Traffic Monitoring System Using Image Processing

PRANALI. R. YAWLE[1], KANCHAN. D. MAHAJAN[2], SHITAL.V. SHELKE[3], ATUL. B. VITEKAR[4], MAHAVIR.S. KASAR[5], RAJESH U. YAWLE[6]

[1, 2, 3, 4, 5] *Dept. of Electronics and Telecommunication Engineering, Bharati Vidyapeeth's College of Engineering for Women, Pune, India*

[6] *Sinhgad Academy of Engineering, Kondhwa (Bk), Pune, India*

Abstract—In India, due to the increase of automobiles, traffic congestion at junctions has become a significant problem. Vehicle density is steadily rising, necessitating the urgent installation of adaptive traffic lights that can track traffic density in real time. By taking image of the traffic at a junction, this system makes excellent use of image processing to control the flow of traffic. Following a step-by-step procedure of image collecting, processing, and algorithm implementation, the duration of the traffic light is changed depending on the volume of vehicles on different roads at the traffic signal.

Indexed Terms— Arduino Nano, Camera calibration, Image resizing, Thresholding, Traffic congestion.

I. INTRODUCTION

On a daily basis, traffic lights are crucial for regulating and controlling traffic. Three lights make up the modern traffic signals: red for stop, yellow for wait, and green for go. The signal must first shift from red to yellow, then from yellow to green, before users can proceed. The length of time a commuter must wait is determined by traffic signals. The modern traffic lights are hardwired when they are installed. They are pre-programmed to wait for a predetermined period of time when a signal changes. It operates without regard to the volume of traffic on the roadways and doesn't change. On sometimes, a certain road will be significantly more congested than others. By giving the vehicles on the busiest road more space to pass than those on other, less busy roads, you can easily relieve traffic congestion. The system ought to be capable of making daily priority decisions.

In this method, the waiting time for drivers on roads with larger densities is essentially decreased. The image for each lane are captured and processed simultaneously as a result, and a judgement is made regarding which lane should be given the most time and attention.

A camera is used to take image of the junction of the roads. The density of vehicles on each road at that same moment is then calculated using the images that were captured. Each road receives a priority list in a single cycle, and the waiting time for that road is adjusted based on its density. The amount of traffic at the junction is decreased by giving a denser road more time to pass all of its vehicles. As this system uses a camera located on top of the signal to take pictures and convey them for image processing, it is less susceptible to hardware failure. Python programming language is used to calculate the density of the roads, and the controller modifies the length of the green light for each road based on the results of image processing.

II. METHODOLOGY

A. Image acquisition

A webcam is used to record the image. A USB cable is then used to transfer it to the computer. Python programming language is used for picture acquisition and subsequent processing.

B. Image processing

Using a webcam positioned at the junction of two roads, the image is taken. It is capable of snapping pictures of the intersection where the roads converge. For further processing, the acquired image is changed to a greyscale version. After that, the grayscale image is transformed into a binary image, which only has the two colours black and white. This image is known as the threshold image. Thresholding the image is mostly used to drastically reduce the amount of information in order to streamline further processing. After that, the threshold picture is supplemented for additional image processing.

i. Image Resizing/Scaling

All digital photographs experience image scaling at some point, whether it's during Bayer demosaicing or photo enlargement. Every time you change the size of your image from one pixel grid to another, it happens. When you need to change the total number of pixels, image resizing is required. Depending on the algorithm used, even when the identical image is resized, the outcome can change greatly.

There are several reasons why images are shrunk, but one of them is crucial to this system. Every camera has a different resolution, so when a system is created for one set of camera specifications, depending on how closely those specifications differ for other cameras, it may not function properly. It is therefore necessary to make the resolution constant for the application and thus perform image resizing.

ii. Image Enhancement

This procedure involves adjusting the photos so that the outcomes are better suited for processing. In this, the obtained image is turned into a greyscale image.

iii. Thresholding

The greyscale image is converted to a black-and-white image (binary: white=1, black=0) using the thresholding process. Thresholding is mostly used to drastically reduce the amount of information in order to streamline subsequent processing. All of the pixels with luminance greater than the threshold level are given the colour white, while the others are given the colour black.

C. Count of Vehicles

The traffic signal is set based on how many vehicles there are overall.

III. BLOCK DIAGRAM

Using the camera, traffic intersection photographs are captured. After the image processing unit receives it, basic picture processing will start. To determine whether a picture of a car or another type of vehicle is presented, a machine learning classifier is used. It will display the total number of automobiles. Arduino and PC communicate through serial lines. According to total no of vehicles delay is set to the traffic lights.

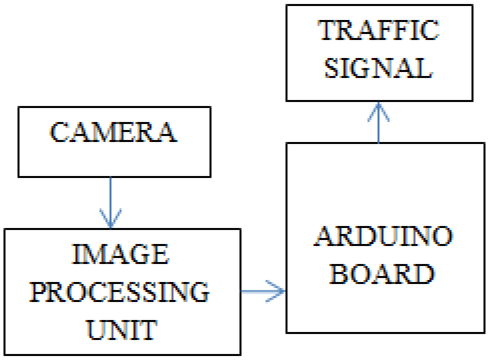

Fig 1: Block diagram of Proposed System

A. Hardware

i. Camera

It is a high-quality device that is compact and elegant. This little camera can be used on a laptop or a desktop. With the aid of the camera, we were able to record traffic photographs that are now being sent to the image processing unit for further processing.

ii. Image Processing Unit

This unit displays the following operations: 1) Image Acquisition 2) Image Processing - Image Resizing - Image Enhancement – Thresholding

iii. Arduino Board

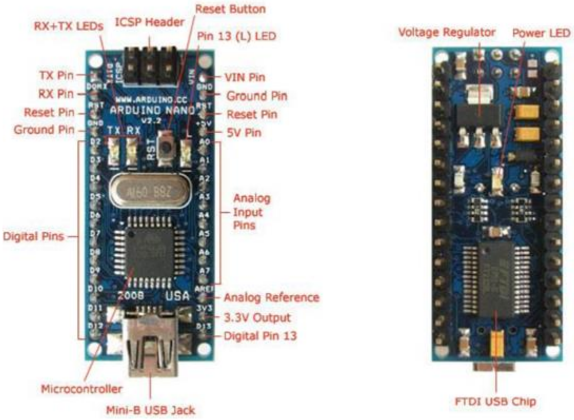

Fig 2: Auduino Nano Board

The breadboard-based ATmega328 device is compact, comprehensive, and has Arduino Duemilanove capability. Power is provided by a Mini-B USB connector, which continues to a traffic light.

IV. ALGORITHM

1. Start
2. Access camera in system.
3. Take image from system.
4. Basic image processing will take place i.e.-Filtering, etc.
5. Machine learning classifier to detect car or other vehicles. It will give total number of count.
6. Serial communication is used between PC and Arduino.
7. Comparing the total count of vehicles with the threshold count of vehicles with the threshold count, i.e. heavy, medium, low output of it goes to Arduino and set the delay.
8. Depending on that green lights of signal will change.

V. HARDWARE DESIGN

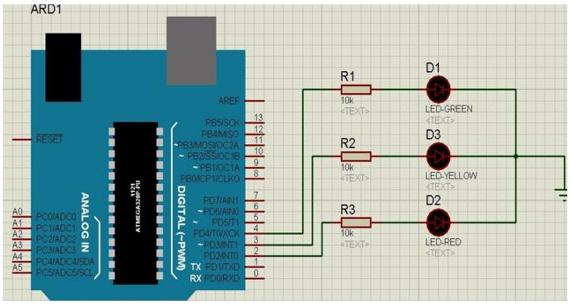

Fig 3: Arduino Board connections with LEDs

VI. CONCLUSION

This research presents a technique for estimating traffic using image processing. This is done by using the camera images captured. The number of cars has been counted after each image and analyzed independently. The timing of red traffic lights becomes minimum and the timing of green traffic lights becomes maximum if the number of vehicles exceeds a predetermined level. This novel technique offers benefits including using image processing rather than sensors, low cost, simple setup, and quite excellent accuracy and speed. Production expenses are inexpensive while attaining high speed and precision using Python software and Image Processing tools.

REFERENCES

[1] Li Z, Shahidehpour M, Bahramirad S & Khodaei, "*Optimizing traffic signal settings in smart cities" -*IEEE Transactions on Smart Grid, Vol: 8, Issue: 5, September 2017

[2] Vismay Pandit, Jinesh Doshi, Dhruv Mehta, Ashay Mhatre and Abhilash Janardhan, *"Smart Traffic Control System Using Image Processing"*- International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)Volume 3, Issue 1, January – February 2014

[3] Er. Kavneet Kaur, Vijay Kumar Banga, *"Number Plate Recognition Using OCR Technique"* - IJRET: International Journal of Research in Engineering and Technology eISSN: 2319-1163 pISSN: 2321-7308 Volume: 02 Issue: 09 Sep-2013

[4] V. Milanés, J. Villagrá, J. Godoy, J. Simó, J. Pérez, and E. Onieva, *"An intelligent V2I-based traffic management system,"* IEEE Trans. Intell. Transp. Syst., vol. 13, no. 1, pp. 49–58, Mar. 2012

[5] H. Chourabi, T. Nam, S. Walker, J. R. Gil-Garcia, S. Mellouli, K. Nahon,T. a. Pardo, and H. J. Scholl, *"Understanding smart cities: An integrative framework,"* Proc. Annu. Hawaii Int. Conf. Syst. Sci., pp. 2289–2297,2011.

[6] Singh L, Tripathi S & Arora H, *"Time optimization for traffic signal control using genetic algorithm*" - International Journal of Recent Trends in Engineering, vol. 2,no. 2, November 2009

[7] J. Melo, A. Naftel, A. Bernardino and J. Santos-Victor, *"Detection and classification of highway lanes using vehicle motion trajectories"*- IEEE Transactions on Intelligent Transportation Systems, vol. 7, no. 2, pp. 188-200. 2006

[8] L. Li, W. Huang, I. Y. Gu, and Q. Tian *"Foreground object detection from videos containing complex background"* ,- Proc. of the Eleventh ACM international Conference on Multimedia, Berkeley, CA, USA, November 02 08, 2003.

[9] R. Cucchiara, M. Piccardi, and P. Mello, *"Image analysis and rule-based reasoning for a traffic monitoring system"*-IEEE Trans. Intell.

Transport. Syst., vol. 1, no. 2, pp. 119-130, June 2002.