Abstract

This project suggests a better system to monitor and control traffic. A real time approach of image acquisition, image processing followed by implementation of an algorithm to change traffic light time interval based on the density of vehicles at a junction is followed.

Problem Statement

One of the main reasons behind today's traffic problem are the techniques that are used for traffic management. It has no emphasis on *live traffic scenario*, thus leading to inefficient traffic management systems. These traffic timers just show the pre-set time. Also, traffic density is ignored while managing the time interval for which these lights switch colours.

If the traffic light timers are showing correct time to regulate the traffic, then the time wasted on unwanted green signals will be saved. Timer for every lane is the simplest way to control traffic. And if those timers are predicting exact time then automatically the system will be more efficient and this project is working on balancing that time and its switching between Red and Green lights.

Methodology

The best possible way to deal with bottleneck situation at a junction is to provide it longer period of Green light than others with less density of vehicles. This will result in reduction of vehicle density at that particular junction. This is done using capturing images and working on them continuously.

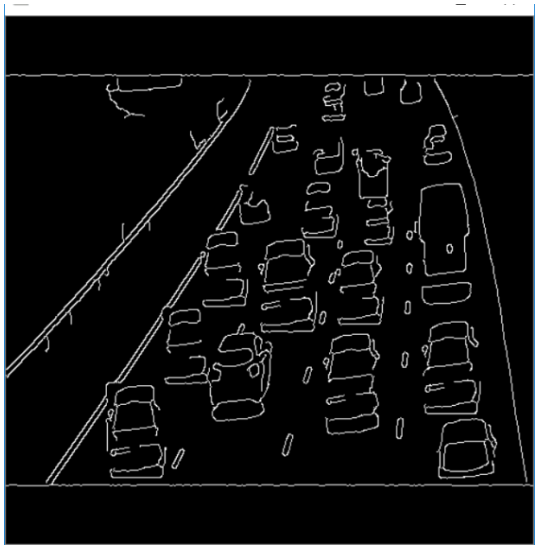
Step1: OpenCV Approach

Cameras are installed on each traffic light junction to find number of vehicles on each side of the intersection.

The Image taken is processed:

1. Read using OpenCV Library in Python
2. Resized according to need
3. Converted to Black and White
4. Removing Gaussian Noise
5. Removing Salt and Pepper Noise
6. Dilation
7. Take Difference from reference image
8. Find the percentage change
9. Get Number of Cars

Fig(i): B/W image



Fig(ii): Edge Detection

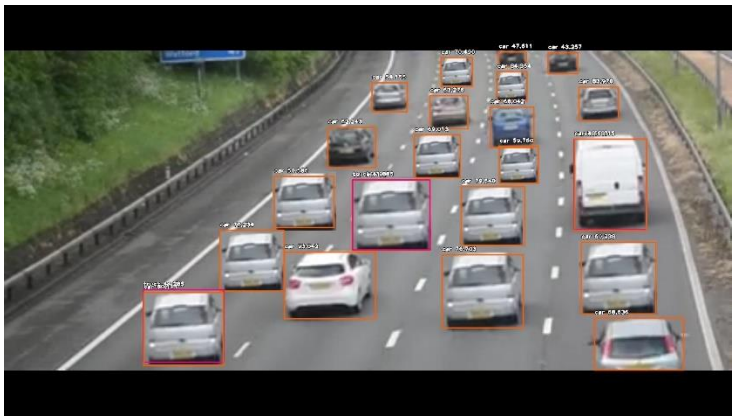
Step 2: Machine Learning Approach

The Image taken is now:
1. Pre-processed to determine vehicle's path
2. Either a Model is inherited or New one is made to detect cars and different vehicles
3. Image is loaded and all the detections are saved and later analysed
4. Cars are counted in the detections.

Step3: Now, we have number of vehicles on each side of intersection so, we have to develop an algorithm to solve this overcrowded junction problem. This algorithm only comes into play when number of vehicles at 1 junction is way higher than the rest 3. Following conditions must be followed while trying to

develop this algorithm:

- Cycle is completed i.e. Once a signal is green, it turns green again once all 3 other signals get their chance.
- Max Waiting time i.e. The maximum waiting time for any side is 180 seconds or 3 minutes. If a side has very a smaller number of cars e.g. 3 then the corresponding side will turn to green instead of making those few cars wait for 180 seconds at max.
- Single sided case i.e. In a situation there is just one-sided traffic flow and all the other sides of the intersection are empty then complete preference will be given to the corresponding side. **This is very useful in situations like evacuation.**



fig(iii): Count vehicles on road

There are 2 type of algorithms that are developed in the project:
1. Variance-Based Algorithm: Variance in statistics is a measure of how much a value deviates from its expected value. We calculate variance of number of cars on all 4 sides, time slots are allocated to all 4 junctions in such a manner to bring max. value close to other values and making the list less complex and more uniform.

2. Slab Division Algorithm: Once the number of cars are calculated in each lane, Slabs can be used to allocate time slots to different sides with the side with maximum number of cars given the priority of crossing the intersection and taking care of the above conditions so as to keep the flow of traffic efficient , smooth and fair . Timers to the slabs are allotted so as to make the maximum number of cars in that slab half, by letting half of the cars pass the intersection.

Variance-based Algorithm	Slab Division Algorithm
Calculates the Time according to the variance of the List	Calculates the Time according to a Single index.
Results in a more Dynamic approach	Comparatively low Dynamic approach
More efficient	Less efficient

Table(i): Comparison of 2 algorithms

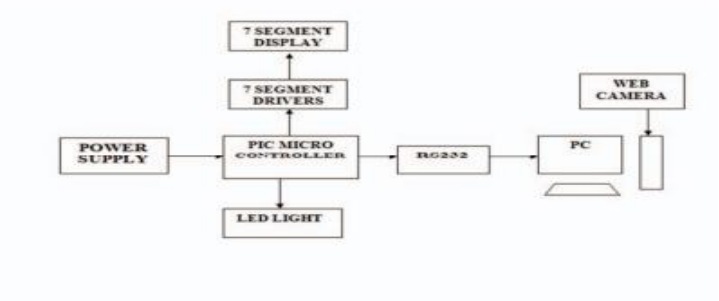
Literature Survey

We have gone through 3 different paper in order to study existing state-of-the-art work. The takeaways from each of them is mentioned below:

1. **Chirag Thakkar, Rajesh Patil** proposed a technique which uses a reference image for detecting the amount of traffic on the road. This reference image is taken when the road is completely empty in order to compare this image with the images that will be captured when vehicles pass by the traffic light. The images are captured using a digital camera on the traffic light which are processed in MATLAB. The image is first pre-processed where it is scaled and converted from RGB to Gray Scale image. This image is enhanced to remove several problems such as difference in illumination, poor contrast, noise, etc. Canny Edge detection is applied on the image which is one of the most important phase in order to find sharp changes in brightness in the image which can increase the change of the captured image with the reference image. This change is irrelevant and canny edge detection resolves such problem. The final phase is the comparison of the captured image with the reference image and based on the difference between the two images the traffic light will work.

2. **Wojcikowski, M.Zaglewski, B.Pankiewicz** described the idea and the implementation of the robust algorithm dedicated to extraction of moving vehicles from real-time camera images for the evaluation of traffic parameters, such as the number of vehicles, their direction of movement and their approximate speed. The algorithm is designed and implemented in pipelined hardware in FPGA; therefore, high frame-rate efficiency has been achieved. The possible applications of the proposed algorithm and its implementation are in the area of intelligent image processing sensors in low-power, cost-effective sensor network applications, i.e. for detecting the moving vehicles on the road. The already existing systems are Pneumatic road tube, Video Image Processor which are also used but here the proposed system is much better which is able to sense the presence of a number of vehicles in a certain range by using the sensors and time allocation for the direction of

traffic. The 2014 IEEE International Conference on Computational Intelligence and Computing Research Block diagram shown below. The camera is fixed to identify the number of vehicles present. According to the output of the camera the traffic signal gets changed. Time is varying according to the density. Need not to wait for a longer time. Density of all the roads is identified by a single camera. The camera is connected to the PC and the images of the roads are given to the PC. In PC, MATLAB is used for identifying Objects on the road and depending on the number of objects the time is calculated. For more traffic, the green signal time is more and when the traffic is less the green signal is less. According to these values the micro controller instructs the led to glow. Hence, based on the traffic density it changes the traffic signals. The time seconds are indicated in the seven-segment display which is connected with the microcontroller. The micro controller controls the LED and the seven-segment display according to the output of the PC.



Fig(iv): Block Diagram for methodology

An experimental result shows that the semantic context information is effective to improve object detection, object classification and abnormal event detection. To simplify the hardware, a set of morphological operations can be used. The novel combinations of masks from backgrounds enhance the detection performance. Future work will be focused on a more detailed evaluation of the results of the system and comparison with existing data. The future enhancement is to distribute the system results to remote clients.

3. **P. Chaudekar, S. Banerjee, M.K. Muju** discusses the advancement of the technology in identifying the traffic density of the road. As a result of which we can alter the timings of the lights and make them to be optimized to the time for which each car has to wait on an average in the rush hours especially.

   The Main Aim/Advantages of the Paper:

   1) Identification of an empty road and turning the light to go RED.
   2) Turn the traffic light Red in case the max time for the green light.

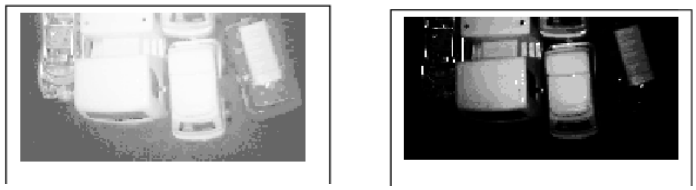
3) Analyze the vehicle on the road and turning the light to corresponding color.

The Important processes to carry out the above tasks:

1) Image Acquisition
2) RGB to grey conversion
3) Image Enhancement
4) Image matching using edge detection

Process/ Procedure:

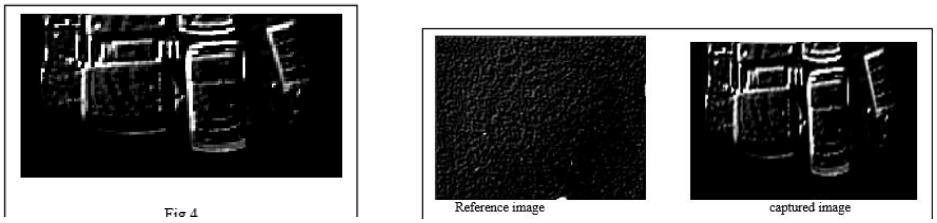
1) **Image Acquisition**: This Initial process is carried out by the simple Web Camera. The Image of the Road is taken, initially when there is no traffic at all (This particular picture is the reference image for the further analysis and application).
2) **RGB to grey:** This is done on a number /sequence of the images we have Acquired. Which is further followed by the gamma correction of the same images.
3) **Image Enhancement:** After the grey conversion we had to make the image contrast right to distinguished it from the background. The Prominent function used for the same are:
    - **Linear transformations**
    - **Logarithmic transformations**
    - **Power Law transformations**
    - **Piecewise Linear transformations**



fig(v): B/W images

4) **Image matching using edge detection:** Out of a number of parts of the image we need to decipher each of the lines, edges points and then Match the image with the reference image.

   For the purpose of Edge Detection, we take use of gradient Based Edge detection. So that we can detect the edges along horizontal, vertical and diagonal directions



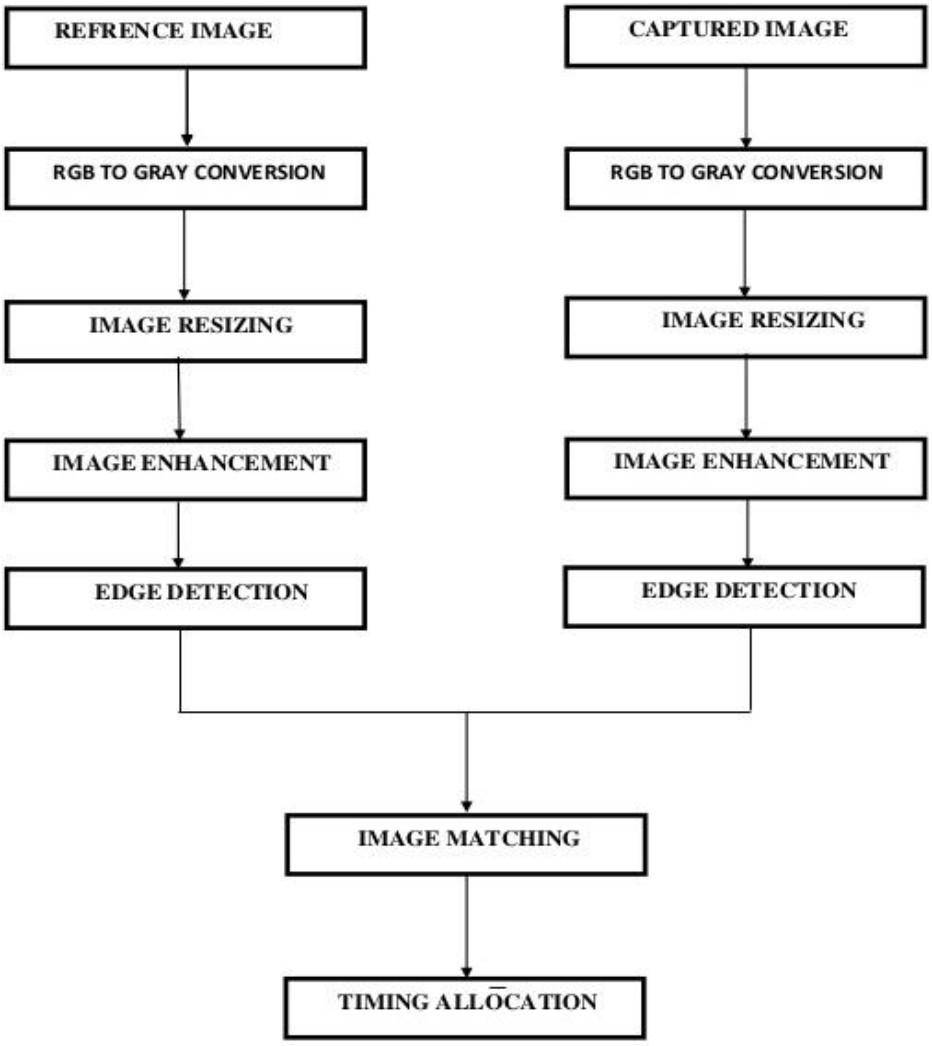
Fig(vi): Edge Detection for fig(v)

After image processing of the Image i.e. from image acquisition to the edge detection, the final image is further matched with the reference image to analyze the result.

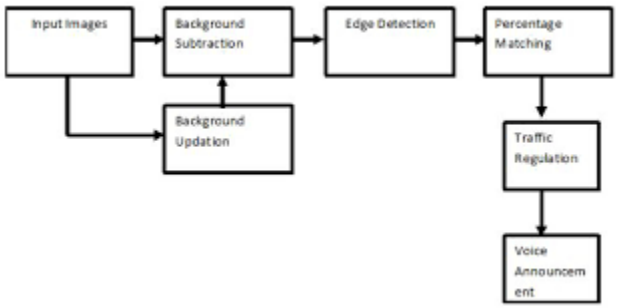
Failure points for existing work:

S.No	Shortcomings	Solution
1	Bad Atmospheric conditions will degrade the image therefore it will affect the working of traffic light and thus the risk of accidents will increase. It fails for weather cases where the visibility is considerably less.	High resolution cameras and further morphological operations to have clearer image for edge detection.
2	Animals and people near the road will also be considered as traffic.	Use of better techniques while training the model.
3	Inability to distinguish between the vehicles itself. Ambulance, fire trucks and police vehicles have priority over other vehicles to move ahead but in this case it doesn't seem possible.	Introduction of OCR so that we can distinguish vehicles that have higher priority than others and set traffic light timer and color according to that.

Block Diagrams



Fig(vii)



fig(viii)

Requirements

GUI/Simulator Development

- Python and JAVA (including Frameworks)
- JS Libraries like Three.js

Image/Video Processing

- OpenCV using Python or c++
- Libraries like imageAI

Machine Learning

- TensorFlow
- Sckit-Learn
- Keras
- Pillows

Data Analysis

- R

How our approach is different

Till now what has been done is a reference image is used to check the difference in density of vehicles at the junction and then traffic lights are assigned time slots dynamically. But we try to set a Machine Learning model with it which tells about the behavior of most vehicles at that junction so we can predict in which direction they are most likely to go which will further help our algorithm to make an intelligent decision on setting timers and color to traffic lights.

Project Status

This project is still under R&D phase. Some small-scale testing has been done on individual levels but practically working with it on a large scale is yet to be done.

Inception of this project

We conceived this idea while studying about smart cities and how traffic moderation and monitoring is feasible by using Image processing techniques as well.

Software Used

- R Studio
- Jupyter Notebook
- Visual Studio
- Java Virtual Environment

References

- Chirag Thakkar, Rajesh Patil, *"Smart traffic control system based on image processing", International Journal for Research in Applied Science & Engineering Technology (IJRASET) Volume 5 Issue VIII, July 2017*
- P. Chaudekar, S. Banerjee, M.K. Muju, *"Real time traffic light control using image processing", Indian Journal of Computer Science and Engineering (IJCSE) Volume 2 No. 1, 94-98(2011)*
- Wojcikowski, M.Zaglewski, B.Pankiewicz,*"FGPA based real time implementation of detection algorithm for automatic traffic survillence", Sensor Network J. sig. Process System, 68(1), 1-18(2012)*