



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VIII **Month of publication:** August 2022

DOI: <https://doi.org/10.22214/ijraset.2022.46280>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

On Board Complete Traffic Signalling System Using Image Processing

M Lohitha¹, M Bhargava Krishna²

^{1,2}Electronics and Computer Engineering, Sreenidhi Institute of Science and Technology (SNIST), Hyderabad, India

Abstract: In this paper, an on-board traffic signalling system which can easily curb the problem of traffic congestion using image processing techniques is presented. In this system, the images of the four sides of the road are captured and the vehicle density is calculated using the haarcascade. This density is used to sort the sides of the roads with the respect to the highest density among the 4 and evaluate its timer for the green signals which can last for maximum of 80 seconds. The traffic on the sides of the roads are released in this sorted order of the vehicle density. After a round of all the roads receiving the green signal, the images of the 4 sides of the roads are again captured and it continues.

Keywords: Traffic lights, image processing, jetson nano, vehicle density, timer.

I. INTRODUCTION

With the fast advancement of road framework and drastic increase in the population growth, the number of vehicles grows rapidly and thus results in traffic jam. The main reason for the traffic congestion is the massive rise in the number of vehicles on the roads. This makes travelling on the roads a very difficult task in the present day. The problem of traffic congestion prevails even today and many solutions have been proposed in order to deal with this problem. The currently implemented model is also known as the timer model. In this model, timers are used to control the traffic. The timer is given a fixed amount of delay for each side of the road and the traffic is released in the specific order each time. Traffic congestion increases vehicle emissions, which in turn leads to air pollution.

Traffic congestion [1] has a huge impact on the environment. In some cases, it also leads to road rage among the drivers and negatively impacts human health. The problem of traffic congestion can be solved by implementing this on-board complete traffic signalling system using image processing. This method greatly curbs the traffic by controlling the traffic lights based of vehicle density.

The proposed model can achieve using the techniques of image processing. In this model, the image processing techniques are used to detect and count the number of vehicles also known as vehicle density. Nvidia Jetson Nano is used to evaluate the timer for the green signals which can last for a maximum of 80 seconds. LEDs are used to depict the working of the traffic lights.

II. PROPOSED SYSTEM

The proposed system is implemented in Python and executed in jetson nano. We use OpenCV for image processing. The proposed system can be accomplished by using image processing techniques.

Four cameras are installed at the intersections of the roads and these cameras are used to capture the images of four sides of the road and the vehicle density is calculated using the haarcascade [2].

This density is used to sort the sides of the roads with the respect to the highest density among the 4 and evaluate its timer for the green signals which can last for maximum of 80 seconds using Nvidia Jetson Nano. The traffic on the sides of the roads is released in this sorted order of the vehicle density.

After a round of all the roads receiving the green signal, the images of the 4 sides of the roads are again captured and it continues. The implemented system comprises of the following stages: a) Image capture b) Density calculation c) Sorting d) Timer Allocation e) Signalling

A. Image Capture

The first step of the proposed system involves capturing of the images of the 4 sides of the road. In order to capture the images, 4 cameras are installed at the intersection of the roads. The captured images are then converted to the png format and then these images are converted into grayscale for further processing which involves detection of the cars using various python functions.

B. Density Calculation

The next step in the system is the density calculation. A pre-trained model is used to detect the cars. The pre-trained XML model containing the feature set is loaded using the Cascade classifier. Cascade classifiers are trained with a lot of positive and negative examples. The trained classifier can be applied to a portion of the image in order to detect the required object. When the camera's capture the images of the roads, these images are read using the imread function and then they are converted into grayscale. The detectMultiScale method detects items of different sizes in the supplied picture. A list of rectangles is returned for the detected items. The detectMultiScale function is used to find the bounding boxes of the detected cars. A variable is used to keep track of the number of cars detected in the frame. Based on the car count we decide whether the traffic is low, medium or high.

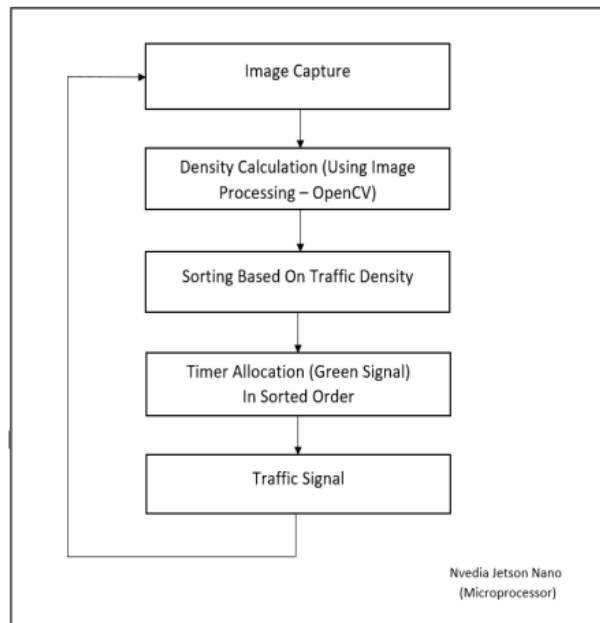


Fig1: Block diagram

C. Sorting

After density is calculated for the captured images, the sides of the road are sorted from high density to low density. The lane having the highest density among the four lanes is given the green light first.

D. Timer allocation

The timer is allocated based on some conditions. If the car count is less than 5 then it is considered as a "no traffic" condition and a delay of 10 seconds is allotted. If the car count is greater than or equal to 5 and less than or equal to 10 then it is considered as a "low traffic" condition and a delay of 20 seconds is allotted. If the car count is between 10 and 15 then it is considered as a "medium traffic" condition and a delay of 30 seconds is allotted. If the car count is greater than or equal to 15 then it is considered as a "high traffic" condition and a delay of 60 seconds is allotted [3]. After the density is calculated and captured images are sorted based on the density, the timer is allocated based on these conditions. The timer for the green light can last for a maximum of up to 80 seconds

E. Signalling

The traffic on the sides of the roads is released in the sorted order of the vehicle density. The lane that has the highest density is given the green light first. After 10 seconds an image is captured for the same lane to check whether the density is increasing or not. If the density is increasing, then the timer will increase based on the conditions. The timer can increase to a maximum of 80 seconds. After the lane is checked dynamically and given required delay, the red light is given to the lane. Now the green light is given to the lane with the second highest density and the same process is repeated for the remaining two lanes, selecting the lane with higher density first. After a round of all the roads receiving the green signal, the images of the 4 sides of the roads are again captured and it continues.

III. IMPLEMENTATION

This is a real time implementable project, implementation of software section of the project basically includes three main parts in it namely Vehicle detection and density calculation, ambulance detection and Prioritising, traffic lights and signalling. The hardware implementation is done on NVIDIA Jetson Nano Because of its virtue in the performance as a microprocessor. NVIDIA Jetson Nano has 40 GPIO pins, these pins are capable of taking inputs to the controller and give outputs from the controller hence the name general purpose input output pins. The hardware section is explained below:

A. NVIDIA Jetson Nano

This is a very powerful and reliable SOC [System on Chip] that is capable of high-level processing like image processing, machine learning, deep learning, artificial intelligence, etc., as a result, this heavy duty of the processor it gets heated up very fast and needs a good heat sink or a cooling system. The previous versions of Jetson by the NVIDIA have a cooling fan on the main processor for the cooling purpose.

B. LEDs

The traffic lights are demonstrated with the help of LEDs (Light Emitting Diode). The LEDs used are not conventional single coloured or monochromatic LEDs but are tri-coloured LEDs, these are special type of diodes which have not just two electrodes but actually four of them. The Tri colour or RGB LEDs are available in the market in two types, one being common anode and the other common cathode.

C. Circuitry

The proposed system deals with four lines or four roads with a traffic signal at the intersection. Hence there will be a requirement for a total of 8 signals each signal Controlling one red and one green light per road. As we have used a common anode LED For the demonstration, we will have to short all the anodes and give a positive supply to it for the LED to get into the forward bias condition and do its working. The circuit will be completed and the LED will be in forward bias when any one of the cathodes (RGB) is given a comparatively negative power supply, in our case grounding the power supply. For the above-mentioned circuitry to be in action, we've written Python program which uses an exclusive Jetson Nano GPIO library and gives us access to those 40 pins [4]. The LED terminals as discussed and shown in the circuit diagram are connected to, he respected pins on NVIDIA Jetson Nano. Each pin being used is declared and defined in the Python code as an output pin, these pins are given outputs of either high voltage or a low voltage level, and these voltage levels decide the positive voltage or the grounding voltage. As the required terminals of green and red colours are already connected to the GPIO pins and the common anode terminals being short and given to one pin, now the control of voltages is done with the help of the code as mentioned earlier with high and low voltage levels.

As mentioned, the common anode terminal needs to be given a positive supply i.e., a high voltage level to enable the LPD, the other terminals can be controlled as required with giving a high voltage level for that coloured light to turn off and a lowered his level for the colour of light connected to that terminal to turn on. Our system intelligently allots an amount of time to a particular lane for it to have a green signal and also the order of the signalling, hence the command for these LEDs to glow either a green or red light is given by our system measuring the traffic density [5]. It is ensured that if one lane has a green signal all the other lanes will have the complementary, which is the red signal.

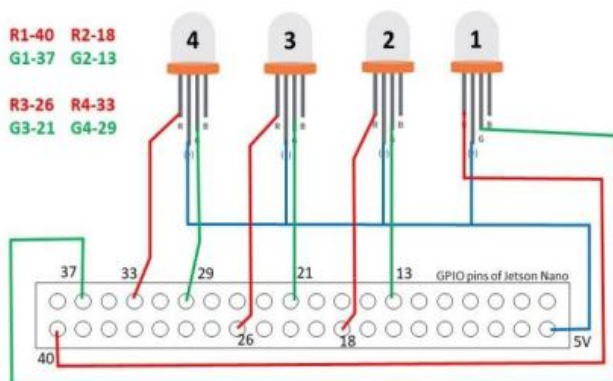


Fig2: Circuit diagram for hardware implementation of the system.

IV. ALGORITHM

The algorithm of our proposed system comprises of the following steps.

- 1) Import the libraries
- 2) Store the trained xml classifiers in a variable
- 3) Capture images of the roads
- 4) Convert the captured images to grayscale
- 5) Detect and count the number of cars
- 6) Calculate the density of the captured images
- 7) Sort the images from high density to low density
- 8) Give green light to the lane with highest density
- 9) Check that lane dynamically
- 10) Increase the timer (maximum 80 seconds) if density of that lane is increasing
- 11) Give red light to that lane after giving required delay
- 12) Give green light to the lane with second highest density
- 13) Repeat the same process for remaining two lanes 14. After all the lanes are given green signal once, capture fresh images of the roads are repeat the same process.

V. OUTPUT



Fig3: Image of the lane captured by the camera



Fig4: Image showing bounding boxes of detected cars in the lane

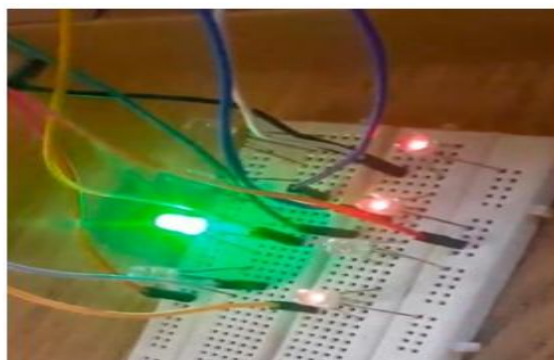


Fig5: Image showing green light for the lane



VI. CONCLUSIONS

This suggested system provides a solution to the problem of traffic congestion by employing image processing skills. We create a system that uses the Nvidia Jetson Nano, an onboard powerful computer, and image processing algorithms to monitor and manage traffic signals based on vehicle density. The system is constructed and simulated in auto signalling mode to monitor traffic density.

REFERENCES

- [1] Chandana K K, Dr. S. Meenakshi Sundaram, Cyana D'sa, Meghana NSwamy, Navya K, "Smart traffic management system for Congestion control and warnings using internet of things (IoT)," Chandana K Ketal.; Saudi J. Eng. Technol.; Vol-2, Iss-5(May, 2017):192- 196
- [2] Md.Rokebul Islam, Nafis Ibn Shahid, Dewan Tanzim ul Karim, Abdullah A Mamun, Dr. Md. Khalilur Rhaman, "An efficient algorithm for detecting traffic congestion and a framework for smart traffic control system," Jam. 31~Feb. 3, 2016.
- [3] Gundeep Kaur, Sonia Sharma, "Traffic Management Using Digital Image Processing", IJCST Vol. 8, ISSue 2, AprIl - June2017.
- [4] Md. Munir Hasan, Aminul Hoque, Gobinda Saha, and Md. Badruddoja Majumder. "Smart Traffic Control System with Application of Image Processing Techniques", 3rd International Conference on Informatics, Electronics & Vision2014.
- [5] G. Monika, N.Kalpana, Dr.P.Gnanasundari, "An intelligent automatic traffic light controller using embedded system," Volume 4, Special Issue 4, April 2015.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)