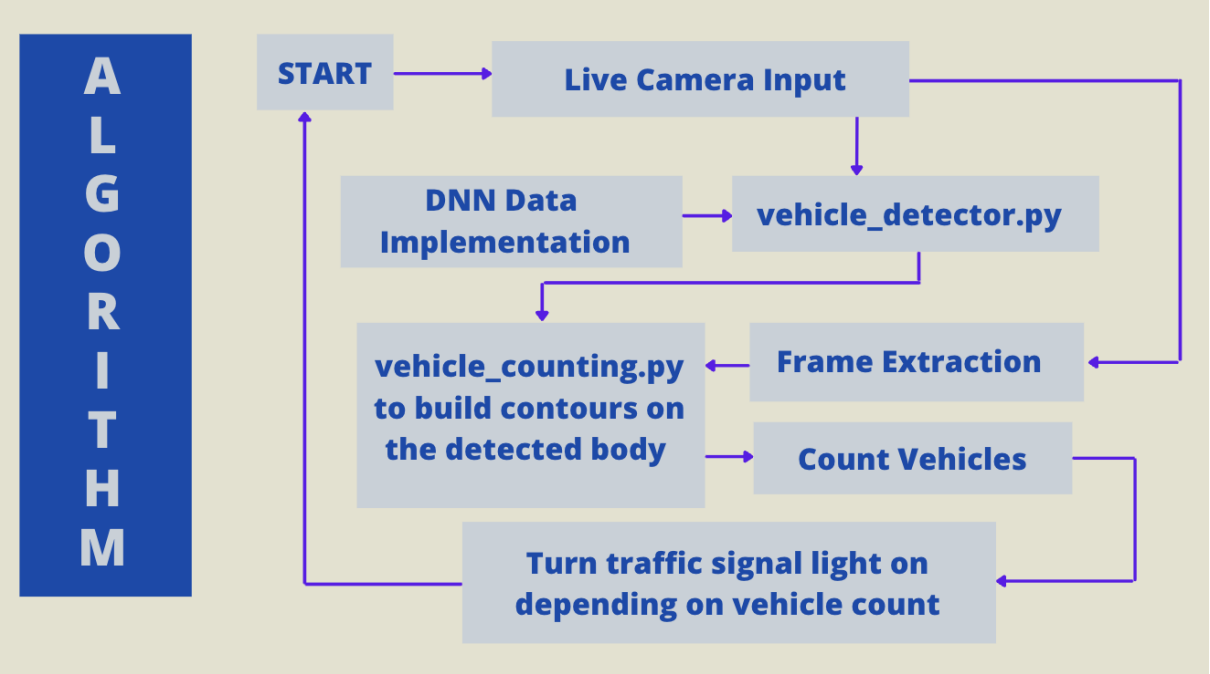
GPIO zero: A simple interface to GPIO devices with Raspberry Pi

Chapter 5

System Design

- **Software Design**

  Yolo Algorithm



The Yolo Algorithm was used in implementing in the software design

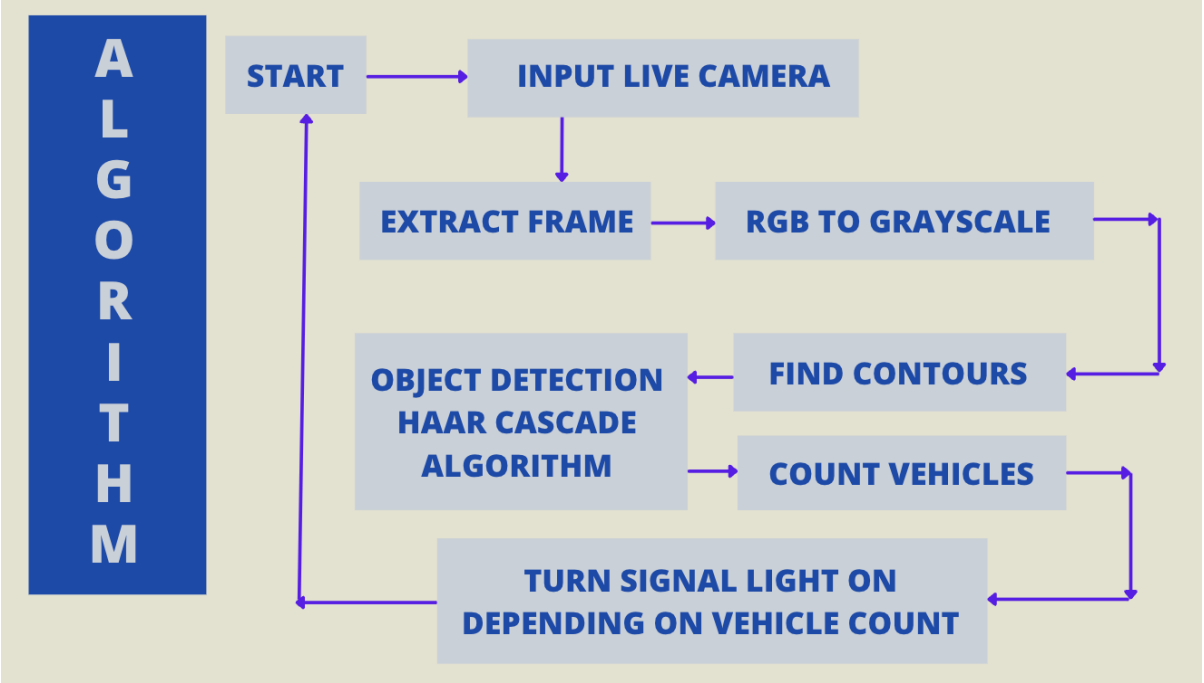
The setup uses the DNN (Deep Neural Network) model which is a trained model having data on vehicle detection.

The code traffic1_final.py uses the trained model to find the number of vehicles. The vehicles above a certain probability (50%) are counted.

The code is given an input of four random images, showing vehicles at a traffic stop from the images folder, herein the vehicle_detector.py file is called by the current running code. The vehicle detection and counting operation executes and 4 variables store the count on the number of vehicles from each of the 4 input images respectively.

The variables passed are given as input to the turtle library codes to display red, yellow and green light operation of a four way traffic signal in synchronization with each other

Haar Cascade Algorithm



The Haar Cascade Algorithm was used in implementing in the Hardware design

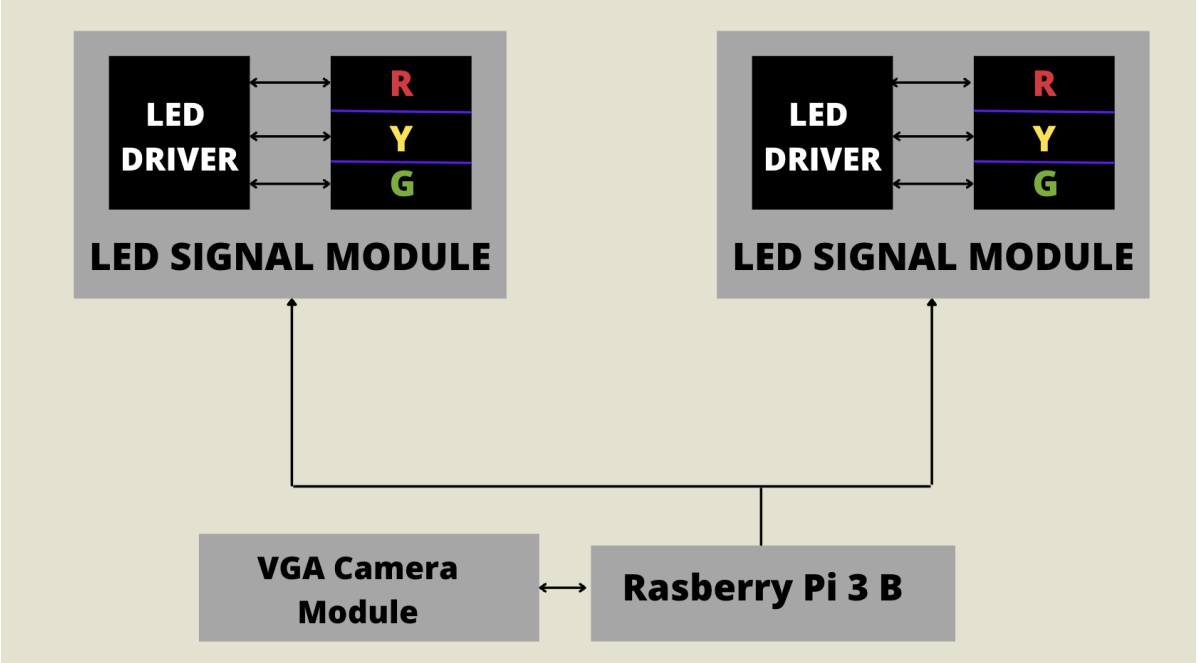
The setup uses the cascade classifier (XML file) model which is a trained model having data on vehicle detection.

The code traffic2.py uses this trained model to find the number of vehicles.

The code is given an input of four random images, showing vehicles at a traffic stop one from a camera input and other three from the images folder, herein the XML file is executed by the current running code. The vehicle detection and counting operation executes and 4 variables store the count on the number of vehicles from each of the 4 input images respectively.

The variables passed are given as input to the GPIO pins to display red, yellow and green light operation of a four way traffic signal in synchronization with each other using the signal modules

- **Hardware Design**



The above model was implemented practically

The Haar Cascade Algorithm was preferred over the Yolo Algorithm to be implemented in raspberry pi 3B because it consumes lower RAM.

The Yolo algorithm consumed over 1GB RAM for highly accurate results. The version of raspberry pi used for the current design itself, only has 1GB RAM. Hence the code terminates abruptly. Therefore the cascade algorithm is implemented for the above design, which consumes just about 200MB with a very poor detection accuracy.

Chapter 6

Software

Raspbian OS - 32 bit Bulls Eye

Python IDE

Python Compiler

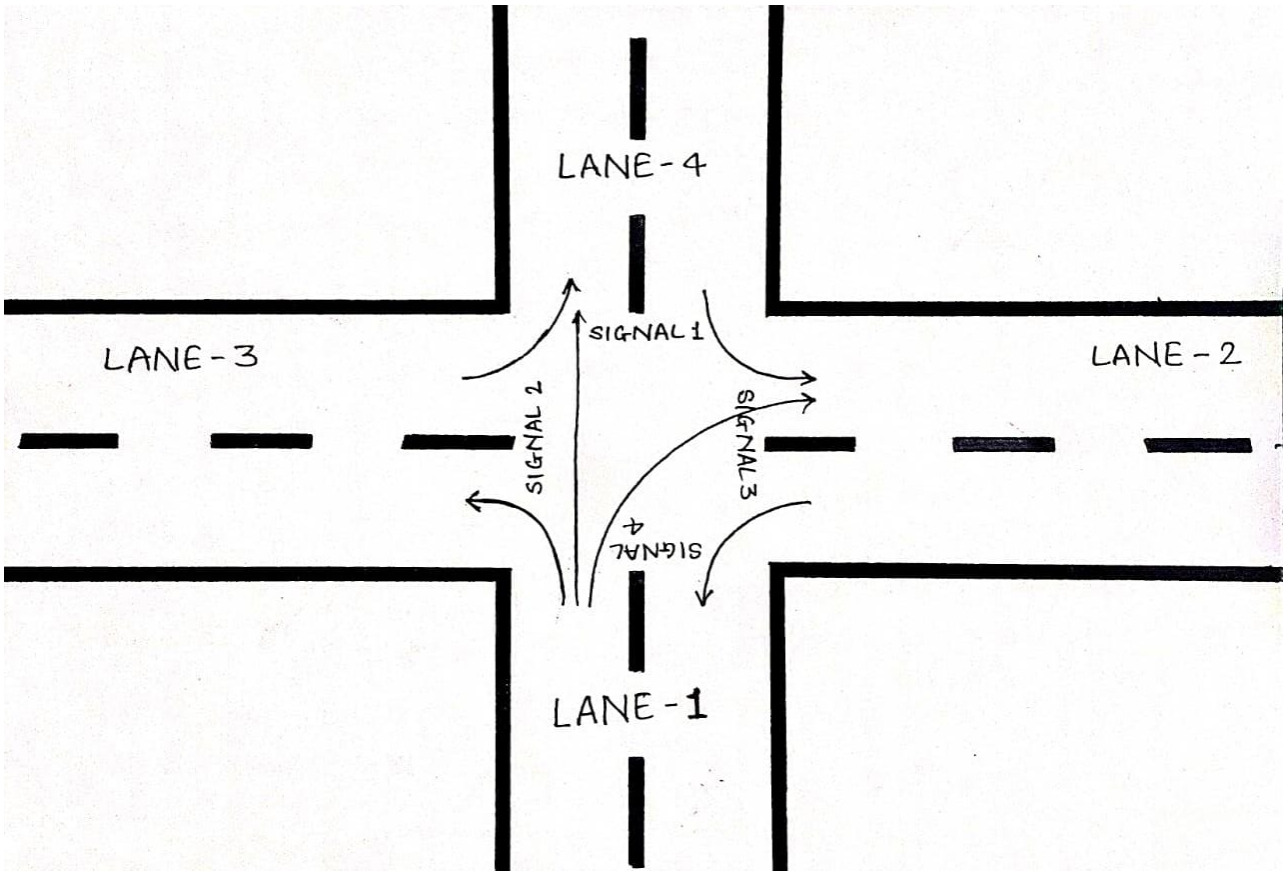
Open CV

Turtle - Graphic Tool

Chapter 7

Simulation & Experimental Results

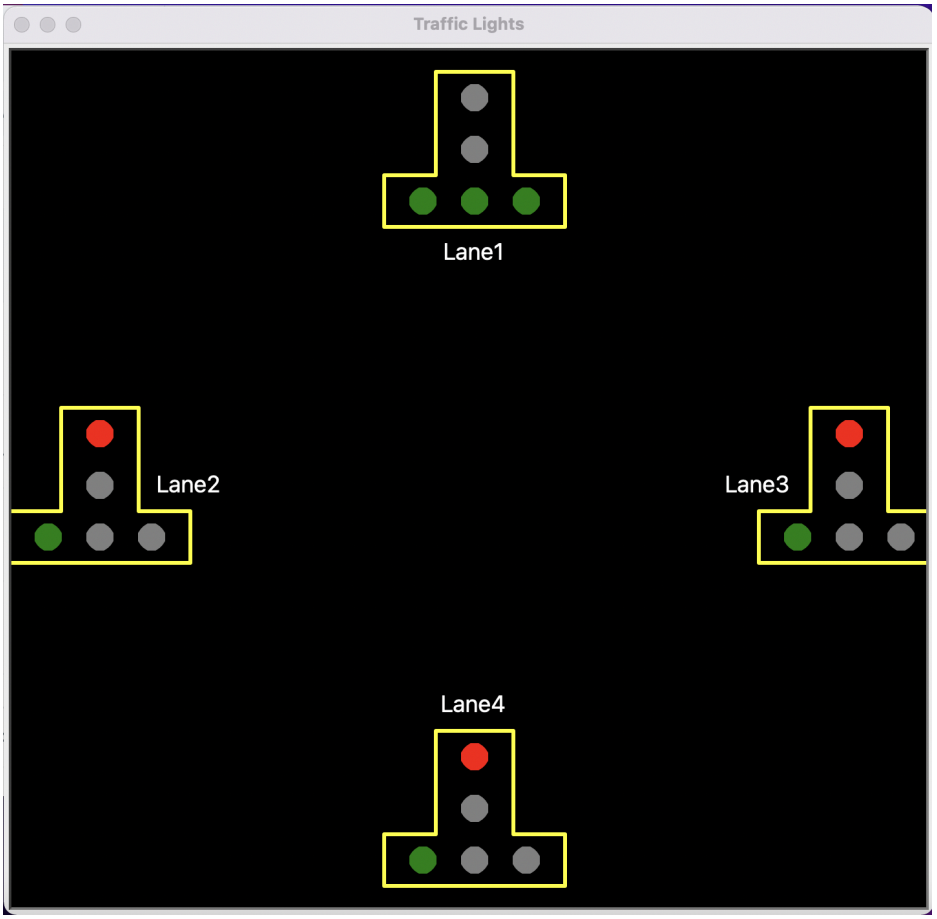
Manual Diagram



A four way Traffic Light Junction along with flow of traffic

The above Diagram shows the flow of traffic from each lane for the duration in which, the Signal 1 has all its green light enabled (left, right, ahead), and Signal 2, Signal 3, Signal 4 all has its left (green) light enabled.

Simulation in Pycharm

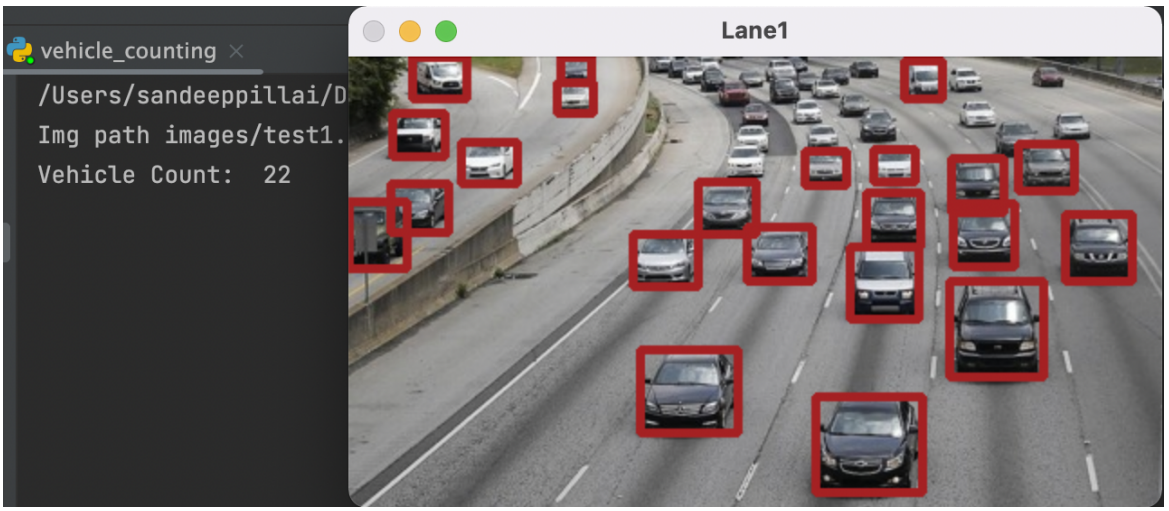


A four way Traffic Light Junction depicting the operation of signals

The above Manual Diagram flow is depicted by showing the operation of traffic signals through a simulation for our Dynamic Traffic Control System.

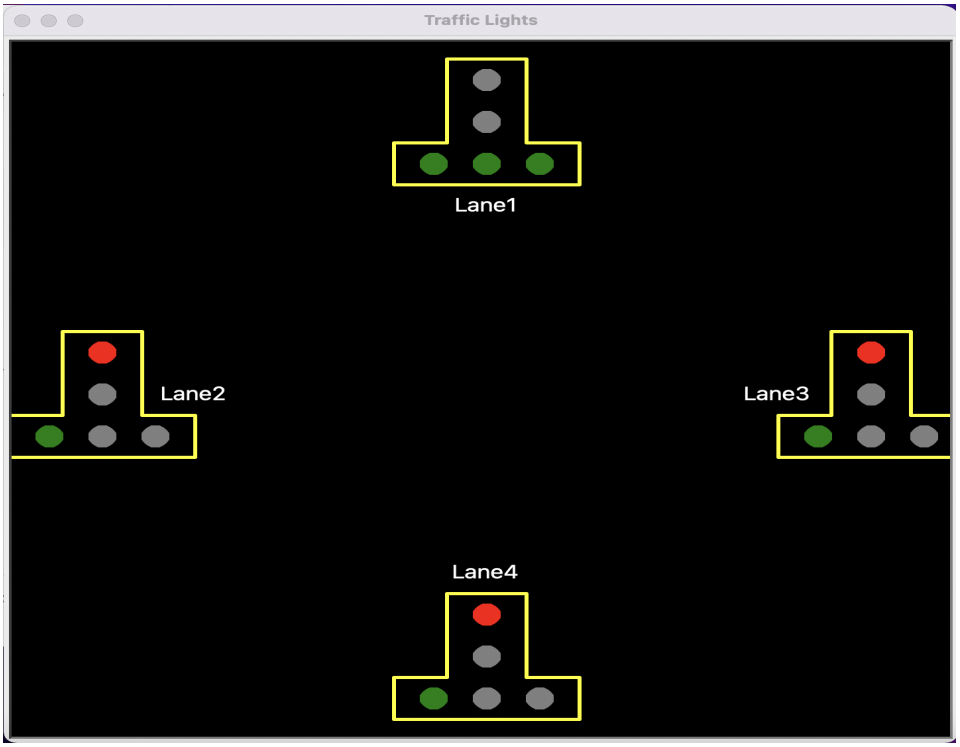
Vehicle Detection and Counting - Software Implementation

Results on execution of the code



22 cars detected at Lane 1 using Yolo Algorithm

Graphical Signal Output

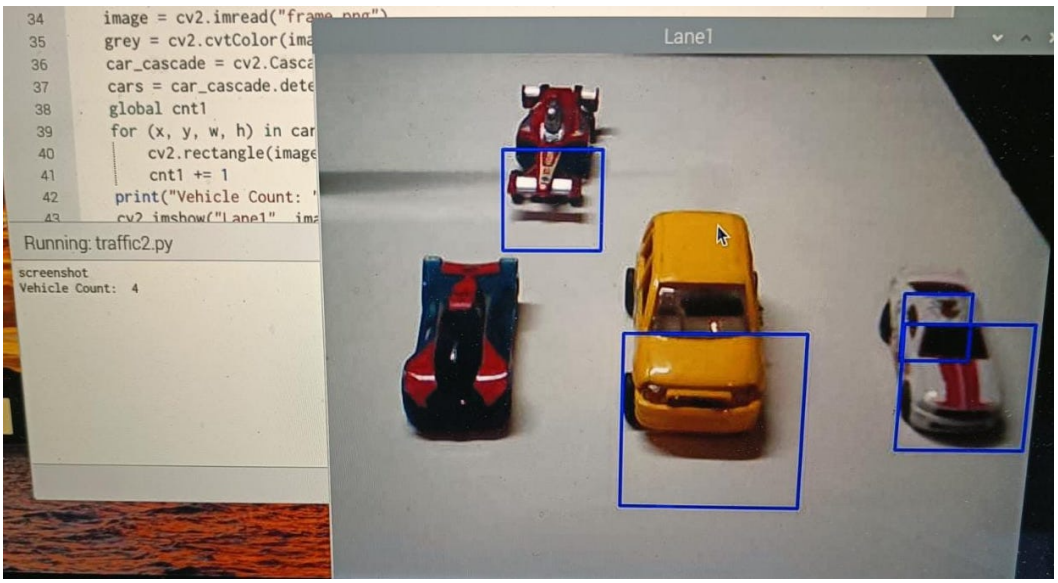


The signal for Lane 1 shows green for 22 seconds.

Above same execution is carried out for Lane 2, 3 and 4 as well. The vehicles at each Lane are detected continuously in a while loop and vehicle count results are obtained, accordingly the delay for the green light is set.

Vehicle Detection and Counting - Hardware Implementation

Results on execution of the code



4 cars detected at Lane 1 using Haar Cascade Algorithm

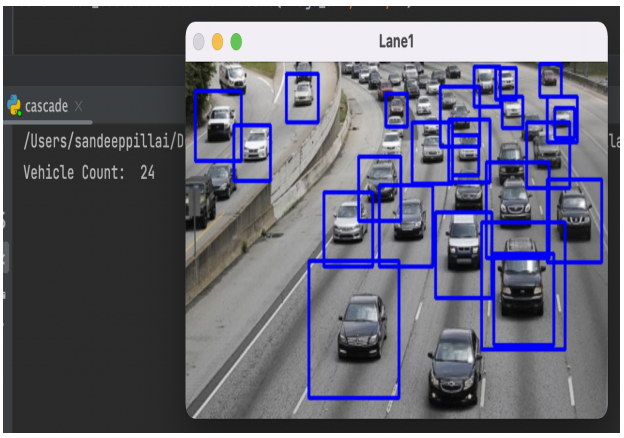
Signal Module Output

The signal for Lane 1 shows green for 4 seconds

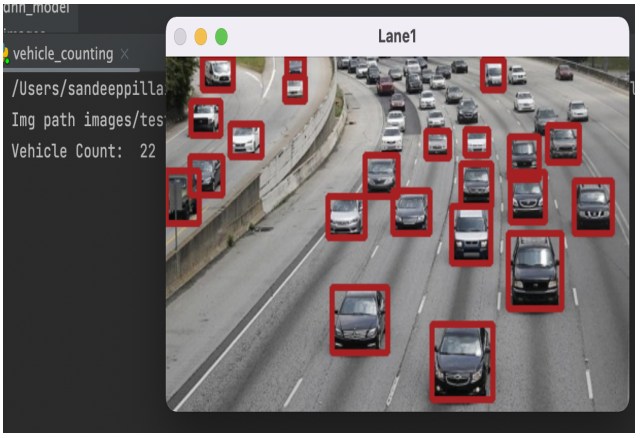
Above same execution is carried out for Lane 2, 3 and 4 as well. The vehicles at each Lane are detected continuously in a while loop and vehicle count results are obtained, accordingly the delay for the green light is set.

Haar Cascade Algorithm V.S. YOLO Algorithm

Haar Cascade Algorithm	Yolo Algorithm
Provides results which are not very accurate	Provides very accurate results
It is an application of Machine Learning.	It is an application of Deep Neural Networks.
It is relatively easier to train	It is very complex to train
Works on philosophy of Adaboost in which the positive and negative images are compared to relatively detect the object.	Works on the philosophy of Bounded box Regression in which an image is divided into small grids and then an object box is created and then bounded boxes are checked and made equal.
Requires less RAM for working i.e. about 200MB, so can work perfectly in a raspberry pi 3B which has an RAM of 1GB.	Requires very high RAM for working i.e. about more than 1GB, so it doesn't work in a raspberry pi 3B model. As this version has only 1GB of RAM.
It is used in areas where speed is the key to detect the object.	It is used in areas where accuracy is the key to detect the object.



Haar Cascade - Machine Learning



Yolo - Deep Learning

Here we can see the Yolo Algorithm is much more accurate than the Haar Cascade Algorithm as it is based on the Deep Learning model which has superior accuracy than the Machine Learning Model.

Process Name	Mem...	Threads	Ports	PID	User
Python	1.08 GB	19	353	3254	sandeeppillai

Yolo Algorithm is a high RAM consuming algorithm. Hence an efficient RAM consuming device is required.

Haar Cascade Algorithm consumes less RAM. Hence any device with a suitable amount of RAM such as Raspberry pi 3B (1GB RAM) can be used to execute such an algorithm.

Conclusion & Future Scope

Conclusion

- Various methods of managing traffic are present but traffic light monitoring is the most important method so that loads of time is saved by the people traveling.

- Our dynamic traffic signal enables road accidents to be reduced by a large amount. It provides an efficient algorithm to save time spent traveling

- We concluded that if limited RAM is present then Haar Cascade is a great way to get things going, which we have done in our hardware part as the Raspberry Pi 3B has only 1GB RAM which can run this Algorithm properly but with very poor detection accuracy.

- We also concluded that the Yolo Algorithm requires a high amount of RAM (more than 1GB) for highly accurate results. The version of raspberry pi used for the current design itself, only has 1GB RAM. Hence the code terminates abruptly.

- Finally concluded that our solution is the best alternative possible in the market as it provides a fast and accurate way to reduce time consumed in traffic.

Future Scope

- Prioritizing the traffic light if the presence of emergency vehicles like ambulances, police vans or fire trucks is detected in the particular lane so that these emergencies can be handled in a better way.

- Allowing face detection in this system so that green light for pedestrians can also be brought into use.

- The accuracy of the system can be improved by bettering the data set in Haar Cascade Algorithm.

- Identification of vehicles in various angles by the system can be brought into the haar cascade to improve its accuracy.

Bibliography

- Aditya Rao, Akshay Phadnis, Atul Patil, Tejal Rajput and Dr. Prof. Pravin Futane, "Dynamic Traffic System Based On Real Time Detection Of Traffic Congestion", 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) on, vol., iss.no., pp. 16-18 Aug 2018

- Anurag Kanungo, Ayush Sharma and Chetan Singla, "Smart Traffic Lights Switching and Traffic Density Calculation using Video Processing", 2014 RAECS UIET Panjab University Chandigarh on vol., iss.no., pp. 8 Mar 2014

- Tanvi Sable, Nehal Parate, Dharini Nadkar, Swapnil Shinde, "Density and Time based Traffic Control System using Video Processing", ITM Web of Conferences 32 (ICACC - 2020) on, vol., iss.no., pp. Jan 2020

- https://www.raspberrypi.com/documentation/computers/configuration.html

- https://pysource.com/2020/04/02/train-yolo-to-detect-a-custom-object-online-with-free-gpu/