

CSE1901 - Technical Answers to Real World Problems (TARP)

Project Report

Smart Traffic Management Using Deep Learning

By

20BAI1087	Pranav Bhanot
20BCE1420	Karthik Raj
20BCE1438	Avneesh Kasture
20BCE1442	Shivam Sharma
20BCE1479	Manvik Sreedath

B. Tech Computer Science and Engineering

Submitted to

Dr Geetha S

School of Computer Science and Engineering



April 2023

Table Of Content

Chapter 1: Introduction	4
1.1 Abstract	4
1.2 Problem Definition	5
1.3 Motivation	6
1.4 Application Scenario	10
Chapter 2: Literature Survey	11
Chapter 3: Architecture	17
3.1 Object Detection and Recognition Process	17
3.2 Architecture brief of models used	18
3.3 Cascade classifier for Vehicle Detection	22
3.4 Modules	23
Chapter 4: Methodology	28
4.1 Analysis of Problem Statement	28
4.2 Brief of Proposed System	29
4.3 Project Component Specification	29
Chapter 5: Implementation	32
5.1 Experimental Setup and Technical Specifications	32
5.2 Cost Of Equipment	33
5.3 Dataset Preparation	33
5.4 Interfacing with Raspberry Pi	34
5.4.1 Onboard Vehicle Detection	36
5.4.2 LED Switching Mode	38
Chapter 6: Results	40
6.1 Results of Experimental Tests	40
Chapter 7: Conclusion and Future Scope	42
7.1 Conclusion	42
7.2 Future Scope	42
Chapter 8: Single Unique Factor	43
Chapter 9: Site Visit	43
Chapter 10: Sample Codes	43
Chapter 11: References	44

List of Figures

Fig 1.1 Percentage of Accidents and Deaths	6
Fig 1.2 Percentage of Injured, Killed, Accidents	7
Fig 1.3 Growth of Vehicles Over The Years	7
Fig 3.1 Comparison of Multiple Object Detection Model Architecture	16
Fig 3.2.1 Cascade Classifier	18
Fig 3.2.2 FPS vs Previous Object Detection Models	19
Fig 3.2.3 Yolov5 Model Architecture	20
Fig 3.2.4 YOLO v5s	21
Fig 3.3.1 Cascade Classifier on Vehicle Detection	23
Fig 3.4.1 Raspberry Pi	24
Fig 3.4.2 Raspberry Pi Camera	24
Fig 3.4.3 LED	25
Fig 5.3.1: Dataset Of Image Before Annotating	34
Fig 5.3.2: Dataset Of Image After Annotating	35
Fig 5.4.1 Interfacing with Raspberry Pi	37
Fig 5.4.2: Vehicle Detection Function	38
Fig 5.4.3: Traffic junction model	39
Fig 6.1 Model detecting vehicles in a real-time video	40
Fig 6.2 Green LED glows for excessive traffic road	41

List of Tables

Table 4.3 List Of Components Used	29
Table 5.1: Specifications of Software	30
Table 5.2: Specification of Libraries	30
Table 5.3: System Specification	31
Table 5.2.1 Equipment Cost	31

Chapter 1: Introduction

1.1 Abstract

Traffic management is the practice of controlling how vehicles, pedestrians and other road users move on public roads, highways and transportation systems to ensure safety, performance and safety.

Deep learning-based traffic management is a field of research that uses artificial intelligence on traffic data to increase safety, improve traffic management and reduce the number of closures.

Deep learning is a branch of machine learning that mimics the way the human brain learns through neural networks. Deep learning algorithms can be trained on large volumes of traffic data, including traffic, travel time and accident data, to recognize patterns and make predictions in traffic management content. For example, deep learning models can be used to predict traffic flow throughout the day, the locations of accidents and hotspots, and to adjust the timing of traffic lights. The ability of deep learning to process and analyze large amounts of data in real time is one of its key strengths in traffic management. This allows traffic controllers to make accurate and timely decisions, ultimately improving traffic flow and safety.

There are also problems with using deep learning for traffic control. The need for high-quality data, the potential for bias in the data or process, and the need for machine learning and data intelligence are a few of them. The traffic light generation process is as follows:

Data collection: Sensors such as cameras, radar and loop detectors are placed at intersections to collect instant data on traffic, vehicles and pedestrians and other variables.

Analysis: The data collected by the sensors is sent to the central control, which uses algorithms to analyze the data and determine the best time of the signal for each operation.

Adjustment: The control system then adjusts the timing of traffic lights based on analysis to improve traffic flow, reduce congestion and increase safety.

Monitoring: The system continuously monitors traffic and adjusts signal timing as needed in response to changing traffic patterns.

Smart traffic lights help create more efficient and safer traffic by using real-time data and advanced techniques to improve traffic flow.

Proper management and control of vehicles should prevent accidents, ensure safety and reduce travel time.

Modern computer vision technology makes it possible to create intelligent vehicle control that can improve traffic flow.

1.2 Problem Definition

Traffic congestion is a major problem facing cities around the world. Poor traffic control not only causes delays, but also causes pollution and fuel consumption. Traffic lights are an important part of traffic management, but currently, traffic lights are not smart enough to adapt to traffic changes.

The purpose of this project is to create a traffic light that uses deep learning techniques to adapt to real-time conditions and improve traffic flow. The system must be able to learn and predict traffic patterns and decide when to signal green, amber or red at each intersection.

This work includes the collection and processing of real-time traffic data from various sources such as traffic cameras, sensors and GPS devices. The data will be fed into a deep learning model, which will be trained to predict traffic patterns and determine the best light timing. The system should be able to adapt to changes in traffic patterns and adjust the time accordingly. Lighting should reduce traffic, improve traffic flow, reduce fuel consumption and emissions, and ultimately be more efficient and sustainable.

One of the leading causes of traffic build up is poor traffic flow management, which is often due to the use of fixed timing or manual adjustments for traffic lights. Smart traffic lights aim to overcome this challenge by utilizing real-time data and intelligent algorithms to optimize traffic flow based on current road conditions, including traffic volume, weather, and accidents. This approach aims to minimize the time drivers spend in traffic, reduce carbon emissions by minimizing idling time, and improve safety for pedestrians.

Our proposed model aims to address the issues related to Traffic Congestion, Safety and Pollution.

1.3 Motivation

According to a report by the Ministry of Road Transport and Highways, India had 4.12 lakh road accidents in 2021, resulting in 1.53 lakh deaths and 3.84 lakh injuries.

According to a study conducted by a global consultancy firm, traffic congestion costs the economy Rs 1.47 lakh crore annually.

The study was conducted during peak hours in Delhi, Mumbai, Bengaluru and Kolkata.

Traffic congestion increases vehicle emissions and degrades ambient air quality, and recent studies have shown excess morbidity and mortality for drivers, commuters and individuals living near major roadways.

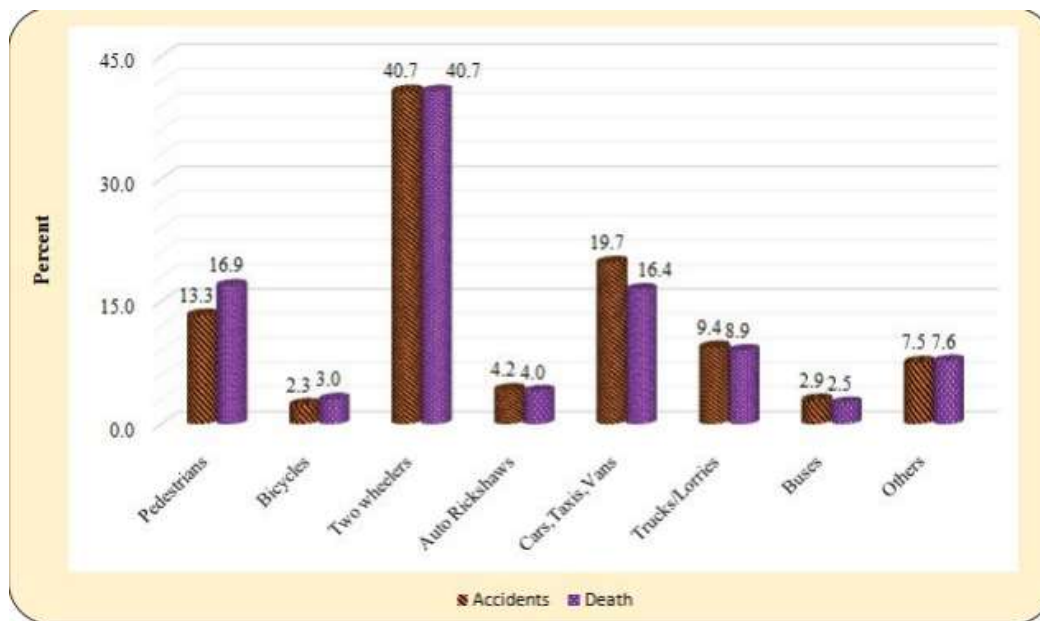


Fig 1.1 Percentage of Accidents and Deaths

Two-wheelers are in more accidents and have the highest death rate according to the ministry of road transport and highways. Especially in high traffic areas with high congestion it makes it more likely for two-wheelers to be caught in an accident.

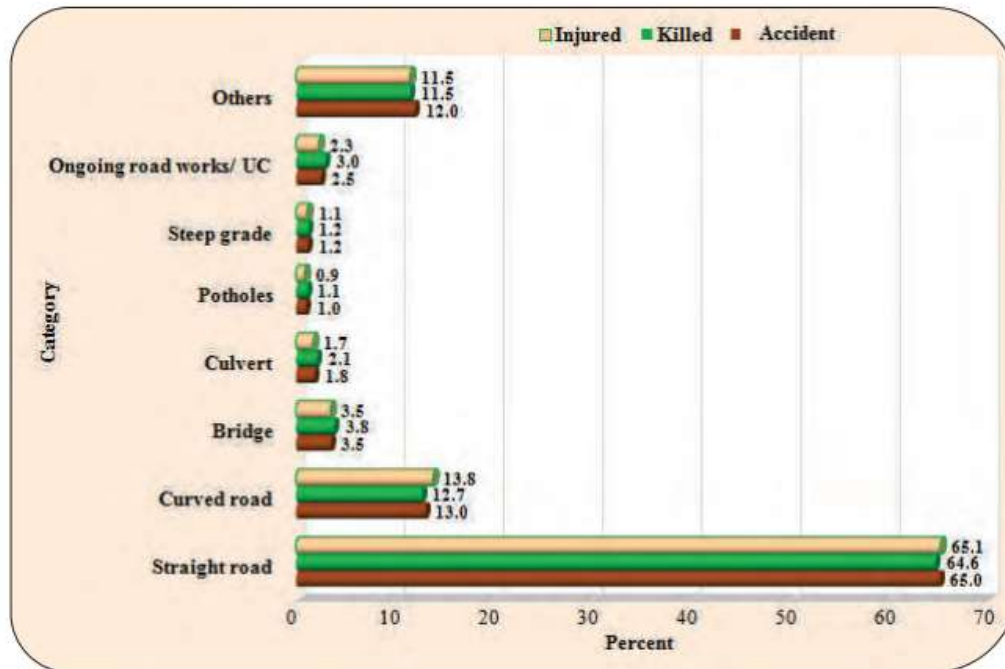


Fig 1.2 Percentage of Injured, Killed, Accidents

With the number of accidents being higher in open roads, accidents are more likely to occur in open roads with traffic congestion.

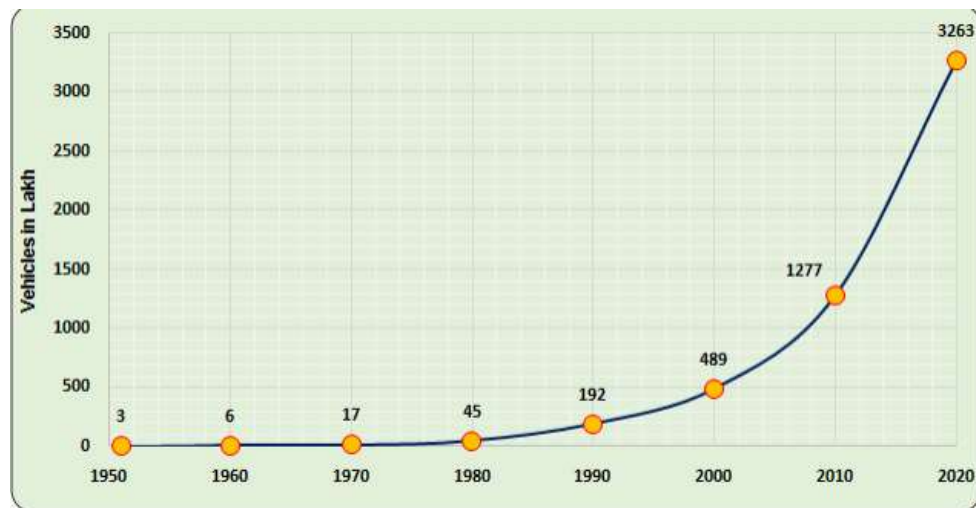


Fig 1.3 Growth of Vehicles Over The Years

There is an increasing trend in the number of roads being constructed with a large percentage of them being urban roads as well as the fact that the number of vehicles owned are also showing an increasing trend leads to the increasing need for traffic management and control.

Thus, we can see that overall traffic management is a necessity and could help save the environment by polluting lessens and also save human lives.

Intelligent vehicle management using machine learning (ML) and Internet of Things (IoT) technology can improve transportation and security. There are many reasons why this technology is used in traffic management.

First, traffic accidents are a major problem in cities, causing delays and increasing pollution.

Traffic management systems can monitor traffic in real time, predict future traffic patterns, and optimize traffic times using machine learning and IoT. By doing this, the system can reduce congestion, shorten travel time, and reduce fuel consumption and emissions.

In addition, the system can use machine learning algorithms to provide drivers with personalized travel advice, such as making recommendations based on their driving habits.

Finally, the implementation of intelligent traffic management can bring economic benefits. By reducing traffic congestion, the system can reduce fuel consumption and save money for people and businesses. In addition, the system enables more efficient use by helping authorities make informed decisions about infrastructure development and traffic regulations.

In summary, there are many benefits of using machine learning and IoT in traffic management, such as reducing traffic accidents, improving safety, improving user experience and benefiting business interests. The majority of object recognition techniques used in traffic flow management are still based on conventional models like CNN and R-CNN, despite the fact that deep learning-based approaches have significantly improved in generic object detection in recent years. As a result, we suggest a computer vision-based smart traffic control model that makes use of recent developments in object detection and deep learning.

Therefore, it is not surprising that many cities are using this technology to solve their increasing urban mobility problems.

Smart traffic lights have recently been used with deep learning techniques to improve traffic flow, reduce congestion and enhance safety. The idea/concept of using machine learning for traffic management is fairly old with one of the oldest papers on it - "Real-time decision support for air traffic management, utilizing machine learning" which was published in 1996 and focuses on air traffic management but due to the complicated implementation and low advancement in machine and deep learning wasn't the focus of the research world at the time. The use of deep learning for traffic management draws on earlier work in data mining and machine learning, but it has gained popularity recently as a result of improvements in deep learning algorithms such as YOLO, cascade classifiers, etc and also due to the accessibility of large traffic datasets. Since then, an increasing number of studies have investigated different deep learning applications for traffic management, such as the prediction of traffic flow, the detection of traffic incidents, and the optimization of signal timing.

Here is how recently developed deep learning algorithms can be used to implement smart traffic lights with the help of IoT sensors and actuators:

Object Detection: Deep learning algorithms are used to detect objects such as vehicles, bicycles, pedestrians, and animals at intersections. These algorithms use computer vision techniques to analyze live video feeds from cameras installed at the intersections. This helps the traffic light system to adjust the signal timings based on the real-time traffic situation.

Predictive Traffic Analytics: Deep learning models are trained on historical traffic data to predict future traffic volumes and congestion patterns. The models use a variety of data sources, such as weather forecasts, events calendars, and social media feeds, to predict the expected traffic conditions. Based on these predictions, the traffic light system can adjust signal timings and optimize traffic flow.

Intelligent Signal Control: Deep learning algorithms are used to optimize traffic signal timings based on real-time traffic data. The algorithms can dynamically adjust the timings of the green, yellow, and red lights based on the volume and speed of vehicles at the intersection. This helps to reduce congestion and improve the overall efficiency of the traffic system.

Emergency Vehicle Detection: Deep learning models are used to detect emergency vehicles such as ambulances and fire trucks, and prioritize their movement through the intersection. The models can analyze the unique sound signatures of emergency vehicle sirens and use this information to automatically adjust the traffic signal timings to allow them to pass through safely and quickly.

Machine learning-enabled smart traffic lights have the potential to revolutionize the way we manage traffic flow at busy intersections. Our project analyzes the benefits of this technology and its impact on research and innovation in the transportation industry. Our project finds that machine learning-enabled smart traffic lights offer numerous benefits over traditional traffic light systems. By analyzing real-time traffic patterns and adapting traffic light timings accordingly, these systems can significantly reduce waiting times and increase the overall efficiency of traffic flow. This, in turn, can reduce fuel consumption and air pollution, improve the driving experience, and increase safety on the roads. Moreover, machine learning-enabled smart traffic lights offer significant value to research and innovation in the transportation industry. These systems generate vast amounts of data on traffic flows, road conditions, and weather patterns, which can be used to develop more advanced machine learning algorithms and improve the overall efficiency of transportation infrastructure. Furthermore, the implementation of machine learning-enabled smart traffic lights can drive further innovation in related fields such as autonomous vehicles. As more autonomous vehicles hit the roads, smart traffic light systems can help them navigate through intersections more efficiently, ultimately reducing congestion and increasing safety. In conclusion, machine learning-enabled smart traffic lights offer numerous benefits to the transportation industry and have the potential to drive further research and innovation. Governments and transportation authorities should consider implementing these systems in high-traffic areas to improve the efficiency and safety of their transportation infrastructure and drive further innovation in the industry.

1.4 Application Scenarios

The following are some real-time use cases of smart traffic lights:

1. Real-time traffic management: Intelligent traffic lights use real-time data to adjust the timing of traffic lights to improve traffic flow. This means that during rush hour the green light will last longer in most traffic directions, allowing more vehicles to pass.

2. Emergency: Traffic lights can use sensors to detect and prioritize emergency vehicles. This may include delaying green times, clearing intersections and changing lights so that emergency vehicles can pass quickly and safely.

3. Pedestrian Safety: Smart traffic lights can use sensors to detect pedestrians and adjust the lights so they can cross the road safely.

This may include delaying crossings, providing audible or visual signals to indicate that it is safe to cross, and even stopping traffic in all directions to allow pedestrians to cross safely.

4. Traffic monitoring and control: Traffic lights can be used to monitor traffic patterns and adjust traffic lights to improve traffic flow. This can include adjusting the timing of the lights, changing the light sequence, and providing drivers with real-time information about traffic conditions.

5. Environmental benefits: Good lighting can help reduce traffic accidents, which reduces pollution and greenhouse gas emissions. By improving traffic flow, lighting can help reduce idle time in traffic, which can affect the city's air quality.

Junction safety: Smart traffic lights can use sensors to detect when a vehicle is approaching an intersection too quickly and adjust the light accordingly to prevent accidents. This may include delaying the yellow light to give drivers time to stop, or even turning the light green to allow traffic to cross the intersection quickly and safely.

6. Dynamic Lane Management: Traffic lights can be used to control lanes according to traffic conditions. This may include changing the direction of traffic in a lane, opening and closing lanes based on traffic, or even creating temporary lanes during rush hour.

Chapter 2: Literature Review

A unique technique that makes use of computer vision to enhance traffic flow and reduce the average waiting time at traffic crossings is presented by Kelathodi Kumaran, S., Mohapatra, et al. in 2019. The authors share two video datasets with the community of computer vision researchers as another contribution (4WAY and IITBBSR). The proposed system makes use of a cutting-edge inference technique that can successfully track clusters that represent moving objects even when they are obscured during tracking. By detecting cluster mobility inside regions-of-interest (ROI) marked at the entry and exit sites of intersection approaches, cluster counts of approach roads are used for signal timing. The research also suggests a novel method for intelligent signal timing based on traffic flow, employing a modified K-means algorithm to temporally cluster optical flow features of moving cars.[1]

An approach for non-stop passage of signal-controlled crossings using dynamic signs and computer vision is put forth in "Method of non-stop passage of signal-controlled intersections using computer vision" by Vladimir Gorodokin et al. The authors provide two novel low-cost processing approaches, one of which was built with hardware assistance and the other without, both of which have been created with little overhead. The technique enables the use of already-installed video surveillance systems to watch traffic, gather data on how conditions are changing in real-time, and count precisely how many vehicles are passing through an intersection. It is suggested to use dynamic road signage displaying the top speed restriction at a road section heading towards a signal-controlled road network junction to ensure nonstop coordinated traffic.[2]

Traffic signals play a significant part in the transportation system and provide a safe drive at road crossings, according to the article titled "Traffic Management using Computer Vision and SUMO" by Bodicherla Digvijay Sri Sai, Ramisetty Nikhil, Payarda Santosh Babu, and Nenavath Srinivas Naik. In order to handle traffic signals more effectively, the study suggests a novel strategy that makes use of computer vision and SUMO (Simulation of Urban Mobility).

Conventional signal scheduling minimizes traffic congestion by granting green lights in each direction for a set period of time. The suggested method makes appropriate adjustments to the signal timings based on computer vision measurements of the quantity of cars at each intersection. This facilitates a smooth flow of traffic and helps to lessen traffic congestion.[3]

In August 2013, Ninad Lanke and Sheetal Koul published a paper in the International Journal of Computer Applications with the title "Smart Traffic Management System." The paper explores Radio Frequency Identification (RFID), a brand-new technology that can be used with current technologies to produce a smart traffic control system. The authors describe how RFID can be

applied to track moving objects and deliver real-time traffic data. Moreover, the system can be utilized to manage traffic lights and ease congestion.[4]

“Geometric invariant features for the detection and analysis of vehicle” by Anandhalli et al. discusses how shape descriptors can be used in conjunction with rule-based classifiers to detect vehicles. The process entails extracting features from the input image using shape descriptors and contours to determine if the input is a vehicle or not. Using shape descriptors such as Eigenvalues and Hu-Moments, the features are retrieved. For further classification of vehicle and non-vehicle images, the classifier is trained using extracted characteristics. The tests are carried out on several benchmark datasets, and the outcomes are examined while taking the total classification accuracy into consideration. [5]

By using laser sensors to measure traffic density, the system described in Prasad et al’s “Smart Traffic Monitoring and Controlling” implements dynamic temporal slots with various levels. The system utilizes a cutting-edge concept for an IOT-based traffic monitoring and control system. With this system, a central unit will be used to regulate the traffic lights at each intersection in the city. The traffic controller will have an Android phone with the complete route information. Via an IOT link, the phone communicates with the system. The controller must enter the password in order to connect to the system and operate the traffic lights. The traffic police generally maintain control over the current traffic system. This system, which is managed by the traffic police, has a major flaw in that it is not intelligent enough to handle traffic congestion. It is totally up to the traffic police officer's judgment whether to continue to block one road or allow traffic on another to pass, therefore the official's choice may not be wise enough. Also, even when using traffic lights, the duration for which the drivers will see a green or red signal is fixed. As a result, it might not be able to address the issue of traffic congestion. The proposed system solves this problem. [6]

A research paper titled “Intelligent Traffic Light Control” was published by Marco Wiering, Jelle van Veenen, Jilles Vreeken, and Arne Koopman in 2004 which presents an adaptive optimization algorithm based on reinforcement learning for simulating and optimizing traffic control algorithms in a city. The authors have implemented a traffic light simulator, Green Light District, that allows them to experiment with different infrastructures and to compare different traffic light controllers. Experimental results indicate that their adaptive traffic light controllers outperform other fixed controllers on all studied infrastructures. The paper describes how reinforcement learning with road-user-based value functions is used to determine optimal decisions for each traffic light. The decision is based on a cumulative vote of all road users standing for a traffic junction, where each car votes using its estimated advantage (or gain) of setting its light to green. The gain-value is the difference between the total time it expects to wait during the rest of its trip if the light for which it is currently standing is red, and if it is green. The

paper also surveys several previous approaches to traffic light control and introduces some limitations to the usage of intelligent traffic control. [7]

A model that describes the evolution of queue lengths in each lane as a function of time at a junction of two two-way streets with adjustable traffic lights on each corner is covered in "Optimal Traffic Light Control for a Single Intersection" by B. De Schutter and B. De Moor. In addition, this system's optimal and suboptimal traffic signal switching schemes—possibly with variable cycle lengths—are discussed in the study. The strategy described in this study differs significantly from the majority of other approaches in that it allows the green-amber-red cycle time to change from cycle to cycle.

In contrast to current models, the study "IntelliLight: A Reinforcement Learning Method for Intelligent Traffic Light Control" suggests a deep reinforcement learning model for traffic light control¹. The authors demonstrate several intriguing case studies of policies that were learned from the real data while testing their method on a sizable real traffic dataset gathered from security cameras. An effective transportation system must have intelligent traffic light regulation. An intelligent traffic light management system should dynamically respond to real-time traffic, as opposed to existing traffic lights, which are typically operated by hand-written rules. The research demonstrates a new trend in traffic light control utilizing deep reinforcement learning techniques, and recent experiments have produced encouraging findings.[9]

“Intelligent traffic management system for cross section of roads using computer vision” - Osman, T., Psyche, S. S., Ferdous, J. S., & Zaman, H. U. (2017, January) explores how to address the difficult optimisation problem of traffic signal control using cutting-edge AI methods. With AI features, reinforcement learning is a common technique. This technique involves allowing one or more agents (such as traffic signals) to interact with the environment (such as automobiles) and use what they learn to gradually make better decisions. The paper offers a deep Q learning-based approach to route planning. It also goes through previous uses of agent-based research with traffic signals. In a single intersection scenario, the study demonstrates how RL and Deep Q-Network (DQN) algorithms may enhance the state of intersections and decrease vehicle waiting time by more than 20%.

Khan et al. gave a solution to traffic congestion control which was a deep learning model-based solution for smart cities. For region-based traffic flow forecasts in smart cities, a hybrid model based on convolution neural network (CNN) and long short term memory (LSTM) architectures is used in this study. This paper is distinctive in the way it uses deep learning models to create a data-fusion-based traffic congestion management system. The CNN-LSTM hybrid model that was used in this study was trained on a sizable dataset of traffic movement data that was gathered from numerous sources, including GPS devices, cameras, and sensors. The model uses data from various sources to forecast traffic flow patterns in real-time. The suggested system can be used to reduce traffic congestion by anticipating patterns of traffic flow and modifying traffic signals appropriately.

In order to increase the effectiveness of traffic light controlling, Thakare et al. suggest a traffic light controller that combines sensor networks with embedded technology. The system makes an effort to minimize traffic congestion brought on by signal lights. A microcontroller, which serves as the system's brain, is the foundation of the system. The system uses embedded technologies, sensor networks, and ultrasonic sensors on the sides of the highways. The microcontroller and microprocessor used in traffic light controllers (TLC) have restrictions since they employ pre-defined hardware and operate in accordance with programmes that are not flexible enough to be changed in real-time.

A technique for performing unsupervised domain adaptation among various cameras to accurately count the number of vehicles in a city was proposed by Luca Ciampi et al. (2020). Although the suggested methodology focuses on counting vehicles, it can be used to count any other kind of object.

This paper is unique in the way it uses multiple camera domain adaptations to create image-based vehicle density estimators with little labeled data. The suggested methodology can be used to track vehicle movements in urban areas and enhance the livability of residents.

The findings of this article demonstrate that the suggested methodology can count vehicles unsupervised with an accuracy of about 90%.

In order to gather data on traffic flow, Zhe Dai et al. (2019) suggest a video-based vehicle counting framework that uses a three-step method of object detection, object tracking, and trajectory processing. The suggested methodology can be broken down into three stages: dataset creation, model development for vehicle detection, and vehicle tally.

The strategy of Zhe Dai et al. was creating a deep learning framework for video-based vehicle counting that sets it apart from others in the field. The suggested approach can be used in a variety of traffic situations and for real-time vehicle counting. The findings of this paper

demonstrate that the suggested technique for vehicle counting can achieve an overall accuracy of more than 87.6%.

A cognitive traffic management system (CTMS) based on the Internet of Things approach is described by Miz et al. (2014). The paper suggests a smart traffic light that could be a component of CTMS and the IoT. This infrastructure's primary goal is to handle road traffic in an efficient and ideal manner.

This paper is unique in how it approaches creating a smart traffic light that can be included in the IoT as a component of CTMS. The suggested methodology can be used to improve government authorities' decisions regarding traffic administration. The findings of this paper demonstrate how traffic control procedures can be improved and errors brought on by human factors can be eliminated.

Alaidi et al.'s paper from 2020 details a smart traffic control system made with Arduino to address the issue of congestion at the Dor al Moalemen intersection in Wasit City. The suggested method makes use of an Arduino Mega, an ultrasonic sensor, and an esp32 camera to analyse and handle regular traffic at a three-line intersection.

This paper is unusual in that it uses Arduino to create a smart traffic control system that alleviates congestion at the Dor al Moalemen intersection in Wasit City. The suggested approach can be used to resolve the issue of transportation congestion in Kut, Iraq.

The findings of this article demonstrate the implementation of the smart traffic light system (STLS) using an Arduino, camera, and IR sensor to resolve the issue of traffic congestion in Kut, Iraq.

A clever traffic management system based on multi-agent systems (MAS) was proposed by C. Huang et al. to enhance traffic flow and lessen congestion. The suggested approach makes use of a multi-agent system to control traffic signals, provide dynamic route guidance, and govern traffic flow while reducing congestion.

This paper is unique in that it takes a multi-agent systems (MAS)-based approach to developing a smart traffic management system that will enhance traffic flow and relieve congestion. The suggested approach can be used to solve the issue of transportation congestion in smart cities. The findings of this article demonstrate that the suggested multi-agent system (MAS)-based smart traffic management system can enhance traffic flow and lessen congestion.

Kapileswar et al. in their paper surveyed current urban traffic management strategies for priority-based signaling and lowering congestion and the Average Waiting Time (AWT) of vehicles. In order to avoid congestion, give emergency vehicles priority, and reduce the amount of time that vehicles spend waiting at intersections (AWT), the paper emphasizes that substantial research has been done on Traffic Management Systems using Wireless Sensor Networks (WSNs).

The originality of this article resides in the way it presents a survey of the existing priority-based signaling and congestion-reduction urban traffic management schemes. The paper also emphasizes how traffic congestion may cause a static control system to block rescue vehicles.

The findings of this paper demonstrate that Wireless Sensor Networks (WSNs) have attracted growing interest in traffic monitoring and preventing traffic jams². The suggested approach can be used to solve the issue of transportation congestion in smart cities.

Chapter 3: Architecture

3.1 Object Detection and Recognition Process

Object detection is a computer vision task that is important to many consumer applications today, such as surveillance and security, cell phone recognition, and diagnostics with MRI/CT scans. Product search is also an important part of increasing promotion. Autonomous vehicles rely on awareness of their environment for safe and efficient driving performance. The sensor uses object detection tools to accurately identify objects around the vehicle, such as pedestrians, vehicles, traffic signs, and obstacles. Deep learning-based object detection plays an important role in finding and repairing these objects in real time.

Vehicle detection using deep learning is done by R-CNN and bounding box features are combined with CNN features. Detection with R-CNN is a two-way process in which weak regions with potential objects are identified and detected object regions are classified in the second step. Here, the feature extraction network is basically the pre-trained ResNet-101 architecture. The first subnet is a regional network trained to classify objects from the background, and the second subnet is trained to identify objects (cars, pedestrians, etc.) individually. For training, the mini batch size property is set to 1 to recognize vehicles at different intervals. If the object to be detected has a size, the size can be greater than 1.

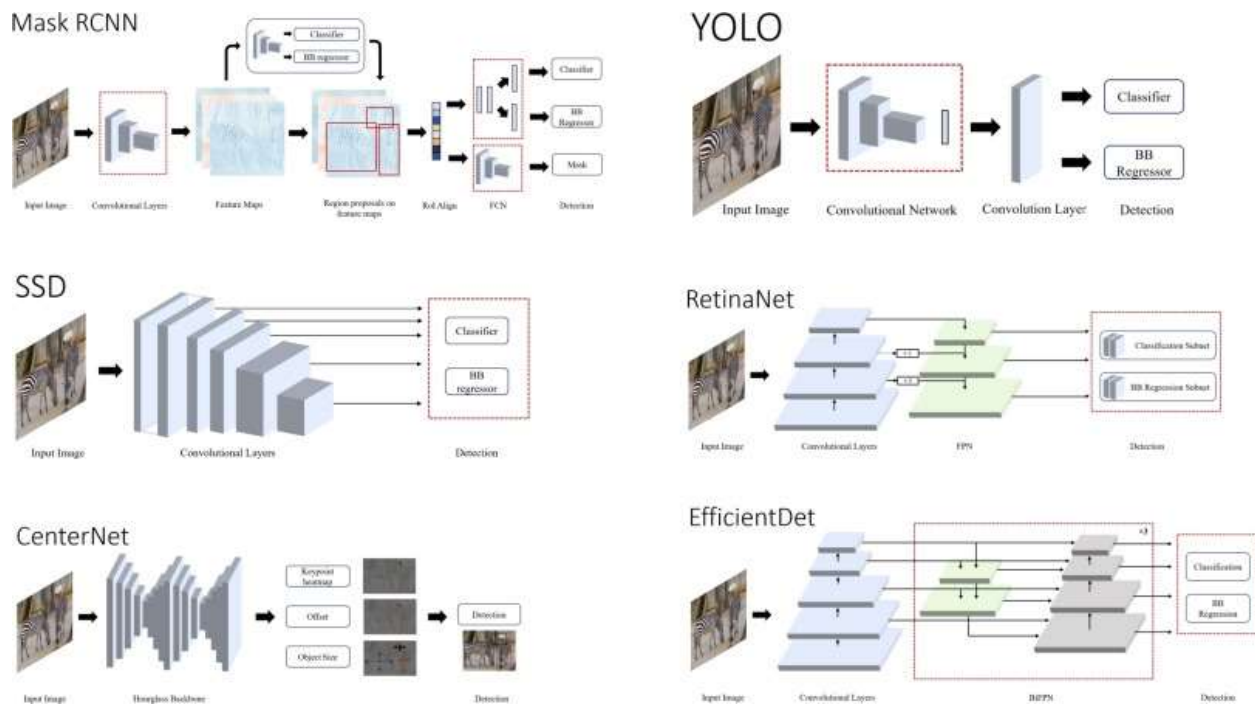


Fig 3.1 Comparison of Multiple Object Detection Model Architecture

An object discovery model based on deep learning usually has two parts. An encoder takes an image as input and writes it through a series of blocks and layers that learn to extract features used to find and save objects. The encoder's output is then forwarded to the decoder, which envisions a linked box and label for each object.

The simplest decoder is a pure regressor. A regressor is connected to the output of the encoder and directly estimates the position and size of each bounding box.

The output of the model is the X, Y function of the object pair and its position in the image.

Although simple, these models are limited. You must first specify the number of boxes. If your photo has two dogs, but your model is designed to capture only one object, one of them will not have a name. However, a single regression model is a good choice if you know in advance how many items you need to predict in each graph.

An extension of the regression method is the regional consensus network. In this decoder, the model represents the viewport that the object thinks it will be. Pixels in these areas are then placed in a subnet to determine text (or reject the offer). It then runs the pixels containing these areas through the distribution network. The benefit of this approach is a more accurate and flexible model that can suggest more spaces that can contain containers.

SSD creates lots of conflicts because each port has multiple boxes and the ports will be close together. SSD output should be post-processed to eliminate most guesswork and choose the best.

The most popular finishing technique is called unlimited. Object detectors display the location and label of each object, but how do we know how well the model is performing?

The most commonly used metric for the location of parts is intersection over union (IOU). Given two connected boxes, we calculate the intersection area and divide it by the junction area. The value ranges from 0 (no overlap) to 1 (total overlap).

3.2 Architecture brief of models used

1. Haar Cascade Classifier

Cascading classifiers are a type of machine learning algorithm often used in computer vision to detect objects in images or video streams. It is particularly useful for evaluating products with special lighting properties that can be reliably seen in a cascading stage. The main idea behind the classifiers array is to use an increasingly complex process to identify objects of interest. Each stage of the cascade has a simple classifier that quickly rejects image fields that do not contain objects, followed by a complex classifier for fields that go through previous phases. This process goes all the way to the final stage, which is usually a product that can identify the product.

One of the main advantages of the cascading classifier is its speed. By rejecting image regions with no objects, the algorithm can focus computational resources on regions likely to contain more objects. This can be faster, especially in real-time applications like video games.

Digit classifiers are generally used in applications such as face detection, object tracking and orientation recognition. They can be trained using a variety of machine learning algorithms, including AdaBoost and Support Vector Machines (SVM).

However cascade classifiers also have some limitations. They can be sensitive to changes in lighting or lighting conditions, which can reduce their accuracy.

Cascading classifiers work by filtering out non-objective regions of the image at all levels. Each level consists of multiple weak classifiers trained on a set of positive and negative samples. Weak classifiers are simple binary classifiers that determine whether a region of an image contains objects. At each level, the classifier evaluates the region of interest using the weak algorithm associated with that level. If the domain passes all weak classifiers at one level, it is considered a good domain and moves to the next level. If it fails in any of the weak classifiers, it is rejected and the classifier moves to the next relevant region. The cascading classifier architecture is designed to be efficient, allowing it to quickly filter out large amounts of images that are unlikely to contain objects. This makes it ideal for real-time detection devices such as surveillance devices and autonomous vehicles.

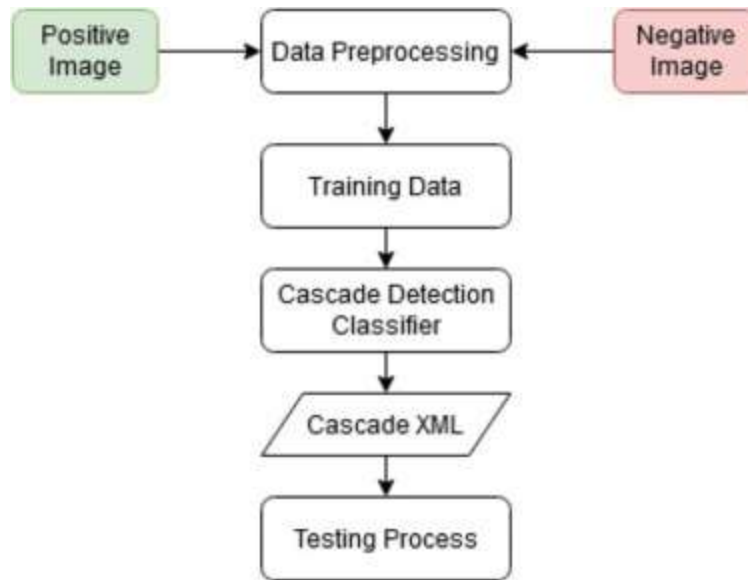


Fig 3.2.1 Cascade Classifier

2. YOLO v5

Since the YOLO v5 is a single-phase detector, it has three main features like other single-phase detectors.

- Model Spine
- Model Neck
- Model Head

Model Spine is used to extract important features from the input image. In YOLO v5, CSP - Cross Stage Partial Networks is used as the backbone to extract the richness and information of the input image. Handle is used to create pyramids. Feature pyramids help the model generalize well to scale objects. It helps to know the same product with different shape and size.

Feature pyramids are useful in helping to model the performance of invisible data. FPN, BiFPN, PANet, etc. There are other models that use different pyramid schemes, such as YOLO v5 used PANet as the backbone to obtain the pyramid shape.

Understanding the Feature Pyramid Networks (FPN) for product detection the Model Head is used for the detection end only. It uses junction boxes on properties and generates final output with useful classes, object scores, and bind boxes. The head of the YOLO v5 model is the same as the previous YOLO V3 and V4 versions.

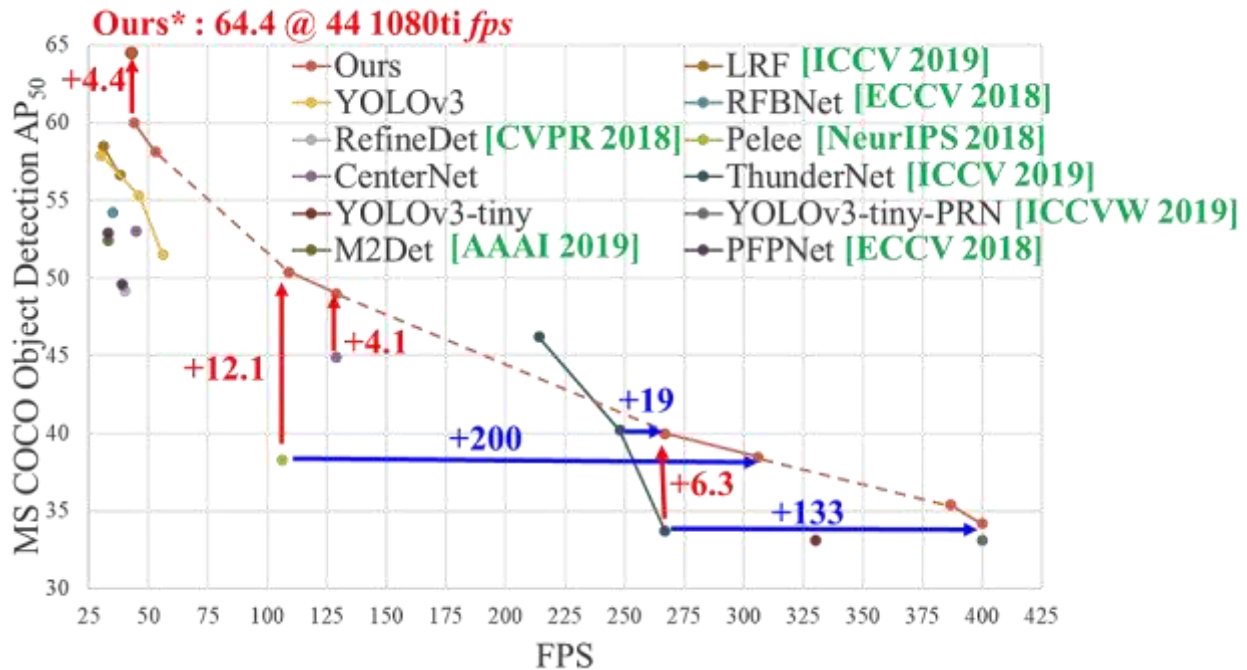


Fig 3.2.2 FPS vs Previous Object Detection Models

Activation Function

The choice of activation function is most important in any deep neural network. Recently Leaky ReLU, miz, swish etc. Many activations are available.

The authors of YOLO v5 decided to use Leaky ReLU and Sigmoid optimization.

YOLO v5 uses the Leaky ReLU activation function for the middle/hidden layer and the sigmoid activation function for the final detection process. You can check it here.

Optimization Function

We have two options for the optimization function in YOLO v5 SGD and Man

In YOLO v5, the default optimization function for training is SGD.

However, you can change it to Adam using the "--man" argument line.

Cost of Performance or Loss of Performance The YOLO family has a loss factor that includes negative scores, class scores, and backbox scores. Ultralytics uses binary cross-entropy and PyTorch's Logits loss function to calculate loss in value classes and product points. We can also select the Defocus function to calculate the loss.

We can optionally use the fl_gamma hyperparameter to find out the Focus Loss.

Weights, Deviations, Parameters, Gradients and Final Model Summary

The weights, deviations, shapes and parameters of each layer in the

You can also refer to the following brief summary of the YOLO v5 small model.

model details: 191 layers, 7.46816e + 06 parameters, 7.

46816e + 06 gradient

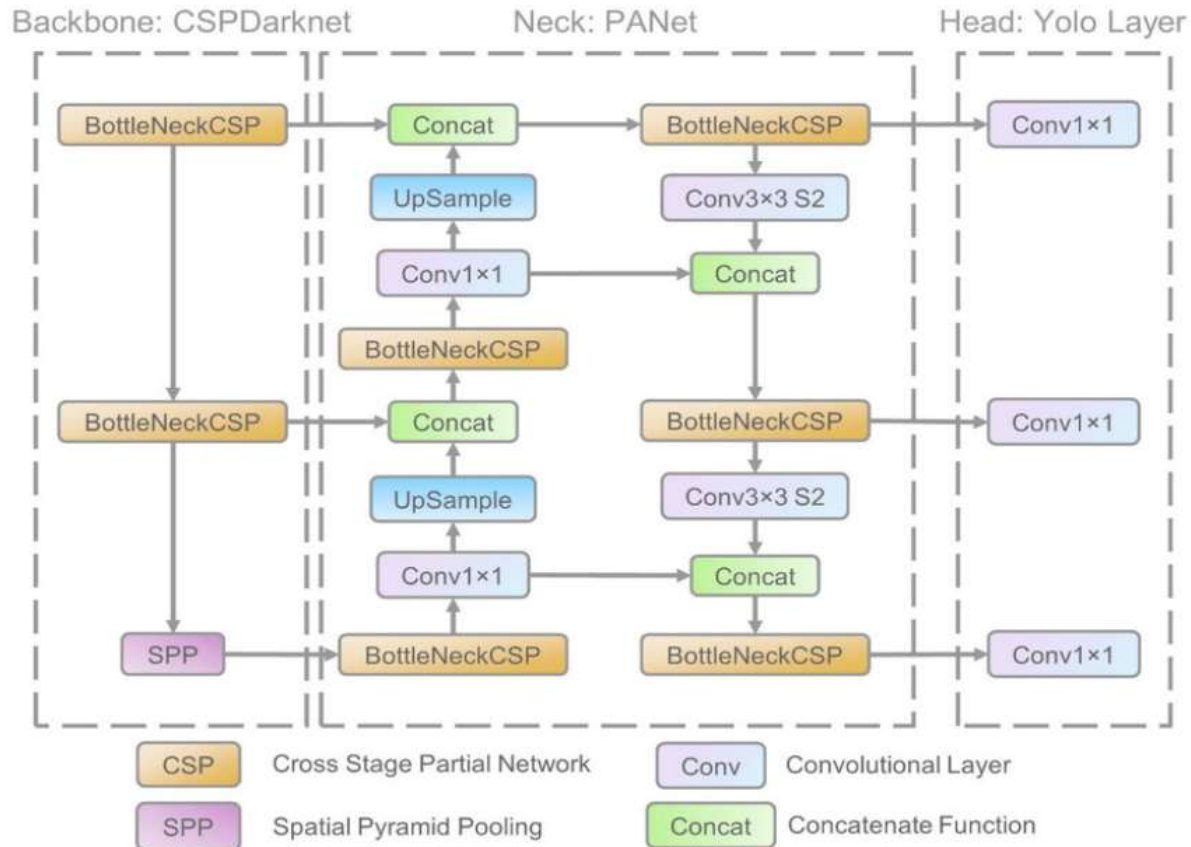


Fig 3.2.3 YOLOv5 Model Architecture

3. YOLO v5s

It is Ultralytics real time for product discovery in 2020. YOLO (You Look Alone) is an updated version of the product discovery algorithms. The "s" in YOLOv5s stands for "small", the smallest and fastest version of the YOLOv5. It contains fewer bugs than other versions (YOLOv5m, YOLOv5l and YOLOv5x), making it suitable to work on an unlimited number of devices such as mobile phones and embedded systems. YOLOv5s Detects and classifies multiple objects in live video or images simultaneously using a neural network. It achieves high accuracy using junction boxes and link-based search objects, and combines several enhancements such as feature pyramid networking, loss of focus, and other techniques to increase the accuracy of product detection. YOLOv5s is a powerful and efficient locator that can be used for a variety of applications including automotive, security and robotics, making it an excellent choice for our search engine model.

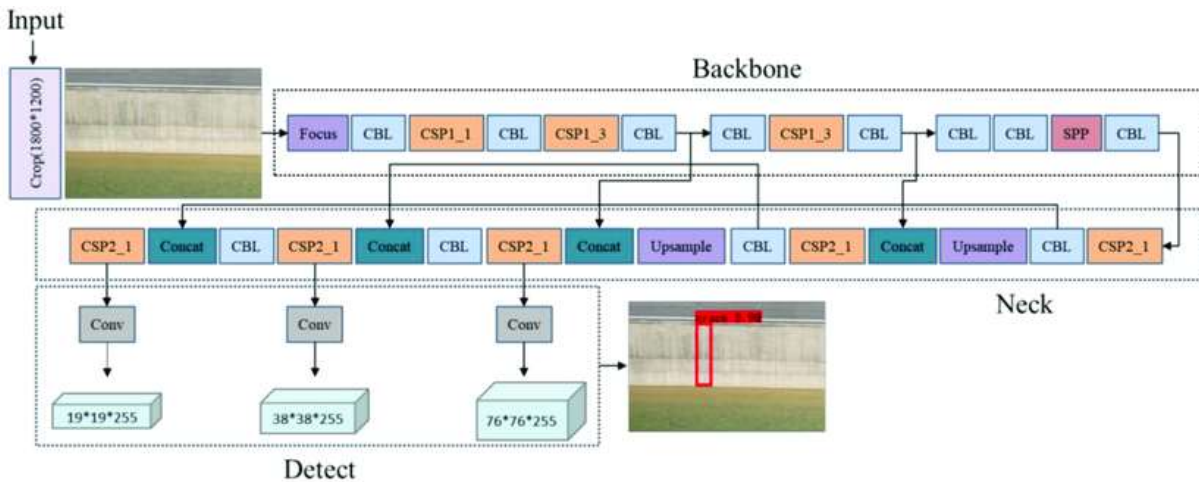


Fig 3.2.4 YOLO v5s

YOLOv5 is the most popular real-time search engine algorithm developed by Ultralytics. It is an evolution of the YOLO (You See Only One) family of algorithms and has significant improvements in accuracy and speed compared to its predecessors. YOLOv5 model architecture is based on modification of EfficientDet backbone, which is the best performance in network detection based on EfficientNet architecture.

The network is divided into three main parts:

1. Backbone Network: The backbone mesh is responsible for extracting features from the input image. The YOLOv5 backbone is based on the EfficientDet architecture, which consists of convolutional and feature fusion blocks. The backbone is pre-trained on the ImageNet dataset and can be fine-tuned for specific object detection tasks.
2. Neck Net: The neck net is responsible for connecting the features of the hind web and creating a series of different maps. The YOLOv5 necknet uses a modified version of the FPN (Feature Pyramid Network) architecture that allows the network to identify objects of different sizes.
3. Mainnet: The mainnet is responsible for estimating classes and bounding boxes from the custom maps generated by the necknet. YOLOv5's mainnet consists of convolutional and predictive algorithms that output useful classes and binaries for each object contained in the input image.

YOLOv5 model architecture is designed to be lightweight and efficient, enabling today's devices to achieve runtime. Its ability to detect and classify objects in a variety of situations makes it a popular choice for computer vision applications such as driverless cars, surveillance and robotics.

3.3 Cascade classifier for Vehicle Detection

The cascading classifier architecture is a popular technique for object detection in computer vision. It is widely used for vehicle detection in vehicle inspection and maintenance. The architecture consists of a set of classifiers trained to ensure the existence of an object of interest (in this case, vehicles) at various stages of the search.

The steps for vehicle detection using cascade classifiers are as follows:

Data Collection: The first step is to collect a large image dataset containing the cars. These images are used to train the cascade classifier.

Feature Extraction: The next step is to extract features from the training image. These features are often expressed as number vectors that capture the main features of the object of interest. In this case, features such as color, shape, texture are extracted from the vehicle image.

Training: Train a separate machine learning model after extracting features. A classifier is trained on good and bad examples.

Good examples are pictures with cars, bad examples are pictures without a car.

Cascading Classifiers: Ordered classifiers are then divided into a number of smaller classifiers, each designed to control objects at a certain level of investigation. Each classification is applied to a set of images and filters out artifacts. The output of the first classifier is used as the input of the second classifier, and so on.

Testing: The final step is to test the performance of the cascade classifier on validation images. Classifiers are evaluated for their ability to detect the presence of a vehicle and avoid false positives.

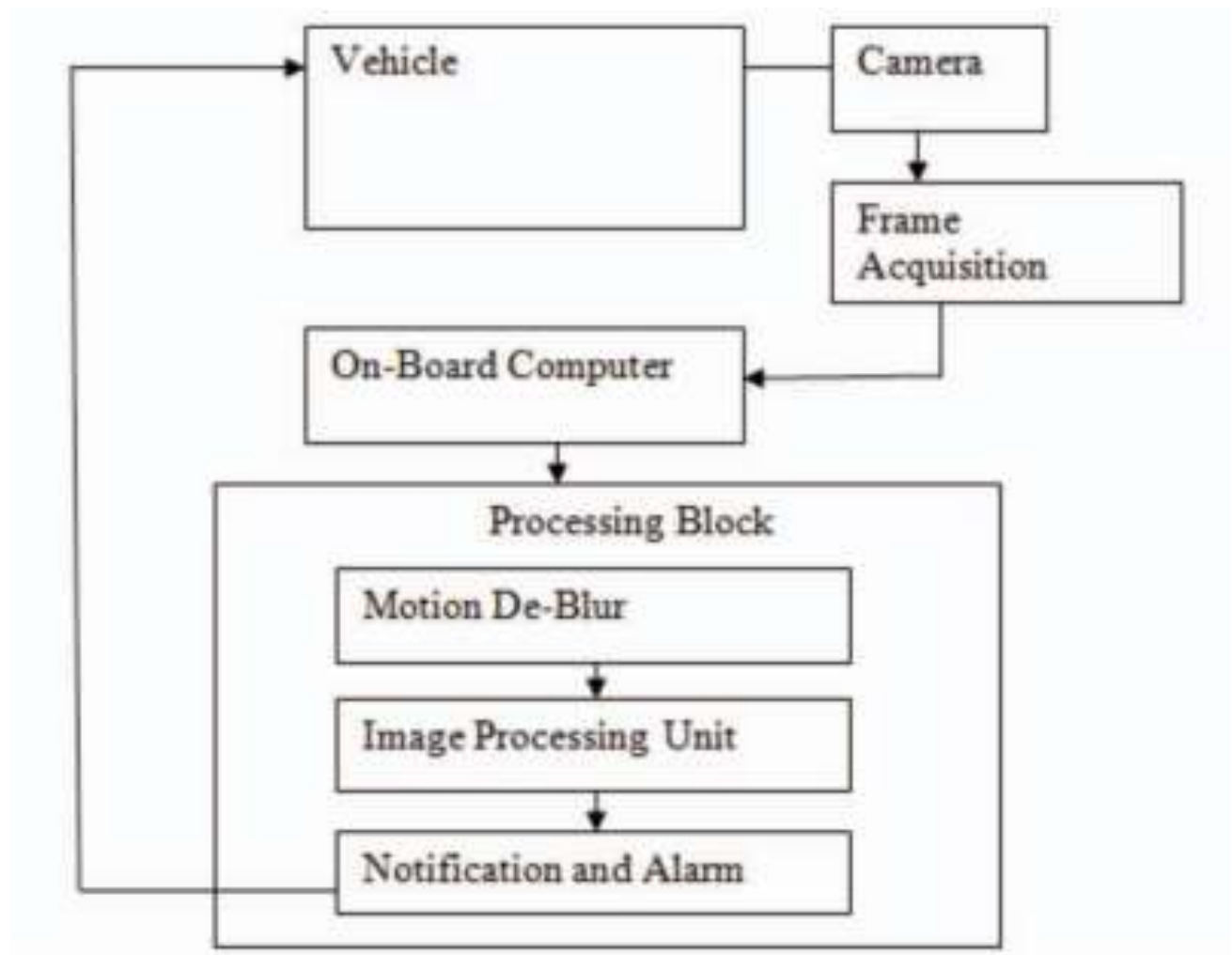


Fig 3.3.1 Cascade Classifier on Vehicle Detection

3.4 Modules

3.4.1 Raspberry Pi

The Raspberry Pi is a cheap computer that runs on Raspbian OS. It also provides a set of GPIO(general purpose input/output) pins that allow you to control electronic components for physical computing and explore the Internet of Things.



Fig 3.4.1 Raspberry Pi

3.4.2 Pi Camera

The Pi camera module is a portable lightweight camera that supports Raspberry Pi. It communicates with Pi using the MIPI camera serial interface protocol. It is normally used in image processing, machine learning or in surveillance applications. A 5MP camera is used to get a live stream of the traffic junctions.



Fig 3.4.2 Raspberry Pi Camera

3.4.3 LED

The 5V LEDs are used to simulate the traffic signals used in the real world. They are connected to the GPIO pins of the raspberry pi.

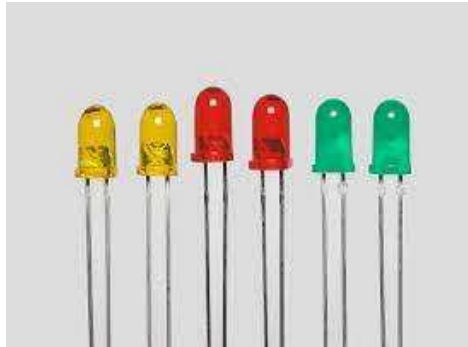


Fig 3.4.3 LED

Chapter 4: Methodology

4.1 Analysis of Problem Statement

The primary goal of traffic management is to enhance safety for both drivers and pedestrians while reducing congestion. One of the leading causes of traffic buildup is poor traffic flow management, which is often due to the use of fixed timing or manual adjustments for traffic lights. Smart traffic lights aim to overcome this challenge by utilizing real-time data and intelligent algorithms to optimize traffic flow based on current road conditions, including traffic volume, weather, and accidents. This approach aims to minimize the time drivers spend in traffic, reduce carbon emissions by minimizing idling time, and improve safety for pedestrians. Additionally, smart traffic lights can be integrated with other intelligent transportation systems to provide a more comprehensive solution to traffic management challenges. Smart traffic lights, or intelligent traffic systems (ITS), are advanced traffic management systems that use sensors, cameras, and other technologies to monitor traffic flow and optimize traffic signal timings. They have the potential to address various traffic-related issues, such as reducing congestion, improving safety, prioritizing emergency vehicles, reducing fuel consumption and emissions, and enhancing safety for pedestrians and cyclists. Furthermore, smart traffic lights can provide real-time traffic management, allowing engineers to make adjustments as needed to optimize traffic flow and reduce congestion. Some of the problems solved by this are given below:

- Emergency vehicle prioritization: Smart traffic lights can also prioritize emergency vehicles, such as ambulances and fire trucks, by changing traffic signal timings to allow these vehicles to pass quickly through intersections.
- Reduced fuel consumption and emissions: By reducing the amount of time vehicles spend idling at red lights, smart traffic lights can also help reduce fuel consumption and emissions, which can have a positive impact on the environment.
- Improved pedestrian and cyclist safety: Smart traffic lights can also improve safety for pedestrians and cyclists by providing dedicated signal timings for these groups, and by using sensors to detect when pedestrians or cyclists are waiting to cross the road.
- Real-time traffic management: Smart traffic lights can also provide real-time traffic management, allowing traffic engineers to monitor traffic flow and make adjustments as needed to optimize traffic flow and reduce congestion.



Overall, the use of smart traffic lights is an advanced and effective solution that utilizes cutting-edge technologies to enhance traffic flow, reduce congestion, and improve safety on the roads.

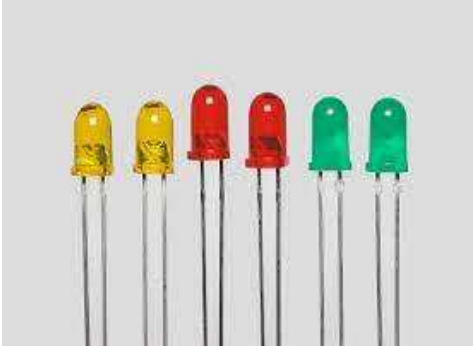


4.2 Brief of Proposed System

The system uses cameras to capture images of the traffic flow at each intersection and analyzes them on the microcomputer. The system then uses the on-board Raspberry Pi computer to run a highly efficient model which analyzes video frames from the camera to determine the number of cars in the frame. It handles input from 4 different camera systems and decides optimal traffic light timings based on the inputs.

The data is analyzed and calculated according to the traffic congestion in each road and traffic light duration for each intersection an algorithm is used to decide which road gets the right of way.. The system then sends commands to the traffic lights via Wi-Fi to adjust their timing accordingly.

4.3 Project Component Specification

Component	Component Feature	Use Case
 Raspberry Pi	Raspberry Pi 4	It is used for all onboard computation and is the brain of the project
 Raspberry Pi Camera	Raspberry Pi Camera 5MP	To capture live videos of vehicles in the traffic junction

 <p>LEDs</p>	<p>Red, Green and Yellow LED</p>	<p>So simulate the traffic signals used in the real world</p>
 <p>Resistors</p>	<p>100ohm Resistors</p>	<p>Used to prevent excess current flow to the LEDs</p>
 <p>Jumper Cable</p>	<p>Connecting wires</p>	<p>To connect all the components</p>

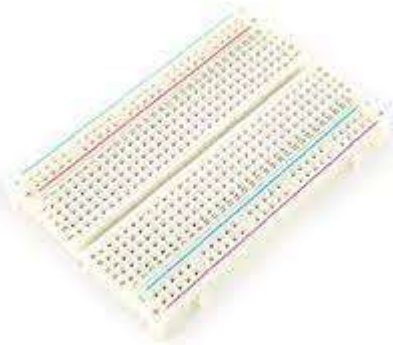
 <p data-bbox="375 638 526 669">Breadboard</p>	<p data-bbox="724 212 873 243">Breadboard</p>	<p data-bbox="1141 212 1459 317">It helps to neatly arrange all components like the LEDs and resistors</p>
---	---	--

Table 4.3 List Of Components Used

Chapter 5: Implementation

5.1 Experimental Setup and Technical Specifications

Table 5.1: Specifications of Software

Sr. No	Software	Version	Use Case
1.	Python	2.7	A powerful programming language which supports development using Machine Learning Libraries.
2.	Jupyter	1.0.0	An application that simplifies python code and output combinations.
3.	Remote Desktop Connection	11.0	Used to remotely connect and control the Raspberry Pi

Table 5.2: Specification of Libraries

1.	OpenCV	3.4.11	Open source computer vision and machine learning software library
2.	Pillow	9.2.0	Used for all the basic image processing functionality
3.	Numpy	1.24.2	It is used to perform a wide variety of mathematical operations on arrays
4.	RPi.GPIO	0.7.1	Used to control the LEDs connected to the Raspberry Pi

Table 5.3: System Specification

Sr. No	Software/Hardware	Version	Use Case
1.	Windows	11	Operating System.
2.	Raspbian OS	5.15	Operating System
3.	Ryzen 5 4600H	3.0 GHz	Processor.
4.	16 GB RAM	DDR4 3200 MHz	Memory that is used for running the applications.
5.	Nvidia GTX 1650	4 GB VRAM	GPU that is used for running the deep learning methodologies.
6.	Raspberry Pi	4	Onboard Computation

5.2 Cost Of Equipment

Table 5.2.1 Equipment Cost

Sr. No	Hardware	Cost (In Rs)
1.	Raspberry Pi 4	6000
2.	Leds + Resistors	200/- each road
3.	Jumper Wires	50

5.3 Dataset Preparation

The dataset called Indian Driving Dataset is used for the training of our model. The dataset is available on Kaggle and is open source. For the training of a cascade classifier we need to split the data into positive and negative images. All images from the dataset are split into two folders named Positive and Negative respectively.

For the folder with the negative images, we iterate through all images and add the path to each file into a text file. Each file path is separated on a different line. This process is automated by a custom script made in Python. For the positive images we require the path to the image, the number of occurrences of cars in the particular image and the bounding box for each object detected in the positive image. OpenCV has an annotation tool that allows us to go through all the images in our folder and it will display each of the images and allows us to draw a bounding box around each car in the image.

After annotating all positive images, OpenCV can automatically create the text file which stores the path of the image, number of cars in the image and bounding box coordinates of each car in the image. All these values are space separated.



Fig 5.3.1: Dataset Of Image Before Annotating



Fig 5.3.2: Dataset Of Image After Annotating

Next we need to create a vector file from our positive data file. We use the create samples program to do this task. This is also done automatically by OpenCV. We pass the width and height of the detection window size that we wish to use. We provide 24 x 24 which means the model will not be able to detect cars in the image whose dimensions are smaller than 24px size. This gives us a positive.vec file.

Finally we can train our cascade model. We pass the vector file, the negative images, the width and height of the detection window and how many positive and negative samples we want to use for training and the number of stages we train for. We use 80% of our data for training and 20% for testing and training for 100 stages. Once the training is complete we get the cars.xml file which is our trained model that is used for vehicle detection.

5.4 Interfacing with Raspberry Pi

All code was written in Python Programming. The model was trained on a Windows 11 laptop with a GTX 1650 graphics card. The trained model is then used to detect the vehicles. The model and python file is then uploaded to the Raspberry Pi. Leds are connected to the GPIO pins in the Raspberry Pi and these are connected to a resistor to avoid the LEDs from a power surge. The respective pins are then coded in the same python script along with all the conditions to select which lane to get a green light. The respective pins are then set to high or low to set the LED to on or off respectively.

The raspberry pi can be connected to a laptop screen to view and interact with the raspbian OS. This can be done using Remote Desktop Connection. Both the laptop and the raspberry pi have to be connected to the same WiFi network and the ip address of the raspberry pi is required. On filling the above details we can login to the raspberry pi.

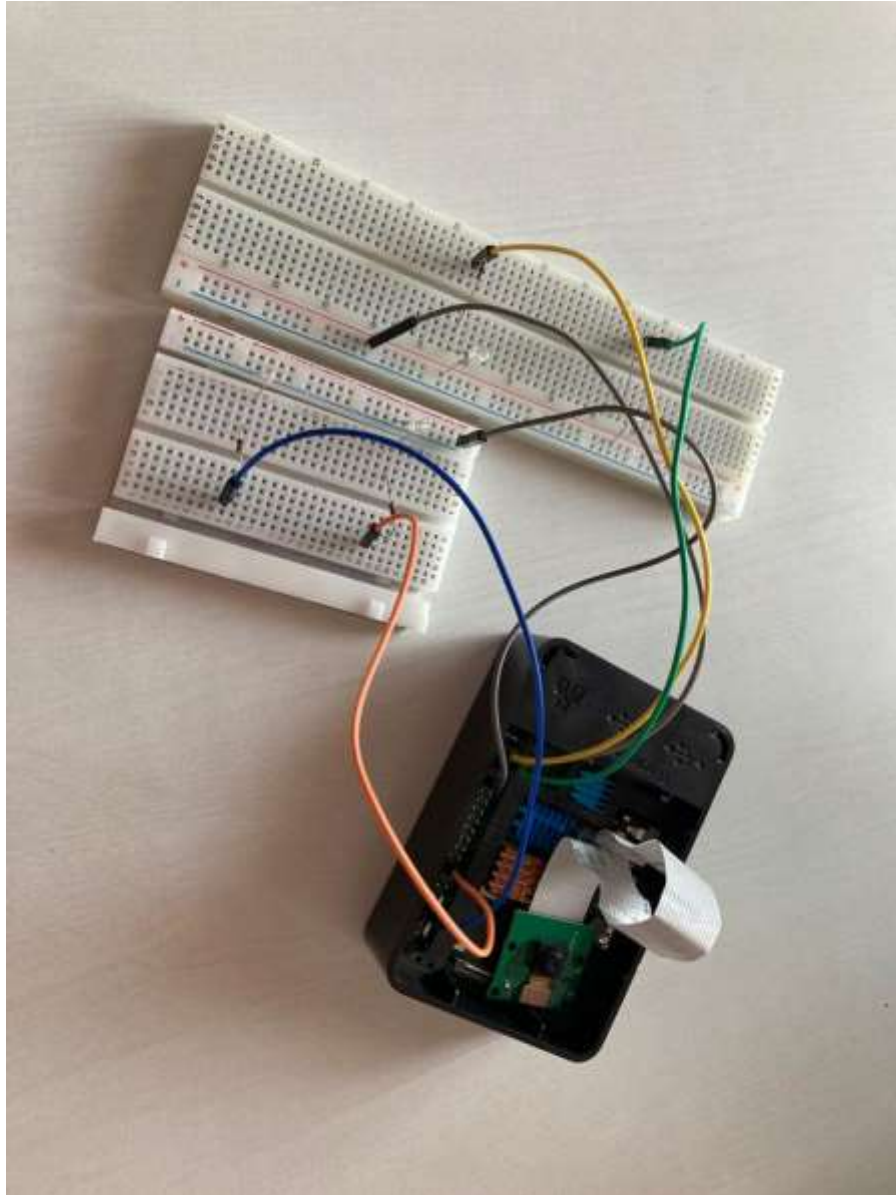


Fig 5.4.1 Interfacing with Raspberry Pi

The raspberry pi draws power from a 5V adaptor. All connections were made using Male-Female Jumper cables and the LEDs and resistors were placed on a breadboard. Our laptop screen is used to display the Raspberry Pi interface. This is done using Remote Desktop Connection. Both the laptop and the Raspberry Pi are connected to the same WiFi. On successfully logging in to our raspberry pi, we can access the files and run our scripts.

5.4.1 Onboard Vehicle Detection:

We have implemented vehicle detection as a function. This function returns the number of vehicles present in the image.

```
def detect(frame, car_cascade):
    image_arr=np.array(frame)
    grey=cv2.cvtColor(image_arr,cv2.COLOR_BGR2GRAY)
    Image.fromarray(grey)
    blur=cv2.GaussianBlur(grey,(5,5),0)
    Image.fromarray(blur)
    erode=cv2.erode(blur,np.ones((3,3)))
    dilated=cv2.dilate(erode,np.ones((3,3)))
    Image.fromarray(dilated)
    kernel=cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(2,2))
    closing=cv2.morphologyEx(dilated,cv2.MORPH_CLOSE,kernel)
    Image.fromarray(closing)
    cars = car_cascade.detectMultiScale(closing, 1.1, 1)
    cnt = 0
    for (x,y,w,h) in cars:
        cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)
        cnt += 1
    return cnt
```

Fig 5.4.2: Vehicle Detection Function

The following are the steps the code follows:

1. The image is converted to an array for easy processing.
2. The image is converted from BGR to Grayscale. OpenCV by default reads all images as BGR instead of RGB.
3. We apply a Gaussian Blur using a 5x5 mask.
4. The image is then eroded to remove the pixels on the object boundary.
5. It is followed by dilation. Erosion followed by dilation can remove small objects from the image and smoothens the border of large objects.
6. Closing is used to fill small holes in the image while preserving the shape and size of large holes and objects in the image.
7. We then pass the trained model and draw bounding boxes around each vehicle that is detected by the model.
8. The counter is incremented by 1 for each object it detects.
9. The final count is returned which gives us the total number of vehicles in the image.

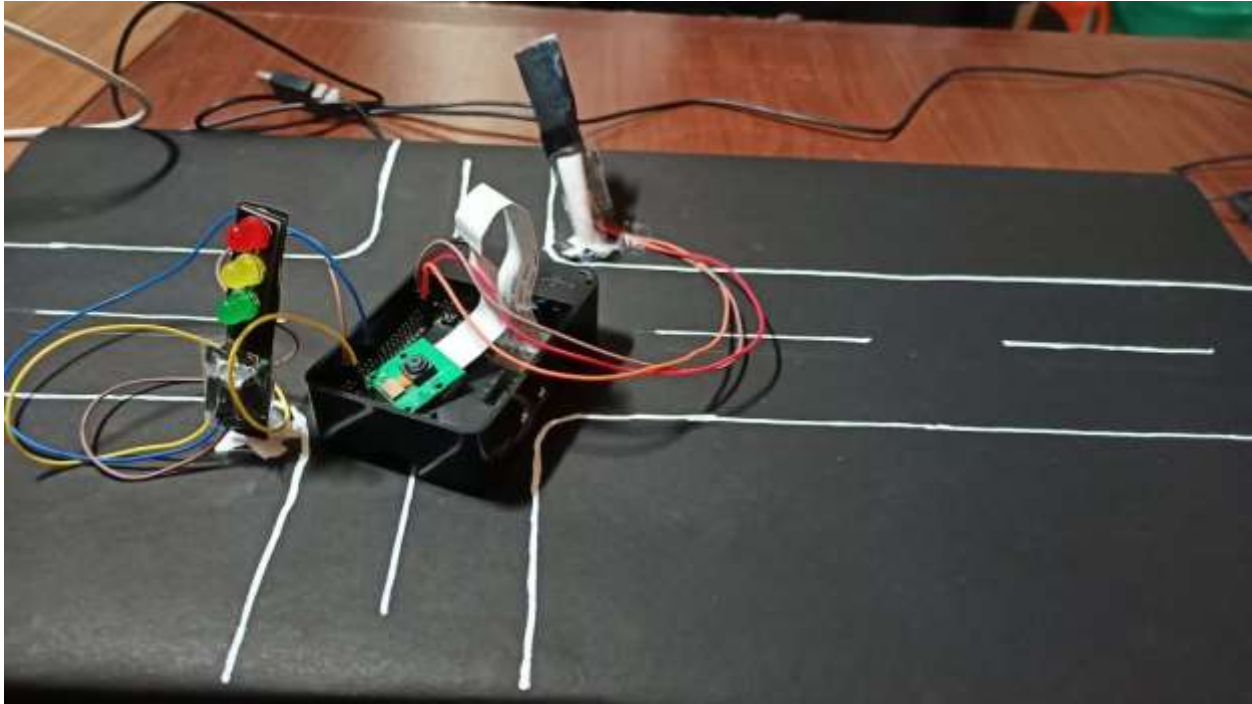


Fig 5.4.3: Traffic junction model

5.4.2 LED Switching Mode:

The model can now be used to detect the number of vehicles on different roads at the traffic junction. Based on the number of vehicles on each side, the traffic light is adjusted accordingly. For the side that has the most traffic, i.e, the most number of vehicles, that side would be provided with green light while the other side remains red. This is done to ease the traffic on the heavily congested side.

A timer is also set to switch the light normally, to give other sides green as well. This is to prevent low traffic roads to remain red for an infinite time if the other sides always have heavy traffic. The LEDs are connected to GPIO pins on the Raspberry Pi. GPIO pins are digital signal pins which may be used as input or output or both. In our case, they work as the output for the LEDs.

Every 5 seconds an image is taken from all the different video sources and the model is run over each image to detect which image has the most number of vehicles. This side is given a green light while the other sides remain red. This 5 seconds image capturing method is followed as the Pi will not be able to handle live video detection from multiple cameras at the same time as it is not as powerful as a full fledged computer. This makes the Pi work faster and more efficiently.

Chapter 6: Results

6.1 Results of Experimental Tests

The Raspberry Pi is able to successfully compute all vehicle detections onboard with an accuracy of 84%. The LEDs change according to the density of traffic on each side of the traffic junction. The side with the most number of vehicles is given a green light while the others remain red. Bounding boxes are drawn around each detected vehicle and the vehicles are tracked every 5 seconds for all roads.

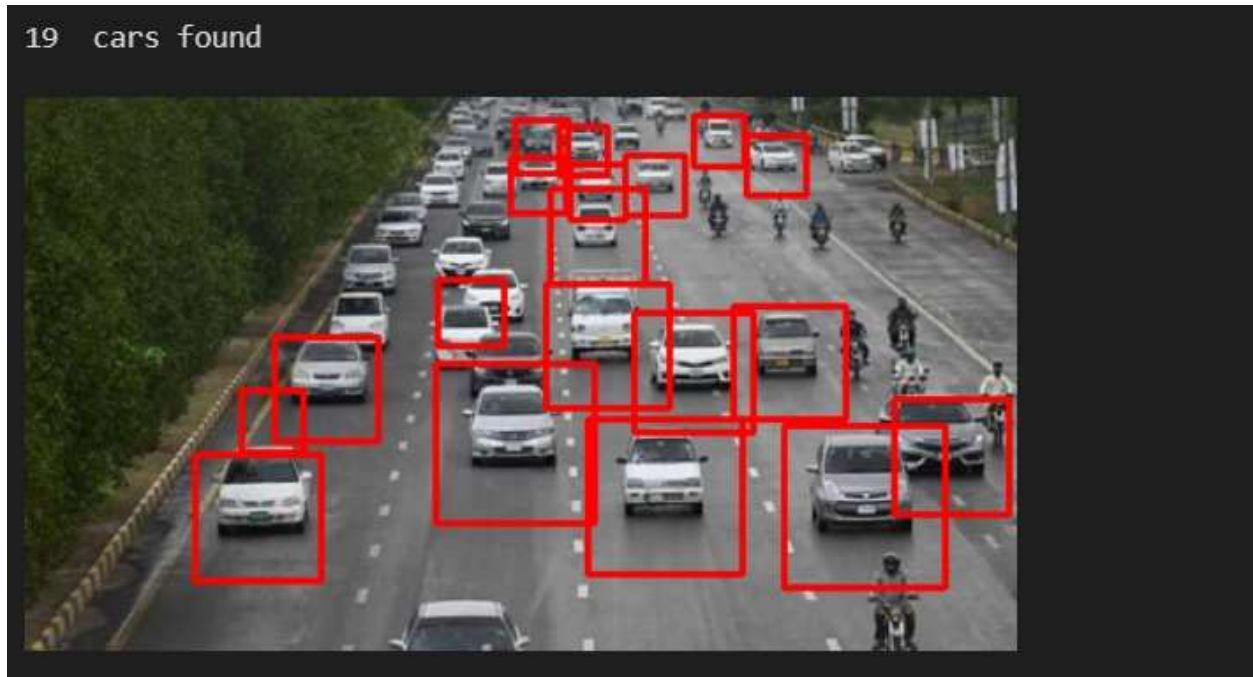


Fig 6.1 Model detecting vehicles in a real-time video

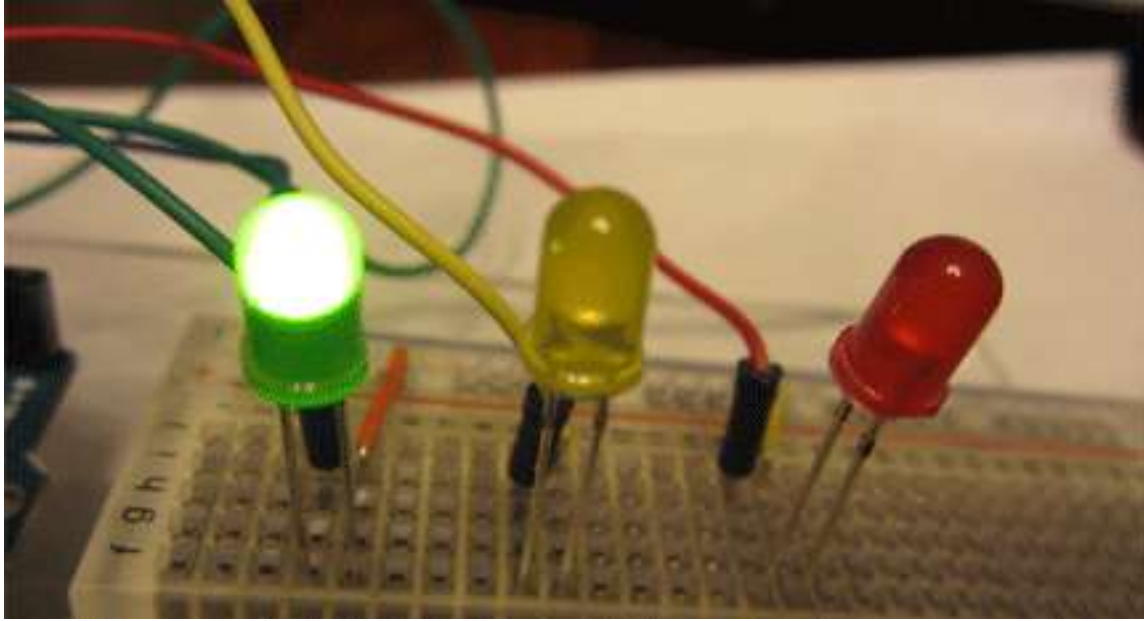


Fig 6.2 Green LED glows for excessive traffic road

As seen from Fig 6.2 based on the number of vehicles on the different roads, the road which has the most density of vehicles would be given a green light. The model will again check after 5 seconds to determine which road to give a green light to. A single road can have a maximum of 30 seconds of continuous green light, after which the model automatically gives green to the other roads. This is to prevent a single road from always having a green light.

Chapter 7: Conclusion and Future Scope

7.1 Conclusion

Smart traffic lights using IoT and deep learning have the potential to change the way we manage traffic. Using real-time data from sensors and cameras, these systems can optimize traffic flow, reduce congestion and improve driver and passenger safety. Deep learning algorithms can identify complex patterns in traffic data to accurately predict traffic speeds, allowing traffic lights to adjust their timing in real time to improve traffic flow.

However, the implementation of such systems requires careful consideration of privacy concerns and security, as well as significant investments in infrastructure and technology. In addition, the effectiveness of these systems may be limited by factors such as measurement accuracy and data transmission reliability.

In conclusion, smart traffic lights using IoT and deep learning are a promising way to solve traffic accidents and improve road safety. However, more research and development is needed to overcome the problems associated with large-scale use of these systems.

7.2 Future Scope

Smart traffic management is a new innovation in the transportation industry to make the city more efficient, safe and sustainable. These traffic lights use a combination of sensors, cameras and machine learning algorithms to adjust installation times based on traffic times. Therefore, lighting can reduce accidents, reduce fuel consumption and reduce carbon emissions.

The future of traffic lights is promising and promises to change the way we manage traffic in cities. Some potential areas where traffic lights can have a major impact are:

- **Junction control:** Traffic lights can keep traffic flowing at intersections by adjusting the signal duration according to sound level and pattern.
This can reduce traffic waiting time at the signal, which can improve traffic flow and reduce congestion.
- **Emergency Response:** In an emergency, such as a natural disaster, injury, or emergency, lighting can provide emergency personnel with an open space to their destination by synchronizing signals and clearing the way for emergency vehicles. This saves lives and improves reaction time.
- **Pedestrian Safety:** Traffic lights can also improve pedestrian safety by using sensors and cameras to detect and timely adjust pedestrians at the crosswalk. This reduces the risk of accidents and improves the overall passenger experience.
- **Public transport:** Intelligent traffic lights can also be used to improve public transport performance before buses, trains and other public transport. By synchronizing signals and generating high traffic, traffic lights can reduce travel time and encourage more people to use public transport.

- **Environmental Sustainability:** Lighting can play an important role in reducing carbon emissions and promoting environmental sustainability. Smart lights can reduce fuel consumption and pollution by optimizing traffic, reducing congestion and minimizing idling.
- **Driverless cars:** As driverless cars become more common, traffic lights can be used to communicate with these vehicles and provide real-time traffic information. This can help driverless cars travel more efficiently, reduce the risk of accidents and improve overall traffic flow.

Chapter 8: Single Unique Factor

The unique factor in our project is that all computation is done on board in the Raspberry Pi and thus, the system does not depend on any cloud services or internet connection. All traffic lights can be controlled by the raspberry pi. The model has been optimized to work smoothly with Raspberry to detect vehicles in multiple roads.

Chapter 9: Site Visit

To test our vehicle detection system, we visited the busy junctions of Kelambakkam and Vandalur and took videos of live traffic present there. Videos were taken at different times of the day - Morning, Evening and Night and were used for simulation in our traffic management model. These videos are real world examples of where our model would be used. In a country like India, with a huge number of vehicles which include not only cars and bikes, but also autorickshaws, trucks, bullock carts, etc, this was a perfect example to test our model.

Chapter 10: Sample Codes

Github Link: [manvix404/vehicle-detection: Detecting and counting vehicles using opencv cascade classifier \(github.com\)](https://github.com/manvix404/vehicle-detection: Detecting and counting vehicles using opencv cascade classifier (github.com))

Video Explanation: [TARP Video Explanation](#)

Chapter 11: References

- [1] S. K. Kumaran et al., "Computer vision-guided intelligent traffic signaling for isolated intersections," *Expert Systems with Applications*, vol. 140, p. 112888, 2020.
- [2] Vladimir Gorodokin et al., "Method of Non-Stop Passage of Signal-Controlled Intersections Using Dynamic Signs and Computer Vision," *Transportation Research Procedia*, vol. 50, pp. 174-181, 2020.
- [3] B. D. Sri Sai, N. Ramisetty, P. S. Babu and N. Srinivas Naik, "Traffic Management using Computer Vision and SUMO," 2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 2021, pp. 1-5, doi: 10.1109/IEEM52672.2021.9484044.
- [4] Lanke, N., & Koul, S. (2013). Smart Traffic Management System. *International Journal of Computer Applications*, 75(7), 19-22.
- [5] Anandhalli, M., & Tanuja, K. (2014). Geometric invariant features for the detection and analysis of vehicle. *International Journal of Computer Science and Mobile Computing*, 3(4), 1-6.
- [6] P. Yadav and R. K. Yadav, "SMART TRAFFIC MONITORING AND CONTROLLING," 2015 International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, 2015, pp. 1-4, doi: 10.1109/ICCUBEA.2015.108.
- [7] M. Wiering, J. van Veenen, J. Vreeken and A. Koopman, "Intelligent Traffic Light Control," Technical Report UU-CS-2004-029, Institute of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands, July 9, 2004.
- [8] B. De Schutter and B. De Moor, "Optimal traffic light control for a single intersection," *Proceedings of the 1999 American Control Conference*, San Diego, California, pp. 2195–2199, June 1999.
- [9] Y. Ma, X. Liu, G. Zhang, Y. Wang and J. Wang, "IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control," 2018 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), London, United Kingdom, 2018, pp. 316-324, doi: 10.1145/3219819.3220096.
- [10] T. Osman, S. S. Psyche, J. S. Ferdous and H. U. Zaman, "Intelligent traffic management system for cross section of roads using computer vision," 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox's Bazar, Bangladesh, 2017, pp. 1-6, doi: 10.1109/ECACE.2017.8302024.
- [11] Khan, S., Nazir, S., García-Magariño, I., & Hussain, A. (2021). Deep learning-based urban big data fusion in smart cities: Towards traffic monitoring and flow-preserving fusion. *Computers & Electrical Engineering*, 89, 106906.

- [12] Zinchenko, V., Kondratenko, G., Sidenko, I., & Kondratenko, Y. (2020, August). Computer vision in control and optimization of road traffic. In *2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP)* (pp. 249-254). IEEE.
- [13] Thakare, V. S., Jadhav, S. R., Sayyed, S. G., & Pawar, P. V. (2013). Design of smart traffic light controller using embedded system. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 10(1), 30-33.
- [14] Ciampi, L., Santiago, C., Costeira, J. P., Gennaro, C., & Amato, G. (2020). Unsupervised vehicle counting via multiple camera domain adaptation. *arXiv preprint arXiv:2004.09251*.
- [15] Dai, Z., Song, H., Wang, X., Fang, Y., Yun, X., Zhang, Z., & Li, H. (2019). Video-based vehicle counting framework. *IEEE Access*, 7, 64460-64470.
- [16] Miz, V., & Hahanov, V. (2014, September). Smart traffic light in terms of the cognitive road traffic management system (CTMS) based on the Internet of Things. In *Proceedings of IEEE east-west design & test symposium (EWDTS 2014)* (pp. 1-5). IEEE.
- [17] Alaidi, A., Aljazaery, I., Alrikabi, H., Mahmood, I., & Abed, F. (2020). Design and implementation of a smart traffic light management system controlled wirelessly by arduino.
- [18] Huang, J., Ji, Z., Xiao, S., Jia, C., Jia, Y., & Wang, X. (2022, March). Multi-agent vehicle formation control based on mpc and particle swarm optimization algorithm. In *2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC)* (Vol. 6, pp. 288-292). IEEE.
- [19] Nellore, K., & Hancke, G. P. (2016). A survey on urban traffic management system using wireless sensor networks. *Sensors*, 16(2), 157.
- [20] Nogami, J., Nakasuka, S., & Tanabe, T. (1996). Real-time decision support for air traffic management, utilizing machine learning. *Control Engineering Practice*, 4(8), 1129-1141.