Digital Image Processing

Traffic Congestion Analysis

Dated: 28-11-2012

By:

Romil Bansal (201207613)
Ayush Datta (201203003)
Rakesh Baddam (200930002)
Abstract

Traffic estimate from the static images is the key issue for automating traffic light controls. In this work, we address the problem of estimating the traffic congestion and vehicles density on the roads. First, we compared most of the image segmentation techniques and currently available methods in estimating the traffic density. Later, we tried to modify some of the algorithms to improve the congestion estimate. Lastly, we tried to club most algorithms to provide a little estimate for traffic estimate from static images.

I. Introduction and Related Work

In today's metropolitan areas, highway traffic congestion occurs regularly during rush hours. In addition, nonrecurrent congestion often takes place as a result of imbalance traffic lights controls like the busy road on the junction receiving equal time slot for green lights as the empty roads on the same junction. Although the traffic monitoring cameras are installed at various junctions but most of them usually detect just road faults or break of rules. If those cameras are also used for traffic monitoring, the congestions could be avoided and lots of resources could be saved.

Many of the papers try to address the problem of estimating the traffic on the road. But most of them either used the pre-learnt models (which are made by training the systems over the pre classified samples and known shapes of object first before using them for the general classification purposes) or used the temporal data available overtime to estimate the cars and background. Junwen Wu used the wavelet transform to estimate the features of the current images and try to classify cars using PCA or LDA techniques from pre-learnt models. The problems with these techniques are that they fail to classify if some untrained examples appear like accidents, change in weather conditions etc. The other works mostly uses the temporal data available over time through these cameras to estimate the traffic.

II. Implementation and Observations

The easiest part of from estimating the cars and traffic using the videos is that you can make use of motion segmentation techniques to detect the background and moving cars. With the detection of cars you can estimate the traffic at a place. We also build a system that estimates the positions of the cars and the speed/traffic from these temporal data (images over time or videos).

a. Cars Detection using Extended Maxima/Minima Transform over a video

First we tried with estimating the cars with matlab's extended maxima transform [1] and extended minima transform. This function tries to estimate the maxima and minima in the given frame. From the given frame, we calculate the extended maxima transform to calculate the bright cars in an image. Then we calculate the extended minima transform for black cars in the image. Then using morphology we remove the stray pixels and display the bright regions as cars.

b. Edges in Images of same place overtime
The problem with the above approach is that it can only find maximal and minimal regions only i.e. cars that are mostly white or black can only be found. But different colour cars are little difficult to find. Also some places that are in dark are also counted as cars. Shadow regions also pose some difficulty in detecting cars by above method. So we tried another algorithm.

Paper [3] states the following methods for detecting the traffic congestion at the signal.

a) Detect the edges from in an image with low traffic using Prewitt filter.

b) Take the power law of the traffic image to remove all the dark regions like shadows etc.

c) Detect the edges from the newly obtained image.

d) Subtract the two images.

e) The amount of white edges in the new image represents the amount of traffic on the road.

1) We tried to implement this method and test it on our own images. The algorithm seems to work fine for only some of the images especially those that are taken from the top and having clear boundaries. For most of the other images the edges are not complete and none of the car is detected completely. Some of the results after applying Sobel edge detection are shown below.
Edge Detection Using Prewitt

Original Image-2
The whole algorithm depends upon the quality of edge detections, since they are being used for subtraction and vehicles density estimation. Prewitt is not giving satisfactory image quality. So we tried to modify the edge detection algorithm to canny edge detection.

2) Since Prewitt is not giving us good results we tried to modify the method and try it out with the canny edge detection. The results are as follows
3) The results are surely better than the Prewitt edge detection but canny is giving many details even which are not required. We can clearly see that there are many stray edges of trees. This might be due to the wind blowing that changes the trees orientation little bit leading to lots of stray edges that are also being detected as edges. So to reduce those edges we tried to threshold the canny edge detector (to reduce too much details of cars as well) and also take only those edges which are not surrounded by any other edge within the small neighbourhood of it in the background image.

The results are as follows:
These results seem satisfactory to us. So we continued with the implementation of the remaining paper to see if the overall results improve or not. We took the reference image when the traffic was not too high and another test image and implemented the algorithm over it. The results after implementation are as follows:

Disk size = 3.
c. **Work b over videos**

Since the above described algorithm gives us decent results, we tried to implement it over the videos. The advantage of the videos over the previous approach is that we can estimate the background very easily by averaging over a large period of time. The background estimation accuracy increases as the time duration for capture increases. This also takes into account the small moving objects like trees etc. So we have nearly ideal image to subtract with. Many other methods like using kalman filters [4] to detect the background can also be used over temporal data to detect background and improve efficiency of the above algorithm. The result of implementation of one such method could be found here.
Also since we have lots of consecutive frames, we can also estimate the traffic congestion by estimating the direction of change in flux. Many good methods are being developed using this technique to estimate the background and to estimate the traffic density in the video.

d. Region Growing
For extending the method we started to think how humans try to identify cars or objects (assuming they don't know shape of cars) from the given image. The best way is based on the colour segmentations. The number of different colours we see, we could make out how many objects are there. This leads us to implement the region growing algorithm over the image. We could start with any random seeds and select those areas whose overall area are within a threshold or form some cluster, not randomly distributed over the image. The number of seeds/area covered may provide us the vehicles density at a place. Here are the few results after applying region growing over the image.
The Arrow shows the position where we are applying region growing.
Here in second image we apply region growing on white car shown by arrow
The above figure shows that it is difficult to decide the threshold value where we want to stop the region growing. For one seed it is giving perfect results but for same threshold but different seed it is giving complete image. So due to difficulty of selecting the threshold in this method we tried to explore some different method.

e. MSER based Segmentation

Maximally stable Extremal Regions (MSER) method is used as a method of blob detection in images. We tried to use this method with varying values of delta to detect the cars in the image. The results are as follows:
MSER provided us with good results. It segmented out the cars well. But how do we know the traffic density from it? We tried to find the number of ellipsoidal curves/area covered by ellipsoidal curves.
If this value is above threshold then the traffic is high else not. The intuition here is that if the road is empty or camera is able to capture only one big truck then the curve will fit the whole road. So there will be just one ellipse with large area thus decreasing the value of the above function below threshold. On the other hand if the road if there are many ellipse of small area then the road is filled with many cars occluded behind one another. This can also be seen from the above images.

But sometimes MSER is also segmenting out the regions that are not the vehicles. For example, look at the following image.
We can clearly see that although there are not many vehicles on the road in the original image, MSER algorithm found out many segments, many corresponds to the surrounding background regions or the marks on the roads. So although MSER algorithm can provide us some intuition about the traffic, it may not be correct always. If we have some prior knowledge about the background, then that can be incorporated in the algorithm to capture regions of interest. Otherwise MSER may also provide wrong information.

f. Hough Transform

Though the above results gave us the rough idea of the congestion over the road, but still most of them fails in some conditions or others. We saw that most of the methods fail in detecting the edges of the vehicles only, while many other fails due to small marks of the neighbourhood or road marks. To reduce the effects due to first problem we tried to implement the Hough Transform that can capture even broken edges.

a) We saw that the traffic cameras are mounted either at the front (or back) at the signal or at top. So mostly either the front or rear rectangular mirrors are visible or the top portions of the cars are visible. All of these form a rectangular structure. So counting number of rectangular structures might give us intuition of how many and where are the vehicles present. Hough transform is a nice way to detect the lines in the image. So we tried to find the lines such that 4 lines form a rectangle. By this we may be able to remove the background noise.
Original Image (Sample -1)

Edge Detection Using Canny
b) The Hough transforms method results in too many small broken lines. This was due to the fact that the cars have many edges and most of them are not complete. There are too many broken edges to form a so called rectangular structure presumed in the previous case. So we tried to complete those edges first by applying morphological image processing. We closed the image using the disk of size 2. This completes the edges. Now we tried to run the line detection through Hough transform over this newly obtained image and the results are as follows:
Applying Morphology before Hough
c) The major problem with Hough is that it was giving lots of stray unrequired edges other than the car edges. On looking closely we say that car consists of many horizontal edges that are captured nicely by Hough along with some other edges. So we tried to modify the Hough

$$\begin{align*}
\theta \\
\rho
\end{align*}$$

Final Output of Hough Transform after applying Morphology
transform in so that it gives us only horizontal edges and threshold those edges whose
length \( > \sqrt{3000} \) and also fills the region which has many horizontal edges within \(+/-15\)
pixels of it. Assumption is that the Hough transform will fill up the cars region with edges
that can be detected easily. The amount of regions filled in this way can give us the overall
traffic density at a place. Hough which works only on horizontal lines + region filling is
applied in this method to show exactly the car.
Result-1

Original Image Sample-2
Image after edge detection and Morphology

Hough transform
These output seems decent that the previously applied algorithms. This method can be used to classify the regions into high traffic density or low traffic density. Also with the help of only horizontal edges all the road marks, poles and most of the buildings edges are not included. Further morphology can be applied over these images to extract cars and further filter out some stray edges that might appear due to shadows etc.

d) We tried to extend the above approach to the videos. Videos provide us with good estimation of the background. The first method used although gave us the intuition where the vehicle is present; we are unable to highlight it properly. Now after estimating the background and subtracting it from the image, we applied the above algorithm. The results seem satisfactory over the applied data set with not too high vehicle density. Below are the results for a frame captured with images at various steps.

Original background Image

Estimate cars in this frame of video
Canny Background

Canny frame

Subtracted Image

Removing stray Edges

Final car detection by closing
Result 2:

Estimated Background
Removing stray edges: Opening
III. Classification

To classify the data into low, medium or high density we took the image and calculated the percentage of it being covered by the horizontal lines. For <30% the traffic density is low, for >70% traffic density is high else the traffic density is medium.
IV. Conclusion

We compared many currently used Image processing/segmentation techniques for finding the traffic density at a place using static images only. Some of the methods seem to work fine for the low traffic density but failed to work at high density, while many others fail to work if the background is not subtracted. We proposed an approach that can work to some extent in predicting the traffic density for single-lane traffic. The algorithm is tested over various images and also videos where traffic is mostly moving only in one direction. It was able to correctly determine the traffic density in most of the cases.

V. Future Work

What if the cameras are placed at the junction? Example: Look at the image below.

Now just detecting horizontal edges won’t work. The traffic is moving in multiple directions. This problem cannot be solved by the proposed algorithm. So we need more robust approach that can handle such cases.

Another area to work on is the shadow regions handling i.e. how to handle shadows, shadow of stationary objects, moving objects, one car over another etc. We didn’t looked much into that problem as we are concerned mostly with high traffic image analysis where shadows usually don’t matter much as objects are highly clustered.
References

1. Detecting Cars: Math works Image Processing Toolbox

2. Vehicle Detection in Static Road Images with PCA-and-Wavelet-Based Classifier IEEE Intelligent Transportation Systems Conference Proceedings - Oakland (CA), USA By Junwen Wu, Xuegong Zhang and Jie Zhou

3. Implementation of image processing in real time traffic light control 3rd International Conference on Electronics Computer Technology (ICECT), 2011 Author(s): Choudekar, Ajay Kumar Garg


5. Data set used for traffic detection : vip_traffic.avi
   http://ee.cuhk.edu.hk/~xgwang/MITtraffic.html

Some Other References

6. Robust Multiple Car Tracking with Occlusion Reasoning Jitender Malik

7. VLFEAT tutorial

8. Efficient Image Gradient Based Vehicle Localization IEEE transaction on Image Processing


10. Shadow Inpainting techniques

11. Robust Tracking of Position and Velocity With Kalman Snakes Natan Peterfreun


13. General Road Detection From a Single Image

14. ShiftingWeights: Adapting Object Detectors from Image to Video


17. Mixture Kalman Filter Based Highway Congestion Mode and Vehicle Density Estimator and its Application Xiaotian Sun, Laura Muñoz, and Roberto Horowitz

18. Video Image Vehicle Detection System for Signaled Traffic Intersection

19. On-Road Vehicle Detection: A Review Zehang Sun, Member, IEEE, George Bebis, Member, IEEE, and Ronald Miller

20. Robust Real-Time Lane and Road Detection in Critical Shadow Conditions* Alberto Broggi

21. Color-Based Road Detection in Urban Traffic Scenes

22. Vanishing point detection for road detection

Terms to refer:

1. Mean Shift (also see mean shift clustering), Gabor Filter, Histogram of Gradients, Ada-Boost