Intelligent Traffic Monitoring System Using IOT

Mukund Vaidya Satish Matsagar Mayur Mandavkar Prof. D. S. Nath
Department of Electronics and Telecommunication
MAEER's MIT College of Engineering
Savitribai Phule Pune University
Pune, MH, India

***Abstract*- Accurate and timely traffic flow information is important for the successful deployment of intelligent transportation systems. Over the last few years, traffic data have been exploding, and we have truly entered the era of big data for transportation. Existing traffic flow prediction methods mainly use shallow traffic prediction models and are still unsatisfying for many real-world applications In this paper, we are presenting the system by which we are able to get the correct information of real time traffic density without totally relying on the conditions like mobile gps as such. We are able to get an information about the rain as well as pollution level of the desired path. This system will help to increase the precise prediction and understanding of traffic as well as atmospheric conditions of the area.**

***Keywords*- image processing, opening, closing, IOT, adroid application development.**

I. INTRODUCTION

In today's world, The communicators in metropolitan cities have to travel a long distance to reach their destination. As the cities are growing so rapidly the traffic in traffic in the cities is going to be very crutial problem. Also due to the bad traffic management, a lot of man hours are being wasted. Increasing congestion on highways and problems associated with existing detectors has generated an interest in vehicle detection technologies such as video image processing. The user uses the GOOGLE MAPS to check for fastest route to select the path. This not very efficient, Many of the users are concerned about the pollution levels as well as rainfall in their path.

Here we have designed a Raspberry Pi based system which monitors the traffic density as well as the pollution levels and rain fall for a particular area at a specific given time. Before the user starts the commute he can enquire about the best route possible based on traffic density, Pollution and rainfall patterns. Here we are storing the data in a database and when a request comes then depending upon the data for consecutive 7 days user can predict the traffic conditions, pollution levels and rain fall pattern to user. When the user makes an request our software will directly show the current traffic scenario with the help of live images to the user.

II. LITERATURE SURVEY

1. Google Traffic - Google Traffic is a feature on Google Maps that displays traffic conditions in real time on major roads and highways. Google Traffic can be viewed at the Google Maps website, or by using the Google Maps application on a handheld device.Google Traffic works by analyzing the GPS-determined locations transmitted to Google by a large number of mobile phone users. By calculating the speed of users along a length of road, Google is able to generate a live traffic map. Google processes the incoming raw data about mobile phone device locations, and then excludes anomalies such as postal vehicles which make frequent stops. When a threshold of users in a particular area is noted, the overlay along roads and highways on the Google map changes color. Google Traffic is available by selecting "Traffic" from a drop-down menu on Google Maps. A colored overlay appears on top of major roads and motorways, with green representing a normal speed of traffic, orange representing slower traffic conditions, red indicating congestion, and dark red (previously red and black) indicates nearly stopped or stop-and-go traffic. A red and white dashed line indicates a road closure. If there is no data available, an overlay line will not appear.

Users can use the "search" feature to display traffic in a particular area. For example, a user may type into the search box "traffic near Edmonton, Alberta" to see traffic for that city on the map. Another feature uses historical data to show users the "typical traffic" for an area based on the time of day and day of the week. Google Traffic also displays "traffic incidents", such as construction, accidents, and road closures. In addition, Google Traffic can be viewed while using Google Maps' "directions" feature.

2.Google Weather- Weather forecasts are made by collecting as much data as possible about the current state of the atmosphere (particularly the temperature, humidity and wind) and using understanding of atmospheric processes (through meteorology) to determine how the atmosphere evolves in the future.

However, the chaotic nature of the atmosphere and incomplete understanding of the processes mean that forecasts become less accurate as the range of the forecast increases.Traditional observations made at the surface of atmospheric pressure, temperature, wind speed, wind direction, humidity, precipitation are collected routinely from trained observers, automatic weather stations or buoys.During the data assimilation process, information gained from the observations is used in conjunction with a numerical model's most recent forecast for the time that observations were made to produce the meteorological analysis. Numerical weather prediction models are computer simulations of the atmosphere.They take the analysis as the starting point and evolve the state of the atmosphere forward in time using understanding of physics and fluid dynamics. The complicated equations which govern how the state of a fluid changes with time require supercomputers to solve them.

III. IMPLEMENTATION

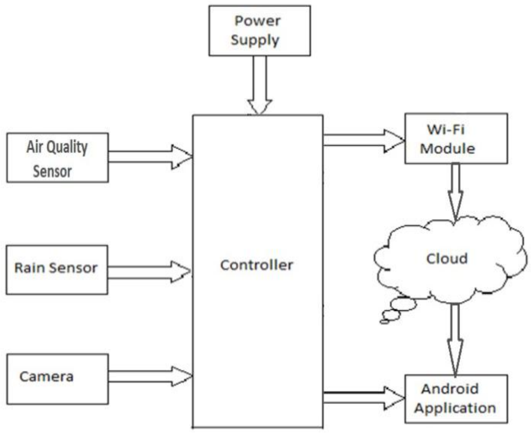


Fig.1 Block Diagram

1. **Hardware Required**

- Raspberry Pi 3 B+.
- Wi-Fi Adapter.
- Camera.
- Air quality sensor.
- Rain-Sensor.

2. Software Required

- OpenCV or
- Matlab
- Python IDE
- Raspbian OS
- Anroid Studio

IV. METHODOLOGY USED FOR REAL TINE IMAGE PROCESSING

1. Input - Here Input is the real time continious video from the camera module interfaced with the Raspberry Pi controller. It is nothing but the continious frames of images.

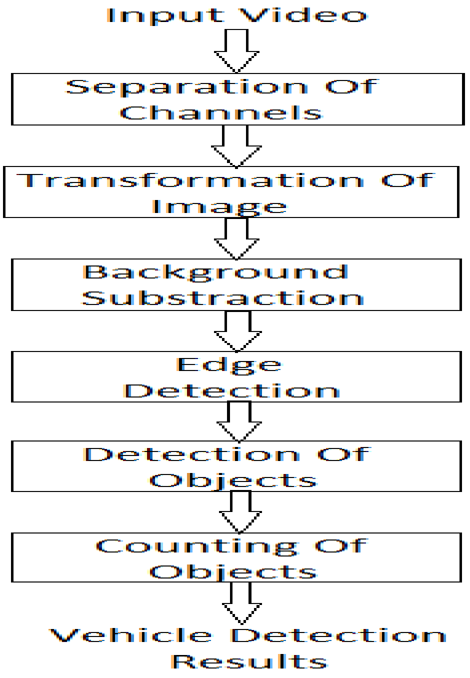


Fig. 2 Flow diagram.

2. Separation of Channels- HSV formats are most suitable for detection, since HSV provide absolute color space of each vehicle. Hence we convert we convert from RGB to HSV where Hue (H) and Saturation (S) has only color information and Value(V) has only intensity information. And the intensities of the channels are processed without altering the color information. After processing, intensity information is recombined with the color information. We want to extract color information from the image so we split the colored image of HSV color space to three different channels and process the pixel intensity value in each channel. Splitting an image in its color channel decreases the time complexity of algorithm.

3. Transformation of Image - Morphological processing is basically collection of non-linear operations related to the shape or features of an image. It depends on relative ordering of pixel values and not on their numerical values. Morphological operations are usually performed to remove noise, isolating the individual elements and joining separate elements in an image and also for finding intensity bumps or holes in an image.

The structuring element is located at all possible positions in an image and it is analyzed with the corresponding proximity of the pixels. Morphological operation differs in how we carry out this comparison. Structuring element is also called as kernel and consists of a template specified as the co-ordinates of a number of distinct points. Consider an example of 8X8 image as shown in the fig below where the structuring element is of 2X2. White color box represents zero pixel value and grey color box represents non zero pixel value. The structuring element is slide across the image and examines the image with the structuring element values. The label _A‗ indicates the structuring element not fitting in the image, 'B‗ indicates structuring element intersecting the image and _c‗ indicates structuring element fitting in the image.

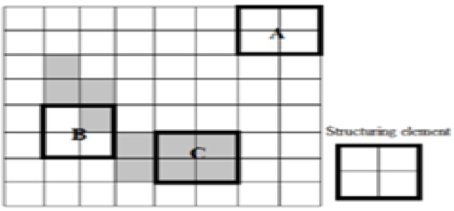


Fig.2. Illustration Of Structuring Element.

The fundamental morphological operations are erosion and dilation.

3.1. Erosion - Erosion operation is to gradually decrease the boundaries of region of foreground pixel, the area of foreground pixels shrink in size and strips away a layer of pixels from an object where the holes within those areas become larger. It considers two pieces of data, one is the input image and other is structuring element. For each of the foreground pixel, structuring element is superimposed on the top of the input image. So that origin of structuring element coincides with the input pixel co-ordinates and the pixels which are not completely enclosed by other foreground pixels are removed, these pixels lay at the edges of foreground regions. So it results in shrinking of foreground region where holes inside a region grows. It results in removal of small spurious bright spots in images.

3.2. Dilation - Dilation operation gradually enlarges the boundaries of regions of foreground pixels, thus area of foreground pixels grow in size where holes within these region become smaller. This operation considers two input data sets, one is the input image and other is structuring element, it considers the background pixel in the input image as input pixel, structuring element is superimposed on the top of the input image.

So that structuring element coincides with the input pixel co-ordinates and if there is an intersection of at least one pixel between structuring element and foreground pixel then the input pixel is set to foreground. This operation sets the background pixels to the foreground pixel value if it has a neighbor foreground pixel and these pixels lay at the edges of foreground regions. So it results in in growing of foreground regions where the holes inside a region shrink and erosion is used in filling the small spurious holes in images.

3.3. Opening and closing - An opening is defined as erosion succeeded by dilation with the same structuring element. It considers two input data sets, one is the input image and other is structuring element.Let _f‗ be the input image and _õ‗ be the structuring element. _Ó‗ indicates opening operator, _Ê‗ indicates erosion and _Ð‗ indicates dilation. The opening operation is represented in equation (1)

Ó (f, õ) = Ð (Ê (f, õ), õ) -------------- (1)

This operation retains foreground regions that have identical shape of structuring element and eliminates other regions of foreground pixels.

Closing operation is performed in revere of opening. It is defined as dilation succeeded by erosion with the same structuring element.

It considers two input data sets, one is the input image and other is structuring element.

_Ç‗ indicates closing operator. The closing operation is represented in equation (2).

Ç (f, õ) = Ê (Ð (f, - õ),- õ) --------------(2)

This operation retains background regions that have identical shape of structuring element and eliminates other regions of background pixels.

3.4. Tophat and Bottomhat Transformation - Tophat transformation is performed by subtracting the original frame from the opening operator. The tophat transformation is represented in equation (3). Let I(x,y) be the original frame. _Ť‗ indicates tophat transformation.

Ť(x,y) = Ó (f, õ) - I(x,y) --------------- (3)

Bottomhat transformation is performed by subtracting the closing operator frame from the original frame. The bottomhat transformation is represented in equation (3). _Ḃ‗ indicates bottomhat transformation.

Ḃ(x,y) = I(x,y) - Ç (f, õ) ---------------- (4)

fter the transformation method, channels are merged and further processed with the combined channels.

4. Background subtraction - Moving foreground objects are isolated from the static noisy background using background subtraction method . It is the most important technique used to extract foreground objects from the image. It performs frame by frame subtraction which results in the removal of static noise in the video. The objects are segmented using Gaussian mixture model of background subtraction method. The proposed system extracts the foreground region by background subtraction which gives the region where the vehicle exists in the scene. A mask is created for this region, convolved over the input image to obtain the vehicle and subtracts the background. Figure 3 shows the result of foreground object extraction.

5. Edge detection - For extracting low level feature edge, classical canny edge detection is used which is less sensitive to noise and reduce the noise by smoothing. Two thresholds T_{high} and T_{low} are set and avoid streaking problem. It provides good localization by considering gradient orientation. Once the edges are extracted from the object, blurring of the extracted object is performed so that superfluous interior edges are not considered in further processing

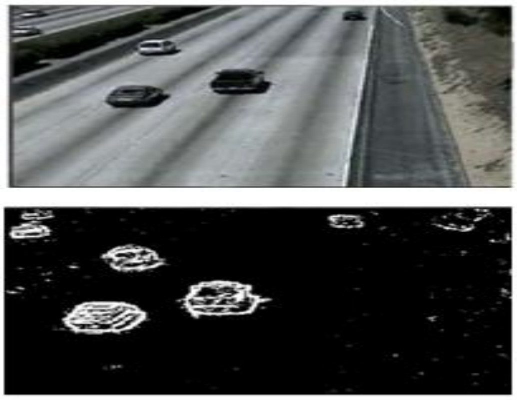


Fig.3. Input Image and Object extracted Image.

6. Object Detection- Grouped object which is called as blob can be identified by using eight way connectivity. In this method the pixels are scanned across the image and foreground pixel is checked with the eight neighboring pixels, whenever the input pixel is connected with the foreground pixel it checks for the label of that pixel if the neighboring foreground pixel is already labeled, then it assigns the same label to the input pixel or it assigns unique label to the input pixel. This process is carried out till it reaches the last pixel of the frame.

Consider an image of 8X8 as shown in the figure below, which consist of three objects which are uniquely labeled as L1,L2,L3. Gray color indicates the foreground pixel i.e _0‘ and white color indicates background pixel i.e _1

					L1	L1	
						L1	
	L2				L1	L1	
	L2	L2				L1	
	L2	L2					
				L3	L3		
					L3	L3	
						L3	

Fig.4. Object detection

Since the grouped object identified has irregular shape convex hull is used to define a regular shape. It considers the exterior points of the blob and polygon is constructed from these points.

7. Object Counting - Size of the blobs are calculated which is based on the calculating the number of occurrence of the foreground pixel in the blob. Size constraint is enforced on the object so that small objects which can act as noise can be removed. Bounded rectangle is drawn on the objects which confines the detected vehicle and a virtual line is drawn to count the number of vehicles, the centroid of the bounded rectangle is calculated. When the centroid intersects the virtual line, count is incremented.

V. IMPLEMENTATION RESULTS

Experimental results of the proposed algorithm is demonstrated in this section which is conducted on different datasets of aerial videos which is taken under different camera angles and heights. Figures in this section demonstrate the detected vehicles with the bounded rectangle on the vehicles and the red color line in the figure is a virtual line which is drawn to count the number of vehicles. Fig (1) shows the detection of all the vehicles in the complex environment which confines the accuracy of the proposed algorithm in the complex environment. Fig (5) shows that the bounded rectangle on the detected vehicle remains until the vehicle goes out of the frame which confines the stability of the algorithm. Time complexity of the algorithm for processing each frame is less which is 15 frames frames per second since the numbers of instructions used are less and the computational complexity of the algorithm is less which makes use of basic techniques like morphological operations, etc.

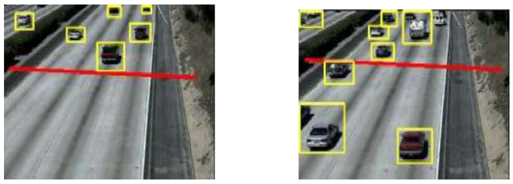
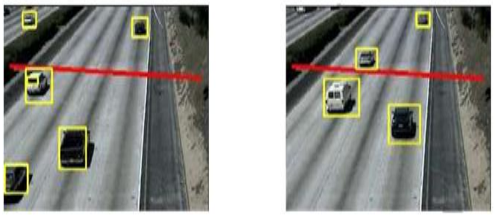


Fig. 5 (a) shows that the bounded rectangle on the detected Vehicle.



(b)

(c)

Table shows the results of the algorithm with number of vehicles detected by the system, number of vehicles not detected and the vehicles detected manually, precision is calculated using the equation

P= M/S

Where,

P = Precision

M = No of vehicles counted manually

S = No of vehicles counted by the system

Table 1 Tabular representation of result.

Scene	No of vehicle Counted manually	No of vehicles counted by the System	Precision
1	6	6	1
2	9	8	0.89
3	5	5	1
4	4	4	1
5	5	5	1
6	5	5	1

VI. CONCLUSION

In this paper, we have presented the project by which we are able to get the correct information of real time traffic density without totally relying on the conditions like mobile gps as such. We are able to get an information about the rain as well as pollution level of the desired path. This system will help to increase the precise prediction and understanding of traffic as well as atmospheric conditions of the area. The proposed system is an effective tool for driving assistance and weather detection as it gives very clear and precised count of vehicles as shown in the simulation results. We can further modify it by using night vision technology and fogg removal system for more clear and precise result in any weather coondition.

REFERENCES

[1] R. Lin, X. Cao, Y. Xu, C.Wu, and H. Qiao, Airborne moving vehicle detection for urban traffic surveillance,‖ in Proc. 11th I nt. IEEE Conf. Intell. Transp. Syst., Oct. 2008, pp. 163–167.

[2] R. Lin, X. Cao, Y. Xu, C.Wu, and H. Qiao,Airborne moving vehicle detection for video surveillance of urban traffic,‖ in Proc. IEEE Intell. Veh. Symp., 2009, pp. 203–208.

[3] J. Y. Choi and Y. K. Yang, ―Vehicle detection from aerial images using local shape information,‖ Adv. Image Video Technol., vol. 5414, Lecture Notes in Computer Science, pp. 227–236, Jan. 2009.

[4] Jin-Cyuan Lai ;Shih-Shinh Huang ; Chien-Cheng Tseng , ―Image-based vehicle tracking and classification on the highway‖, Green Circuits and Systems (ICGCS), 2010 International Conference, 21-23 June 2010

[5] Zhong Qin, Guangzhou, ―Method of vehicle classification based on video‖, Advanced Intelligent Mechatronics, 2008. IEEE/ASME International Conference, 2-5 July 2008.

[6] J. F. Canny, ―A computational approach to edge detection,‖ IEEE Trans. Pattern Anal. Mach. Intell., vol. PAMI-8, no. 6, pp. 679– 698, Nov. 1986.

[7] Omar Javed, Khurram Shafique, and Mubarak Shah. Appearance modeling for tracking in multiple non-overlapping cameras. In CVPR, 2005.

[8] A.G.A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W.S. Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In CVPR, 2006.