# DEVELOPMENT OF DENSITY BASED TRAFFIC LIGHT CONTROL SYSTEM USING RASPBERRY PI

**Shireen fatima [1], Dr Shanila mahreen [2]**

[1] Shireen Fatima (M tech Embedded system), student Nawab Shah Alam Khancollege of engineering and technology, Hyderabad, Telangana India -500024

[2] Dr Shanila mahreen , Assistant professor HOD, Electronics and communication engineering department, Nawab Shah Alam Khan college of engineering and technology, Hyderabad, Telangana, India -500024

## ABSTRACT

Nowadays congestion in traffic is a serious issue. The traffic congestion can also be caused by large red light delays etc. The delay of respective light is hard coded in the traffic light and it is not dependent on traffic. In this paper we studied the optimization of traffic light controller in a city using microcontroller. The system tries to reduce possibilities of traffic jams, caused by traffic lights, to an extent. The system is based on raspberry-pi. Overall, traffic density is a complex problem that requires innovative solutions to manage. By implementing strategies such as density-based traffic control systems, cities can improve traffic flow, reduce congestion, and create a safer and more sustainable environment for everyone. This system uses a Raspberry Pi camera to capture real-time images of the road and processes the data to determine the density of vehicles.Theimagesareanalyzedusinganalgorith mthatdetectsthenumberofvehicleson the road. The system adjusts the traffic signal timings based on the vehicle density, which optimizes the traffic flow and reduces congestion. This approach offers several advantages over traditional traffic management systems, including low cost, easy implementation, and integration with other systems. The system also enhances the safety of motorists and pedestrians by reducing the possibility of accidents due to traffic congestion. This solution can be implemented in urban areas to improve traffic flow, reduce emissions, and enhance the overall efficiency of the transportation system. The density-based traffic control system using Pi Cam is a promising technology that can significantly improve traffic management, making it an essential tool for modern transportation infrastructure.

## 1. INTRODUCTION

The primary issue in today's society that needs in an old automatic traffic controlling a traffic light uses timer for every phase. Using electronic sensors is another way in order to detect vehicles, and produce signal that to this method the time is being wasted by a green light on an empty road. Traffic congestion also occurred while using the electronic sensors for controlling the traffic. All these drawbacks are supposed to be eliminated by using image processing. We propose a system for controlling the traffic light by image processing. The vehicles are detected by the system through images instead of using electronic sensors embedded in the pavement. A camera will be placed alongside the traffic light. It will capture image sequences. Image processing is a better technique to control the state change of the traffic light. It shows that it can decrease the traffic congestion and avoids the time being wasted by a green light on an empty road. It is also more reliable in estimating vehicle presence because it uses actual traffic images. It visualizes the practicality, so it functions much better than those systems that rely on the detection of the vehicles' metal content. Image Processing is a technique to enhance raw images received from cameras/sensors placed on space probes, aircrafts and satellites or pictures taken in normal day-today life for various applications. An Image is rectangular graphical object. Image processing involves issues related to image representation, compression techniques and various complex operations, which can be carried out on the image data. The operations that come under image processing are image enhancement operations. One of the many issues we deal with in modern life is traffic congestion, which gets worse every day. Determining traffic flow might be a key factor in learning more about them. A true record of traffic volume patterns is provided by this data, which is utilized to build censorious flow time periods such as the impact of large vehicles and specific parts on vehicular traffic flow. This data is useful for processing traffic more effectively in terms of the cycle times of traffic lights. There are various ways to gauge the amount of traffic flow by counting the number of cars that pass through an area at a given moment. Today, camera-based systems are superior options for tracking the data of the cars. This study focuses on a unique firmware-based vehicle detecting method. Using an existing identifier for each vehicle, this method locates the automobiles in the given image. Later, it groups each vehicle according to its category and counts each one separately. More precision, greater dependability and fewer errors are produced as a result of the developed approach's implementation in a firmware platform. On a daily basis, traffic lights are crucial for regulating and controlling traffic. The density of the roads is calculated using Python, and the microcontroller modifies the length of the green light for each road in accordance with the results of image processing.

## 2. LITERATURE SURVEY

In [1] a comparison of various deep-learning image identification methods was done by the authors.

The study's goal was to assess and contrast the efficacy of several deep learning algorithms for image detection tasks. The researchers concentrated on well-known deep learning architectures such as Retina Net, SSD, YOLOv3, and Faster R-CNN.

For the purpose of developing and testing the algorithms, the authors gathered and produced datasets. They made use of widely utilized benchmark datasets, such as COCO and VOC, to guarantee the accuracy of their findings.

Precision, recall, and mean average precision (map), among other performance criteria, were used to evaluate the algorithms. The researchers also took into account things like training duration and computing efficiency.

In [2] the author describes a real-time traffic monitoring system that utilizes an intelligent transportation system and computer vision techniques to address challenges such as vehicle counting, speed estimation, accident detection, and traffic surveillance. Deep learning algorithms, specifically YOLOv3 and YOLOv4, are employed for vehicle detection and classification. The system combines a virtual detection zone, Gaussian mixture model (GMM), and YOLO to enhance vehicle counting and classification efficiency. Additionally, the distance and time traveled by vehicles are used to estimate their speed. The study presents a flowchart outlining the vehicle counting and classification process. The materials and methods section discusses data set preparation, vehicle counting using GMM and virtual detection zone, and vehicle detection and classification using YOLO. The YOLOv3 and YOLOv4 architectures are described, along with their loss functions. The text also mentions speed estimation and provides a diagram illustrating the calculation process. Overall, the system aims to achieve real-time computation, accurate vehicle detection, and efficient traffic monitoring.

In [3] the author describes a proposed research work that aims to analyze traffic density using Python Opens and YOLOv3 (You Only Look Once) algorithm. The research work involves recording real-time videos from Sony HD IP cameras in a designated area and using image frames from the video sequence to detect moving vehicles.

The author introduces the use of the YOLOv3 algorithm for vehicle detection and counting without the need for additional hardware. It mentions that YOLOv3 provides better results for rotating and small objects. The section also discusses the experimental setup, including inputs, outputs, anchor boxes, and the encoding process in YOLOv3. It highlights the computation of class scores and explains how the algorithm assigns classes and scores to each box based on the probability of object presence and class probability.

Overall, the research work aims to develop an efficient and accurate system for real-time vehicle detection, classification, and counting using computer vision techniques and deep learning algorithms.

In [5] the author discusses the achievements achieved using convolution neural networks (CNNs) since 2012 in the field of computer vision's object identification models. Before deep learning became popular, object recognition was carried out using manually created machine learning features like SHIFT and HOG. On the PASCAL VOC Challenge dataset, these models performed only moderately, with the top model achieving about 35% mean average precision (map).

Object detection in computer vision involves identifying the class and location of objects within an image. Before deep learning became popular, handcrafted features were used for object detection, but they had limited performance. The introduction of convolution neural networks (CNNs) revolutionized object detection. Models like R-CNN, SPP-net, Fast R-CNN, and Faster R-CNN improved the accuracy and speed of object detection. Additionally, the use of techniques like region proposal networks (RPN) and feature pyramid networks (FPN) further enhanced object detection performance. These advancements in CNN-based object detection models have significantly improved the accuracy and efficiency of object detection tasks.

## 3. EXISTING SYSTEM

In the existing system, Traffic Density Monitoring uses Raspberry Pi and Opens. The traffic lights used in India are basically pre-timed wherein the time of each lane to have a green signal is fixed. In a four-lane traffic signal, one lane is given a green signal at a time. Thus, the traffic light allows the vehicles of all lanes to pass in a sequence. So, the traffic can advance in either a straight direction or turn by 90 degrees. So even if the traffic density in a particular lane is the least, it has to wait unnecessarily for a long time and when it gets the green signal it unnecessarily makes other lanes wait for even longer durations.

## 4. PROPOSED SYSTEM

## 5.

Raspberry Pi turns on when the kit does. The Pi cam counts the number of vehicles on the road while taking pictures along it. Regardless of the amount of traffic on the roadways, the system uses audio to detect the arrival of any emergency vehicles, such as fire engines or ambulances, and signals green.

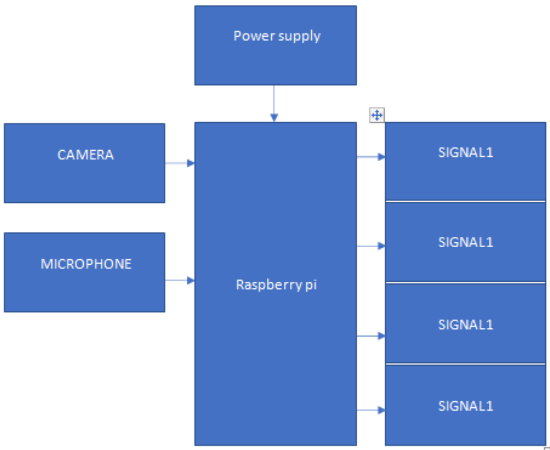


Fig.1. Proposed block diagram

WORKING MODEL:

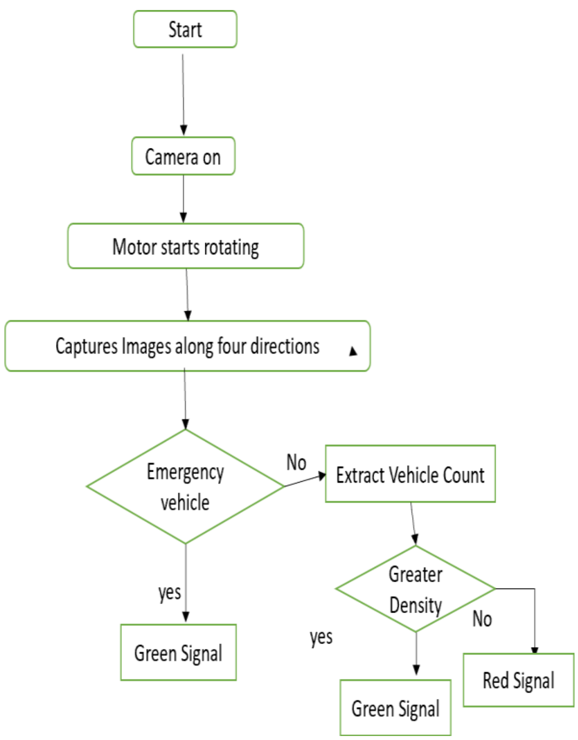


Fig.2. Proposed Flow diagram

Once every picture has been taken, the number of vehicles in each picture is computed and assessed. First, the system is configured to operate in either a clockwise or counter-clockwise orientation when it first switches on. Next, a new traffic loop is set up based on the vehicle count in descending order—that is, the side with the highest count has priority. This order is in effect for a specific amount of time, after which new photographs will be taken and priorities will be allocated based on the number of vehicles.

The fundamental building block of convolution neural networks (CNNs) is the convolution layer, which extracts characteristics from input data—typically images or videos—by using this technique.

In order to create a set of output feature maps, the layer performs convolution operations on the input data using a cluster of learnable filters, sometimes referred to as weights or kernels.

Sliding the filters over the input data and calculating the dot products between the filter values and the associated input values at each place comprise the convolution operations. To add non-linearity and strengthen the network's representational capability, the resulting dot products are summed and then passed through a non-linear activation function that is comparable to Rely.

A convolution layer's size and number of filters are hyper parameters that are usually selected depending on the job at hand and the properties of the input data. Typically, the filters are tiny—3 by 3 or 5 by 5—in order to catch local patterns and structures in the incoming data. The number of output feature maps that the layer generates, each of which captures a distinct collection of features, is determined by the number of filters.

Convolution layers are frequently stacked one on top of the other in a CNN, with the output of one layer feeding into the next. Usually, the layers are organized in a hierarchical manner, with the lower-level features—such as corners and edges—being captured by the early layers and the more abstract and complicated aspects by the later ones.

Back propagation was used to alter the detailed weights of the corresponding filters in the convolution layer during training in order to reduce the error between the true label and the

anticipated output. This procedure entails updating the musing gradient for descent or a variation and determining the effective gradient of the loss function with regard to the weights.
Convolution layers, in general, are a potent tool for extracting feature representations from high-dimensional input data and have been essential in the success of CNNs in a variety of fields, including segmentation, object identification, and picture classification.
Deep learning is the algorithm that we are utilizing. Additionally, there are several kinds of deep learning algorithms; the CNN, or convolution neural network, method is the one that works best for our purpose. We employ one of the two convolution varieties, support vector machines, or SVMs. The project uses convolution neural networks because it is image-based.

Deep Learning

Deep learning is a kind of machine learning that processes and analyzes data using multiple-layered artificial neural networks. It is a potent method of machine learning that has demonstrated outstanding results in a variety of applications, such as robotics, natural language processing, picture and audio recognition, and more.
Deep learning networks are made up of many layers of connected neurons and are patterned like the human brain. A particular function, such feature extraction or classification, is carried out by each layer in the network. Large datasets are used to train the layers so they can find patterns and learn from the data, ultimately increasing the precision of their classifications or predictions.
Deep learning's capacity to automatically learn data representations from raw inputs without the need for manual feature extraction is one of its main features. This makes it ideally suited for applications like speech or picture recognition when the data is complicated or high-dimensional.
Numerous techniques, such as stochastic gradient descent and back propagation, can be used to train deep learning networks. By using these techniques, the network is able to gradually increase the accuracy of the model by adjusting its weights and biases in response to inaccuracies in the predictions.
All things considered, deep learning is an effective method of machine learning that has transformed a number of industries, including robotics, natural language processing, and computer vision. Because of its versatility, scalability, and capacity to learn intricate representations from unprocessed data, it is a preferred option for scientists and programmers attempting to tackle difficult real-world issues.

Neural Convolution Networks

Convolution neural networks (CNNs) are a class of artificial neural networks that find extensive application in natural language processing, image and video recognition, and numerous other fields. CNNs are especially made to work with high-dimensional inputs like photographs, and its structure is inspired by how the visual cortex functions in both humans and animals.
A CNN's core consists of several convolution layers that execute convolutions on the input data to extract features pertinent to the current task. Usually, the convolutions are applied by sliding tiny filters, or kernels, over the input data, calculating the dot products at each location, and creating a fresh feature map.
Additional layers are frequently added to CNNs, such as fully connected layers, which carry out the final classification or regression, and pooling layers, which down sample the feature maps to minimize their size and computational complexity. Usually, the layers are organized in a hierarchical manner, with the later layers capturing more abstract and complicated properties and the earlier layers catching low-level features like edges and corners.
To reduce the error between the predicted output and its true label, a CNN must be trained by feeding it a large dataset of labeled samples

and modifying the weights and biases of each neuron.

CNN's convolution layer

An essential component of CNNs, the convolution layer is in charge of extracting features from the input data, which is usually in the form of pictures or videos. In order to generate a set of output feature maps, the layer convolution ally processes the input data using a set of learnable filters, sometimes referred to as kernels or weights.

Sliding the filters over the input data and computing the dot products between the filter values and the corresponding input values at each place constitute the convolution operation. The dot products obtained are added together and subsequently run through a non-linear activation function, such Rely, in order to add non-linearity and improve the representational power of the network.

Hyper parameters in a convolution layer include size and number of filters, which are usually selected according to the job and the properties of the input data. Typically, the filters are tiny—3 by 3 or 5 by 5—in order to catch local patterns and structures in the incoming data. The number of output feature maps that the layer generates, each of which captures a distinct collection of features, is determined by the number of filters.

In a CNN, convolution layers are frequently stacked one on top of the other, with the output of one layer acting as the input for the layer after it. Typically, the layers are organized hierarchically, with the early layers collecting low-level properties like corners and edges, and

Subsequent layers are able to capture more intricate and abstract elements.

Back propagation is used during training to modify the filter weights in the convolution layer in order to reduce the error between the true label and the anticipated output. This procedure entails updating the weights using gradient descent or a variant after calculating the gradient of the loss function with respect to the weights.

Convolution layers, in general, are a potent tool for extracting feature representations from high-dimensional input data and have been essential in the success of CNNs in a variety of fields, including segmentation, object identification, and picture classification.

Yolo in Deep Learning

The You Only Look Once (YOLO) algorithm is classified as a deep learning method and more precisely as a deep convolution neural network (CNN). A well-liked object recognition algorithm called YOLO locates and finds several items in an image in real time. In order to operate, each image is divided into a grid of cells, and each cell's bounding boxes and objectless scores are predicted.

To produce the final predictions, YOLO employs a deep neural network to extract information from the image, which are further processed through a number of convolution and fully connected layers. Back propagation is used to train the network using a sizable dataset of annotated images, modifying the weights and biases of the network to reduce prediction errors. In general, Yolo is an illustration of how deep learning may be applied to resolve challenging issues. For example, object detection, and it has grown to be a popular technique in computer vision applications.

## 6. RESULTS

Firstly finding the vehicle count along all sides new loop is aligned according to the descending order. By analyzing real-time traffic density data captured by the Pi Cam, the system can adjust traffic signals and synchronize them to match the traffic flow, thereby reducing congestion and delays at intersections. This can result in a significant reduction in travel time, fuel consumption, and vehicle emissions, making the system environmentally friendly and cost effective. In addition, this system can improve traffic safety by minimizing the risk of accidents caused by drivers who are frustrated by congestion and take risky maneuvers to avoid delays.

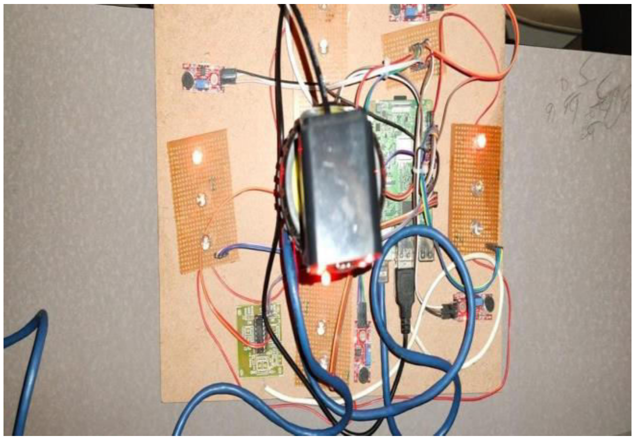

**Figure5.1:** Picture of prototype

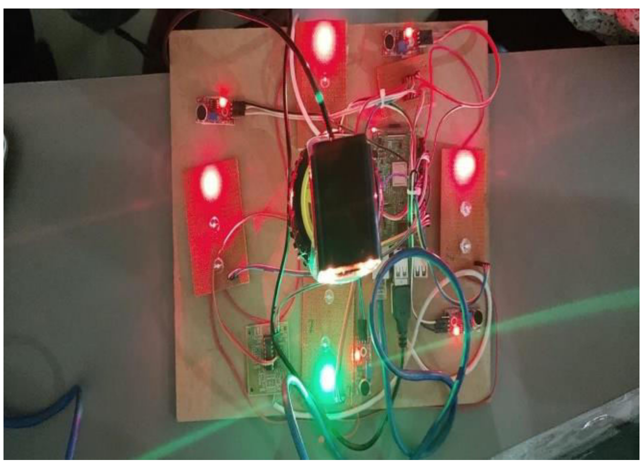

**Figure5.2:** Loop based signal alignment

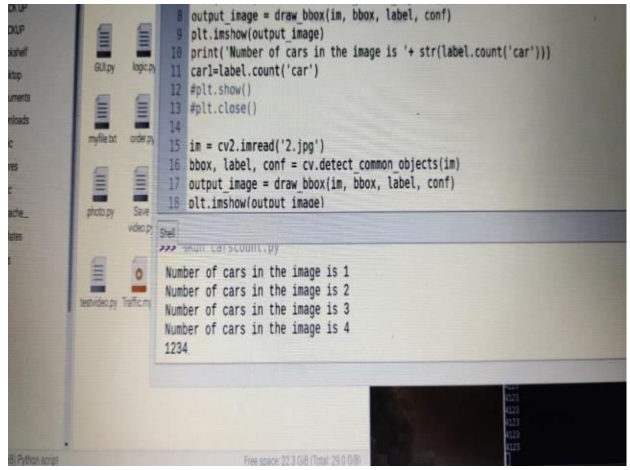


**Figure5.3:** New priority signal alignment

## 7. CONCLUSION

The design we developed is a promising solution to the problem of traffic congestion on roads. By using a combination of hardware and software components, the system is able to accurately detect and monitor the density of traffic on roads and adjust the traffic signal timings accordingly. This can help to reduce the waiting time at traffic signals and improve the overall flow of traffic on roads. The system makes use of Opens and YOLO algorithms for object detection, allowing it to accurately detect vehicles and estimate their speed and direction of motion. It also utilizes a RaspberryPi3 B and ZebCrystal Clear Web Cam for image processing and data analysis. One of the main advantages of the system is its low cost and ease of implementation. The use of off-the-shelf components, such as the Raspberry Pi and the web camera, make it easy to build and deploy the system in real- world scenarios. Additionally, the system can be easily customized to meet the specific needs of different traffic conditions and can be extended with additional features and sensors. Overall, the density-based traffic control system using Pi Cam has the potential to significantly improve traffic flow and reduce congestion on roads. Its cost- effectiveness and scalability make it an attractive solution for governments and municipalities seeking to improve the efficiency of their traffic management systems.

## REFERENCES


[1] W. S. Associate, "Transportation and Economy Report," MDOT State LongRange Transportation Plan, Karachi, 2007.

[2] K. S. D. M. R. B. Patan Rizwan, "Real-Time Smart Traffic Management System for Smart Cities by Using Internet of Things and Big Data," in International Conference on Emerging Technological Trends [ICETT], Kollam, 2016.

[3] C.S.D.L.T.Authority,"AnnualVehicleStatistics,"AnnualVehicleStatistics2015, Karachi, 2015.

[4] M. P. a. B. B. Sivasankar, "IoT Based Traffic Monitoring using Raspberry Pi," InternationJournalofResearchinEngineering,ScienceandTechnology(IJRESTs),vol. 1, no. 7, pp. 2454-664x, 2016.

[5] A. Lone, "Karachi's crime malaise," Tribune.com.pk, 18 August 2011. [Online]. Available:https://blogs.tribune.com.pk/story/7540/karachiscrime-malaise/.[Accessed20 September 2017].

[6] S. Sharief, "Road Accidents in Pakistan Reach Alarming High: Who's Responsible," Pakwheels.com, 2016. [Online]. Available: https://www.pakwheels.com/blog/alarming- increase-of-road-accidentsin-pakistan/. [Accessed 2017].

[7] B. k. Khan, "Traffic jams in Karachi result in losses worth of millions each day," PakWheels.com, 2013. [Online]. Available: https://www.pakwheels.com/blog/traffic-jams-karachi-result-lossesworth-millions-day/. [Accessed 2017].

[8] P. K. K. P. S. T. Prashant Jadhav, "Smart Traffic Control System Using Image Processing,"InternationalResearchJournal ofEngineeringandTechnology(IRJET),vol. 3, no. 3, pp. 2395-0056, 2016.

[9] E.Leigh,"AdvancingintegratedandsustainabletransportfortheCambridgeregion,"Smarter Cambridge Transport, 2017. [Online]. Available: http://www.smartertransport.uk/. [Accessed 2017].

[10] T.C. N.K.KartikeyaJain, "Smart vehicleidentification system using OCR," in 3rd International Conference on Computational Intelligence & Communication Technology (IEEE-CICT 2017), Ghaziabad, India, 2017.

[11] T. e. a. Osman, "Intelligent traffic management system for a cross section of roads using computer vision," in Computing and Communication Workshop and Conference (CCWC),2017IEEE7thAnnual,2017.

[12] [12]D.C.J.E.J.Khac-HoaiNamBui,"Real-Time Traffic Flow Management Based on Inter-Object Communication: aCaseStudy at Intersection," in 7th International Conference on Ambient Intelligence (ISAmI'2016), 2016.