iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 Issue: VII Month of publication: July 2020

DOI: https://doi.org/10.22214/ijraset.2020.30438

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VII July 2020- Available at www.ijraset.com

Smart Traffic Congestion Control System

Pinky Ann Jose[1], Lijo Joseph[2], Mathews Mittathani[3], Jayalakshmi[4], Angel Viji K S[5]
[1, 2, 3, 4, 5]*Department of Computer Science, College of Engineering Kidangoor*

Abstract: India has the second most population in the world and fast growing economy. And there are terrible road congestion problems in many cities. Nowadays signal timings are fixed and they are independent of traffic density. So we require a smart intelligent traffic control system. This paper uses live object detection from a camera to count the number of cars on each road. This system uses Raspberry Pi 3, a camera module, and various electronic hardware for the light itself. Using OpenCV on the Raspberry Pi 3, the information gathered will be run through code that controls the LEDs via the GPIO. Depending on the number of cars, the traffic light will change, letting cars through in the most optimal order. This would eliminate situations when many cars are stopped while there are no cars on the intersecting road.

Keywords: OpenCV, camera module, Raspberry pi, object detection, count comparison

I. INTRODUCTION

In today’s world, traffic lights are essential for a safe road. However, many times, traffic lights can be annoying in situations where someone is approaching the light just as it is turning red. This wastes time, especially if the light is preventing a single vehicle from getting through the intersection when there is nobody else on the road. This system implements a smart traffic light that uses live object detection from a camera to count the number of cars on each road. Depending on these numbers, the traffic light will change, letting cars through in the most optimal order. In this case, the lane with the most cars would be let through so that the lane with fewer cars would be idling, reducing air pollution. This would eliminate situations when many cars are stopped while there are no cars on the intersecting road. Not only does this save time for everybody, but it also saves the environment. Ultimately, this traffic light system could be implemented in cities, suburbs, or even rural areas to be more efficient for people would reduce air pollution.

II. OBJECTIVES AND SCOPES OF WORK

Traffic congestion is one of the basic problems in major urbanized areas. The main problems are the rapid rise in the number of vehicles, lack of road infrastructure and problems in traffic signals. In this scenario, we use a smart traffic light that uses live object detection from a camera to count the number of cars on each road. It keeps a track of the vehicles in each road and accordingly adjusts the time for signals. Depending on number of cars, the traffic light will change, letting cars through in the most optimal order. The project aims in reducing traffic congestion and unwanted waiting for the green signal. Not only does this save time for everybody, but it also saves the environment by reducing pollution.

III. PROPOSED METHOD

In the proposed system, the system takes traffic density as input from cameras, which is abstracted from a Digital Image Processing Technique, which gives output as signals management. Depending on the amount of density, the traffic light will change allowing vehicles through in the most optimal order. The mechanism has the following features: A camera module, Object recognition system, Raspberry Pi and User class and characteristics. Traffic lights can create problems in situations where someone is approaching the light just as it is turning red. This obviously wastes the time. In this scenario we have a smart traffic light that uses live object detection from a camera to count the number of vehicles on each road. In this case, the lane with the most vehicles would be let through so that the lane with fewer vehicles would be resting.

A. Architecture

Here, the camera captures the videos of vehicles. As the first step, a video pre-processing is made on the captured video to provide dramatic quality improvements in the subjective quality of the video. Then the frames are extracted and a HSV (hue, saturation, value) conversion is done. This is done because the RGB components of an object’s color in a digital image are co-related with the amount of light hitting the object, so the object discrimination is difficult. So HSV conversion is done. As the next step, Image Acquisition is done. Among these mainly two processes are done and they are ‘dilation’ and ‘erosion’ to smoothen the image by adding and removing pixels on an object’s boundaries. Followed by this count detection is done. This step detects the vehicles on each lane and as the final step a comparison on this number of vehicles on each lane is made. Depending on these numbers, the traffic light will change, letting cars through in an optimal order.

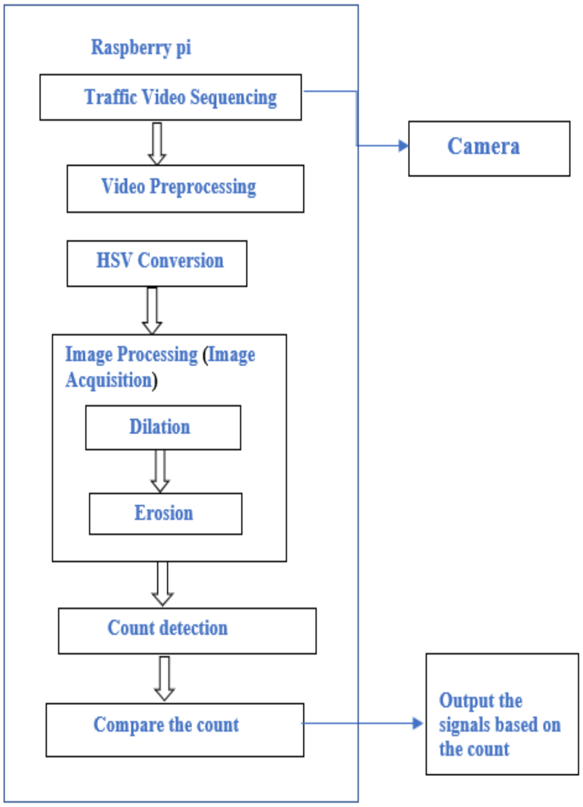


Fig .1 Architecture of proposed system

B. Modules of the system

We have mainly three modules. They are:

1) Object Detection: Here we convert the video input into HSV (hue, saturation, value). It is a type of color plane representation. Then we perform operations like blur, dilate, erode and again dilate. Threshold converts each pixel into black, white or unchanged based on whether the original color value is within the threshold range. Followed by object detection, a circle is drawn through the vehicles' images.

2) Count detection: By using the radius of the circle, vehicles on the horizontal and vertical lanes are calculated.

3) Count Comparison: Vehicles in the horizontal lane and vertical lanes are compared. Signals are given to lanes, either vertical or horizontal with higher number of vehicles and block the other lane.

C. Design

[1] First installing the required OpenCV files, set up the raspberry pi using the required python code. After assembling test out the traffic light with a breadboard. Each opposing set of LEDs share an anode, and all of them share a common cathode (ground). There should be a total of 7 input wires: 1 for each pair of LEDS (6) + 1 ground wire. Solder and assemble the traffic lights. We have to solder the female header pins to about 5 feet of wire. They are the sides that these wires will snake through the PVC pipes later. Make sure to be able to distinguish the different sets of lights (2 x 3 colors and 1 ground). Building the environment make a 2 feet square wood pallet like this. Scrap wood is fine as it will be covered up. Drill a hole that just fits your adapter. Drill screws through the sides of the pallet to secure the PVC pipe into place. Cut the black foam board to match the wood pallet underneath. Drill a hole that fits around the PVC pipe. Repeat on the opposite corner. Mark the roads with some white tape. Next on the top pipe, drill a hole that can fit a bundle of wires. A rough hole is fine as long as you can access the insides of the pipes. Snake the wires through the PVC pipes and elbow joints for a test fit. Once everything is finalized, paint the PVC with some black spray paint to clean up the look of the main frame. Cut a small gap in one of the PVC pipes to fit a T-joint. A PVC pipe is added to this t-joint for the traffic light to hang down from.

The diameter can be the same as the main frame (1/2”), and if you use a thinner pipe, we should make sure that the 7 wires can snake through. Then drill a hole through this pipe for the traffic light to hang from. Re-wire everything as tested previously. Double check the traffic light and wiring with a breadboard to confirm all the connections have been made. The traffic light is soldered to the wires coming through the T-joint arm. Wrap the exposed wires with electrical tape to prevent any shorts and for a cleaner look.(fig 1).

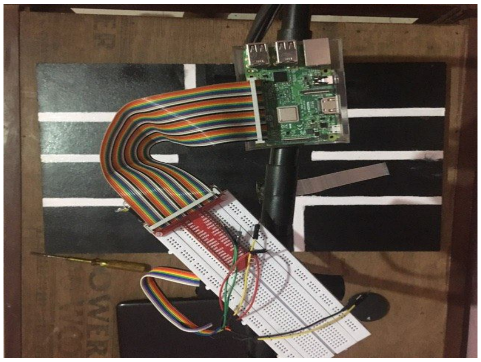


Fig .1 Hardware implementation

IV. OBSERVATIONS

This project takes a stream of data from raspberry pi camera which passes images and changes the traffic light to green for the lane of more cars then to the next lane and so on. If one of the roads has higher traffic density, green signal will turn in that direction and rest will be in red. So that traffic signalling system becomes automatic using image processing through open CV and python. We have a raw image set, threshold images to detect cars and the program takes the position of cars and checks whether it is horizontal or vertical lane. The lane which has the most number of cars will be turned green and others red.

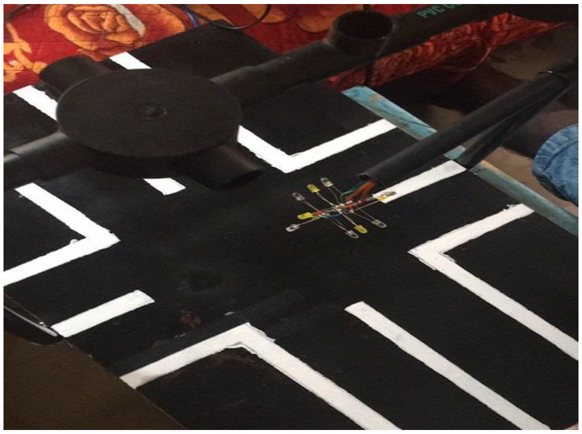
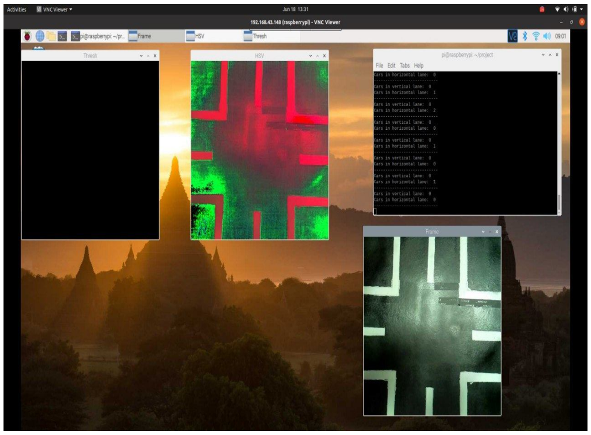


Fig.2 The system



```
Cars in horizontal lane:  0
----------------------------
Cars in vertical lane:  0
Cars in horizontal lane:  1
----------------------------
Cars in vertical lane:  0
Cars in horizontal lane:  2
----------------------------
Cars in vertical lane:  0
Cars in horizontal lane:  0
----------------------------
Cars in vertical lane:  0
Cars in horizontal lane:  1
----------------------------
Cars in vertical lane:  0
Cars in horizontal lane:  0
----------------------------
Cars in vertical lane:  0
Cars in horizontal lane:  1
----------------------------
Cars in vertical lane:  0
Cars in horizontal lane:  0
----------------------------
```

Fig .3 HSV conversion

V. SYSTEM REQUIREMENTS

A. Software Requirements

1) Python: It is a simple , interpreted, high level, general purpose programming language. Its various approaches in language constructs and object orientation aims to help programmers write clear, logical code for all types of projects.

2) OpenCV (Open Source Computer Vision Library) ,is an open source computer vision and machine learning software library .It was built to provide a common infrastructure for applications of computer vision and to accelerate the use of machine perception in the commercial products.

B. Hardware Requirements

1) *Camera:* It is an optical instrument used to record images. Here, the camera is fixed on to the traffic pole in all directions.

2) *Raspberry pi:* The Raspberry Pi is a low cost, credit- card sized computer that plugs into a computer monitor or TV, and make use of standard keyboard and mouse. It is a small device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python.

VI. CONCLUSIONS

Traffic jam has become a serious issue, hence an automated traffic management system is essential in megacities where the number of vehicles has increased by a large extend. Compared to other techniques, using image processing the system can deliver a greater impact and has the ability to implement an efficient result in real-time. extend. The system can detect and identify the object and then counts are compared. The necessary signals are provided accordingly. Hence this would eliminate situations when many cars are stopped while there are no cars on the intersecting road. Not only does this save time for everybody, but it also saves the environment by reducing pollution. Hence traffic can be reduced efficiently without much waiting.

VII. FUTURE SCOPE

We can apply some suitable image processing techniques and enable the system to detect emergency vehicles like ambulances and fire engines to let them pass as soon as possible i.e. providing priority to such vehicles which will save a lot of lives and properties. We can also enable the system to detect number plates of all the vehicles and extract the registration number of vehicles for security purposes. So that, if needed it can help the Police and other authorities to track criminals by identifying the number plate of vehicle in which criminal is escaping.

VIII. ACKNOWLEDGEMENT

The authors would like to thank all who were involved directly or indirectly in the data collection, experiment conduction, and analysis stage of the project.

REFERENCES

[1] Instructables circuits,, Traffic pattern Analyzer https://www.instructables.com/id/Smart-Traffic-Light/

[2] Rajesheswari sunder, Santhoshs Hebber, and varaprasad Golla, “Implementing intelligence Traffic control system for congestion control Ambulance

[3] Faisal A. Al- Nasser,HosamRowaihy “Simulation of Dynamic Traffic control system based on Wireless sensor network”, IEEE Symposium on Computers Informatics 2011, pp 40-45.

[4] M . Abdoos, N. Mozayani, and A. L. C. Bazzan, “Traffic light control in non-stationary environments based on multi agent Q-learning,” in Proc. 14th Int. IEEE Conf. Intell. Transp. Syst., Oct. 2011, pp. 580– 1585.

[5] 2017 IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials 150

[6] Xu Li, Wei Shu, Minglu Li, Hong-Yu Huang, Pei-En Luo, Min-You Wu, “Performance Evaluation of Vehicle Based Mobile Sensor Networks for Traffic Monitoring” IEEE transactions on vehicular technology, May 2009, vol. 58, no. 4, pp. 1647-1653.

[7] Durai. S, Kishore Kumar K, Manikandan N K “Manual Detection Of Feature Envy Bad smell in Software Code” International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 9 (2015) pp. 23319-23331

[8] K. Rajathi, S.Durai, G. Tamilmani, K. Kishore Kumar, “Multipoint Bitmap Filter for Large Volume Data Query Processing” Indian Journal of Science and Technology volume 10, Issue 4, January 2017.