IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

VEHICLE SPEED TRACKING USING OPENCV

J. RAJASEKHAR[1], K. YASHWANTH REDDY[2], M. RUCHITHA REDDY[3], C. GOWTHAMI[4]

Department of Computer Science and Engineering, St. Peter's Engineering College, Hyderabad, Telangana.

Abstract: **The continuously increasing number of on-road vehicles has put a lot of pressure on road capacity and making traffic management difficult and giving way to problems like congestion, collisions, and air pollution, among others. These problems have significant impact on our daily lives. A strong, healthy and efficient traffic management system is needed to reduce the effect. Apart from these problems related to vehicle traffic, Also, various statistical parameters, such as the average number of vehicles on the road at a certain time, and the state of congestion, can be studied, which can provide some for managing the highway.**

I. Introduction

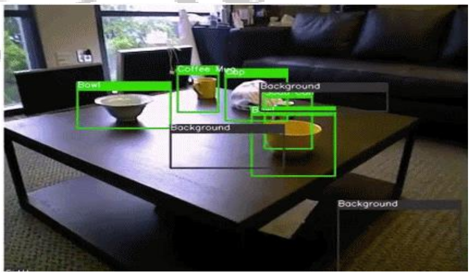
Now a days, someone has to monitor all the traffic, sometimes it takes a huge man power to identity any unusual activity, the person has to check all the resources to identify a particular event. Now think about the following.

If the system could track the activity and report to us. Something like traffic management system, where the traffic is being monitored by the system automatically. This system can reduce a lot of man power and easy to keep track of vehicles. This project is similar to such case. Where traffic is managed by the computer itself. This system is efficient to monitor traffic and vehicle speed.

II. Moving Objects

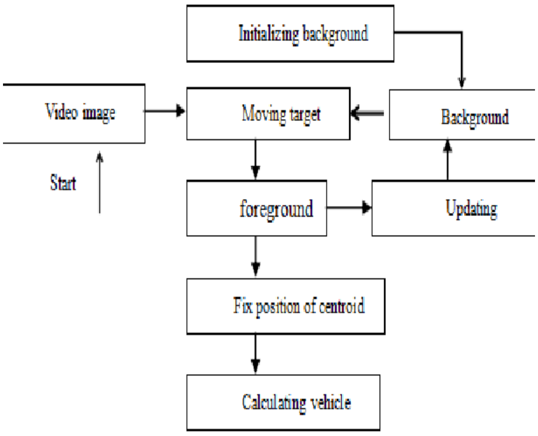
When dealing with the video game the experience goes to whole new level. The complexity rises to top level and so do rewards. We can perform super useful high-value tasks such as surveillance, traffic management, fighting crime, etc. using object detection algorithms.

The subtasks performed in the object detection such as finding the number of objects, the relative size of the objects, distance between the objects. All these sub-tasks are important. These are the sub tasks we can perform as they are the important part of solving the problems in real world case.

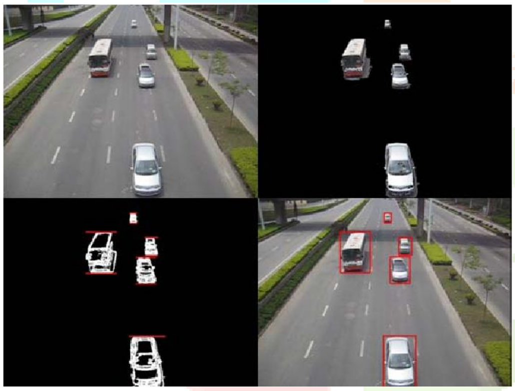


III. Use-Case

There are a number of sub-tasks we can perform in object detection, such as counting the number of objects, or finding the relative distance between the objects. All these sub-tasks are as they contribute to solving some of the toughest real-world problems.



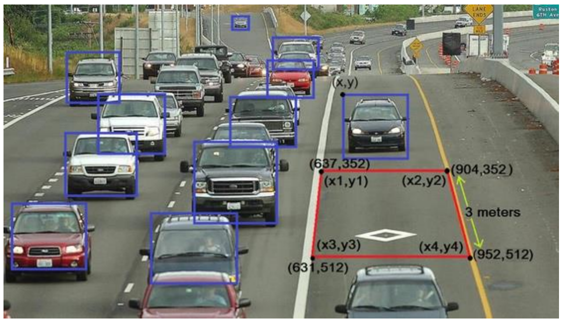
When the video is inserted, it undergoes many mathematical operations, the first step in this process is to determine the moving target and focus on the foreground. In the way the background is initialized. The background is updated throughout the iterations and the fixed position of the centroid is obtained. This point is used as the main point to perform the caliculations.



IV. Proposed Systems

Speed= distance/time. We all know this little formula. Its time to implement code. First we need to know the distance then we need to draw the polygram.

In this system we determine particular area, whenever the vehicle enters into this area, the algorithm will detect the vehicle has moved into this region and marked as the incoming moving object. The area is only limited and fixed, so the calculations are applied only in this region. The vehicle is marked until it leaves the region. Once the next vehicle enters the same thing repeats, this process is endless and effortless. The formulae will only work on a fixed region, the camera needs to be stable and fixed to a point.



The objective of this system is to capture and mark the coordinates of the moving vehicles and highlight it in the video. Now, we can see the vehicle move with the speed tagged.

The box is created around the moving object. This box is usually determined with green square.

V. Calculations

Distance between vehicle and starting point measured in kilometer

$$\text{Distance} = Df * (D / D_x) * (P_n - P_0) \quad \ldots(1)$$

Time that vehicle spent in order to move to P_n in unit of hour

$$\text{Time} = Tf * (t_n - t_0) \quad \ldots(2)$$

Vehicle speed measured in format of kilometer per hour

$$\text{Speed} = \text{Distance} / \text{Time (Kilometer per Hour)} \ldots(3)$$

Where D is the real distance between two marking points (start point and end point) measured in meter

D_x is the distance between two marking points measured in pixels

X is the width of the video scene measured in pixels

Y is the height of the video scene measured in pixels

P_0 is the right most of the vehicle position at time $t = 0$ measured in unit of pixels

P_n is the right most of the vehicle position at time $t = n$ measured in unit of pixels

t_0 is the tickler (timestamp) saved at time $t = 0$ measured in unit of milliseconds

t_n is the tickler (timestamp) saved at time $t = n$ measured in unit of milliseconds

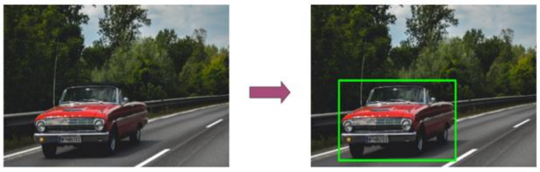
Df is the distance conversion factor from meter to kilometer, which is (1.00/(1000.00*60.00*60.00))

Tf is the time conversion factor. In this case, the conversion is from millisecond to hour, which is (1.00/1000.00)

VI. Essential Concepts

1. Framing Differencing

Video is a set of frames that are put together in the right sequence to generate output. So, when we see an object moving in a video, it means that the object is at a different location at every consecutive frame. If we assume that apart from that object nothing else moved in a pair of consecutive frames, then the pixels difference of the first frame from the second frame will highlight the pixels of the moving object. Now, we would have the pixels and the coordinates of the moving object. This is how the frame differencing works.



2. Image Thresholding

In this method, the pixels value of a grayscale image are assigned one of the two values representing black and white colours based on a threshold. So, if the value of pixel is greater than a threshold value, it is assigned one value, else it is assigned the other value.

The major part of the unwanted area is highlighted and removed. The highlighted edges of the notepad are not visible anymore. The resultant image can also be called as a binary image as there are only two colours in it. In the next step, we will see how to capture these highlighted regions.

3. Finding Contours

The contours are used to identify the shape of an area in the image having the same colour or intensity. Contours are like boundaries around regions of interest. So, if we apply contours on the image after the thresholding steps, we would get the following results.

The white regions have been surrounded by greyish boundaries which are nothing but contours. The coordinates of the contours are being fetched. This means we can get the location of the highlighted regions.

It is also important to have more contours, in case of the maximum area and more incoming objects, the multiple highlighted regions and each region is marked by a counter. Its better to have a more number of contours as possible in case of a backup.

In the image above, there are still some unnecessary fragments of the white region. There is still scope of improvement. The idea is to merge the nearby white regions to have fewer contours and for that, we can use another technique known as image dilation.

4. Image Dilation

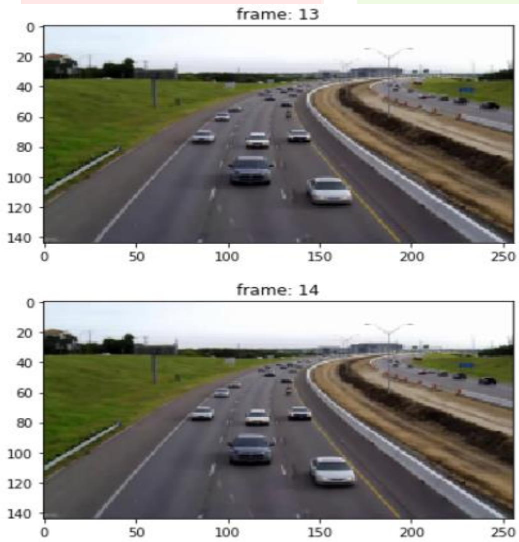
This is an operation in which the dilation expands the pixels on an image wherein a matrix. This is passed over the entire image. Just to give you intuition.

So, let's apply image dilation to our image and then we will again find the contours.

It turns out that a lot of the fragmented regions have fused into each other. Now we can again find the contours.

It turns out that a lot of the fragmented regions have fused into each other. Now we can again find the contours.

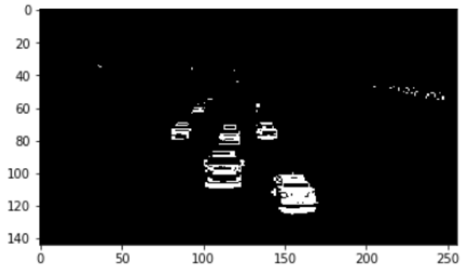
The contours are the important in plot the area around the moving object. We can also observe that the contours form the surrounding the object. We have four contours, these are also known to be four candidate contours.



5. Image Preprocessing

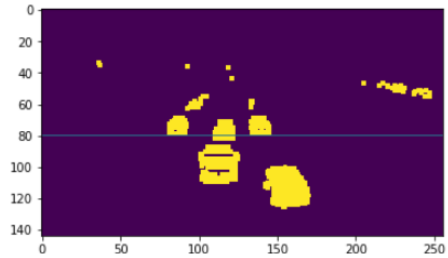
In image preprocessing, the image is prepared to be operated on further calculation. In this image, the vehicles in undesired white regions are noise and look more promising. However, the fragmented bits are highlighted in the region.

Let's apply image dilation over this image.



The moving objects have more solid highlighted regions. Hopefully, three is the limit for the number of contours for object in the frame.

We need to determine the fixed area, so whenever any vehicle enters into this area, the system will detect the object.



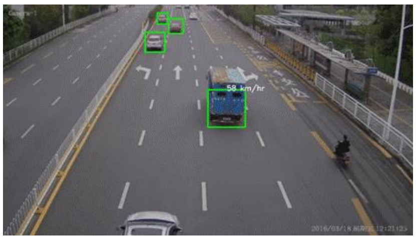
Vehicle detection zone in this senior is below the horizontal line. This is the zone where we detect the moving objects. Since we have to find only those contours that are present in the detection zone, we will apply a couple of checks on the discovered contours. The top-left y-coordinates of the contour should be greater than or equal to 80.

The contour area should be greater than or equal to 25. cv2.contourArea() This function is used to find the contour area.

6. Conclusion

In this paper, we have presented the vehicle speed tracking using python and opencv. In this project we have used the video footage from the highway and applied the algorithm to determine the speed of the moving vehicle. The speed of the vehicl is generated through the video file. This output file consists of video from raw footage and the speed is of the individual vehicles is highlighted in the green square box.

Input



Output

7. References

Vehicle speed detection using image processing – IEEE, Vehicle detection model – AnalyticssVidhya, Vehicle speed tracking – IEEE, Image Processing – opencv.