Traffic Monitoring System

Aranta Rokade[1], Ashish Naik[2], Rashika Koul[3], Shubhi Khare[4]

[1,2,3,4] Computer Science Department,
George Mason University,
Fairfax, VA, USA

[1]*arokade@gmu.edu,* [2]*anaik6@gmu.edu,* [3]*rkoul2@gmu.edu,* [4]*skhare@gmu.edu*

ABSTRACT

Traffic Monitoring System (TMS) is a crucial element in Intelligent Transportation Systems (ITS), especially in places with varied traffic flow of a variety of vehicles like cars, motorcycles, and other heavy vehicles. Due to this, the incorporation of Traffic Monitoring System for dynamic control of traffic light timing is essential. The timing of the traffic light can be dynamically adjusted according to the estimated traffic which is based on the video sequence captured by the camera in its span on the road. The traffic can be described by features such as the density of the traffic and the flow of the traffic which is the number of vehicles that have passed in a certain duration. Thus, a method was implemented to detect and count the number of vehicles that have passed in a certain duration and, to estimate the density of the road occupied on both sides. For this, computer vision based techniques such as Background Subtraction, Contour Detection and Region of Interest (ROI) Mask were used. To test the method, the video was evaluated by manual inspection.

1. INTRODUCTION AND RELATED WORK

The Traffic Monitoring System will essentially calculate the traffic flow and density at regular timestamps, which along with a threshold value can be used to get a sense of what's "too crowded" or "too slow" respectively. The proposed approach is a simple, fast and economical implementation using various computer vision techniques such as background subtraction, filtering, contouring, without the use of any heavy deep learning algorithms. In background subtraction, the background layer is defined as the static portion of a given set of frames. This background layer is used to separate the foreground objects from itself. The filters such as threshold, erode, dilate, open and close are used to remove noise from the foreground object and make it bolder. And finally, contouring is used to detect and count the objects. The results obtained by implementing the aforementioned techniques can be used to calculate the traffic flow. The density of the traffic is calculated with respect to a region of interest with the help of pixels occupied by the vehicles and that unoccupied.

Different ways have been proposed to build the traffic system smarter, reliable, and robust. A real-time optimization model was used by Dotolie et al. investigated the problem of traffic management in urban areas. Albers et al. used real-time data to monitor current traffic flows during a junction in order that the traffic may well be controlled in a convenient manner. Reliable short-term forecasting video captured in a recorder plays a vital role in monitoring the traffic management system. The data required can be easily provided by the CCTV cameras that can be beside the roads as per requirement.

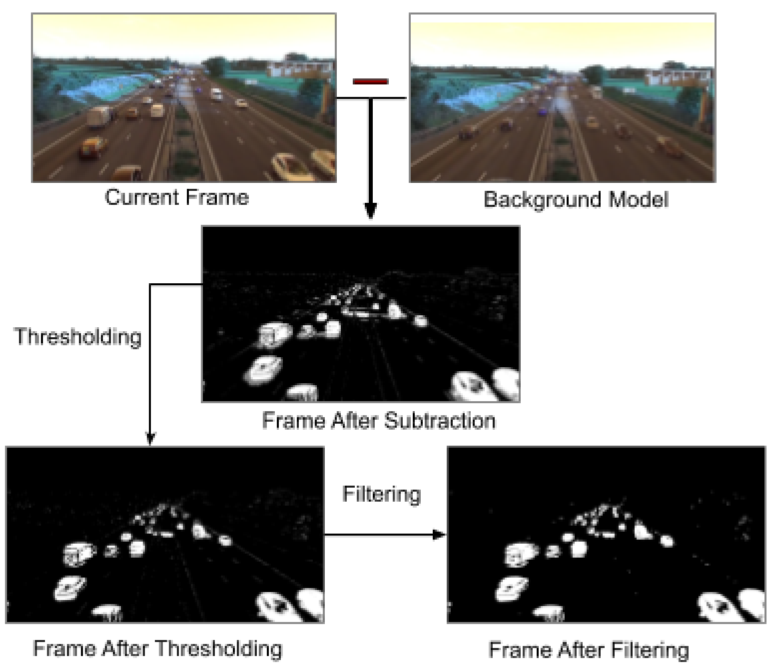
2. TECHNICAL APPROACH

The system implemented two main tasks that are, count the number of vehicles that have passed (i.e. the traffic flow) and calculate the density of the traffic. The methodology used for both the tasks is discussed further in detail.

CAMERA SETTINGS:
The camera is mounted on the median strip of the road and captures the traffic coming towards it and that going away from it. The camera is at a height sufficient to capture the vehicles in the desired stretch of road. The dimensions of each frame are 720 x 1280 px.

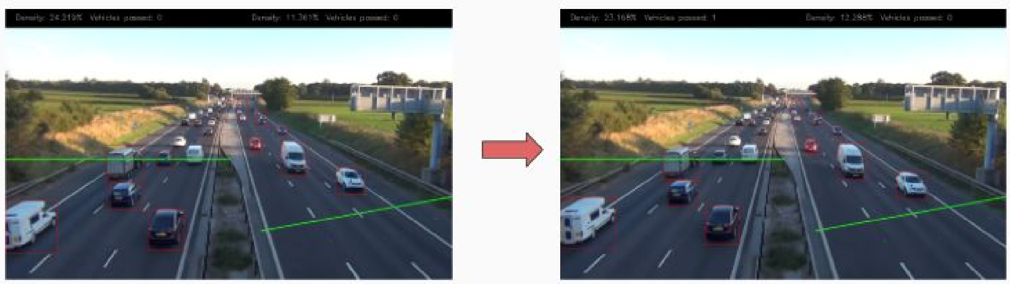
TRAFFIC FLOW:
Before processing the frames, a region of interest was selected for counting the vehicles. It was observed that if the corner points (of the region of interest) too close or too far from the camera are selected, vehicles too close to each other are detected as a single object or not detected at all, thus giving inaccurate results. Thus, the corner points of the region of interest were decided upon on the basis of the accuracy of object detection.

A background image was derived using the apply method of createBackgroundSubtractorMOG2 API from OpenCV, where a history of 500 frames was used as a training set and the parameter 'detectShadows' was set to true so it doesn't detect the shadows as separate objects. This background image was updated after every frame. The apply method also returns the foreground objects. The advantage of updating the background image after every frame is that scene changes due to change in sunlight, or movement of clouds are taken into consideration.



Foreground detection and contouring pipeline

After obtaining the background, each frame was processed. The background was then subtracted from every frame to get the foreground image containing the vehicles. Thresholding was applied on this foreground, and all the values less than 175 were set to 0, thereby removing the shadows. Furthermore, morphological operations, such as closing operation - to fill holes in the frame and opening operation - to remove noise, followed by dilation - to increase the foreground in the frame image, were applied. The processed image was then used to detect contours using the findContours method from OpenCV with RETR_EXTERNAL as retrieval mode to find the external contours. Thresholding was performed on each contour based on its width and height.



(i-1)th Frame ; ith Frame

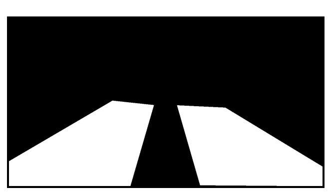
The vehicles are counted by tracking their centroids for a history of 10 frames. If a vehicle's location in the previous frame was outside the region of interest and that in the current frame is inside, the tracking is stopped and the vehicle is said to have passed the road, thereby incrementing the count of the vehicles passed. The vehicles are detected on the basis of the position of its centroid. For every centroid in the current frame, its distance from every centroid in the previous frame is checked. The distance is calculated using a weighted Euclidean norm, that is, translation in the vertical direction is given more importance than the horizontal direction. This is done because vehicles in the video move along the vertical axis.

$$distance^2 = (x_2-x_1)^2/x_{weight} + (y_2-y_1)^2/y_{weight}$$

If this distance is less than a particular value, then that centroid of the current frame is said to be translated from the centroid of the same object from the previous frame. Thus, mapping the same objects from previous and current frame. These centroids are tracked only until the vehicle reaches the exit mask.

DENSITY:

The density of the road is defined as the area of the road occupied by vehicles divided by the area of the road. In this implementation, the density for both the sides of the road was calculated individually. For this, first, the Region of Interest(ROI) was identified.

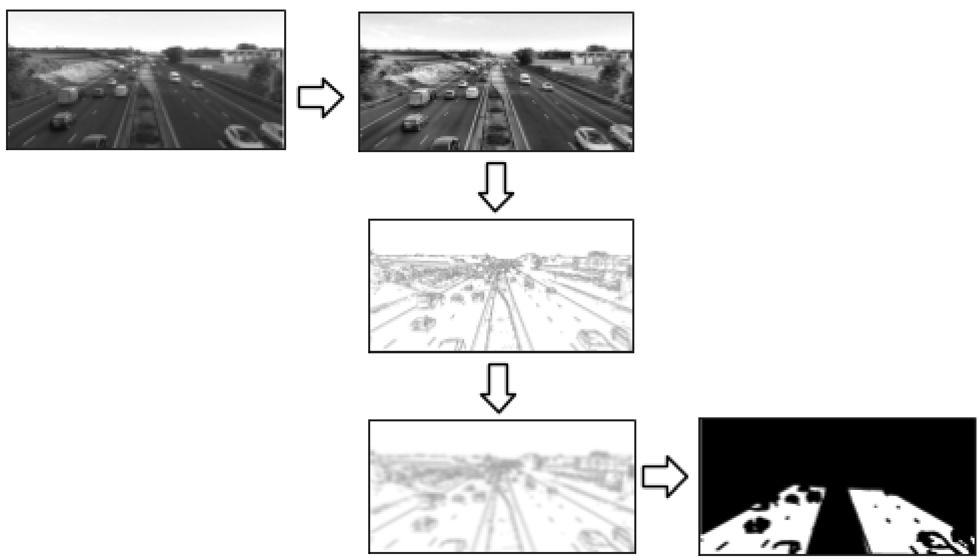


Region of Interest

Thus, objects that pass through the above mask were detected and their area contributed towards the area occupied on the road. After the ROI was fixed upon, each frame was processed as follows:

1. Converted to grayscale
2. Performed histogram equalization on the frame using Contrast Limited Adaptive Histogram Equalization (CLAHE). This was done to adjust the contrast of the frame.
3. Performed Canny Edge Detection to detect the edges of the objects in the frame and inverted the colors in the frame.
4. Applied a bilateral filter; performs smoothening whilst preserving the edges.
5. Applied thresholding on the image.
6. Bitwise AND of the processed image and mask.
7. Calculated the road density as:

$$density \ = \ 1 - (\#free\ points\ (non\ zero)/\#points\ in\ the\ mask)$$



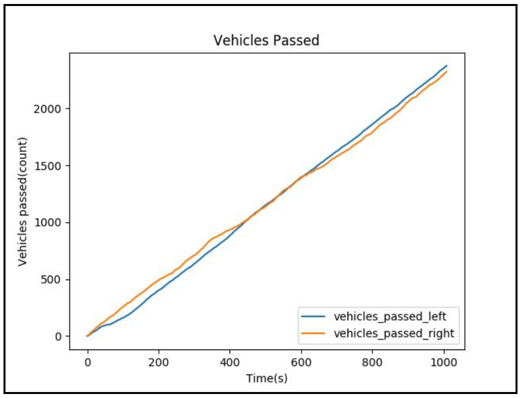
Pipeline of density calculation

3. RESULTS AND DISCUSSION

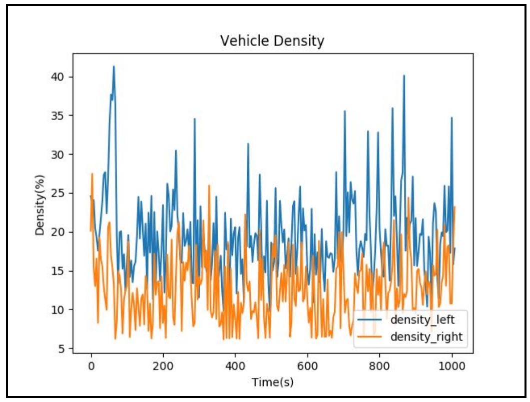
The performance measures were calculated by manual inspection of 500 frames and the following data were observed:

Total vehicles passed = 48 (Left Lane) + 59 (Right Lane) = 107
Total Undetected vehicles = 9(Left Lane) + 5(Right Lane) = 14
Accuracy = 1 - 14/107 = 86.92%
Processing Speed: 10 frames/s

Graphs were plotted for the data gathered from the images. The first graph plots the total number of vehicles passed since time = 0. The second graph plots density vs time. We can use this data to infer various scenarios, for example, traffic jam. The data below indicates that the number of vehicles passed on the left side initially was more than the right side. However, the density of the left road was more than the right. This shows that the left road had a high density and vehicles were moving slowly, indicating a potential traffic jam situation.



Traffic Count



Vehicle Density

Shubhi and Aranta worked on computing the background mask, applying morphological operations on the foreground frame and detecting contours; which were, in turn, is used to track vehicles by their centroids and compute the count of vehicles passed on the road. They also worked on logging the processed data into a csv file which was further used to display the graphs. Rashika and Ashish worked on calculating the density of the traffic at a given frame. Tasks performed were pre-processing the frames and then applying a mask to compute the area occupied by the vehicles on the road. Moreover, they worked on visualizing the results, which included displaying the frames, processing the data and generating an output video by stitching the processed frames together.

Aranta Rokade: Working on this project strengthened my concepts of computer vision. Concepts in theory were better understood by applying them in practice. Thus, the process pipeline was well designed by applying the right operations one after the other. The dataset used was that of a highway; highly static. More experiments could have been conducted by using a dynamic background with noise such as smaller city roads with pedestrians. Thus, separating objects of interest(vehicles) from the other(pedestrians).

Ashish Naik: This project helped me revise my concepts learned through the assignments. This project could have been made faster and more efficient by using computer vision and machine learning based object detection algorithms such as YOLO, R-CNN etc. Getting the algorithm to process real-time frames is a challenge which could have been tested over multiple such algorithms and results could have been compared for respective frame processing speeds.

Rashika Koul: The experience of working on this project was a learning one and it taught me the practical and real-time applications of computer vision techniques. The process of tracking the same vehicle over a sequence of frames was complicated and yet well handled. This project could have been integrated with real-time or simulated traffic control signals and show the functioning of these signals based on traffic count and density obtained by us.

Shubhi Khare: This was an interesting project, which helped in building my concepts in the field of Computer Vision. The vehicle density was evaluated accurately. Given more time, this project could have been extended by using multiple cameras on a large network of roads. Thus, the evaluated results would have been more precise and accurate.

4. CONCLUSION

The Traffic Monitoring System essentially calculates the traffic flow and density at regular timestamps, which along with a threshold value can be used to get a sense of what's "too crowded" or "too slow" respectively. If the density is higher than the threshold and the count of vehicles passed is not increasing, that means there is a traffic jam. If the density is higher than the threshold and the count of vehicles passed is increasing, that means the traffic is slow moving. If the density is lower than the threshold and the count of the vehicles is increasing, that means the traffic is free-flowing. If the density is lower than the threshold and the count of the vehicles is not increasing, that means the traffic is stationary.

This project can be integrated with real-time traffic control signals. Extended by using multiple cameras on a large network of roads. Can be used to monitor the incoming and outgoing traffic in places such as parking lots, malls, tolls, car wash, etc.

5. REFERENCES

[1]https://ieeexplore.ieee.org/document/7562665

[2]https://medium.com/machine-learning-world/tutorial-counting-road-traffic-capacity-with-opencv-998580f1fbde

[3]https://medium.com/machine-learning-world/tutorial-making-road-traffic-counting-app-based-on-computer-vision-and-opencv-166937911660

[4]https://opencv.org/

[5]**Dataset:** https://youtu.be/wqctLW0Hb_0