Traffic Sign Detection and Reading using MATLAB

Prithviraj Kadiyala, School of Electrical and Computer Engineering, OU
prithvirajkadiyala@ou.edu

***Abstract*— The main overview of this paper is to use image processing technique to detect Traffic signs in an Image and Read them to display them to the driver. MATLAB Image Processing toolbox is used in this project. It involves image acquisition, pre-processing such as thresh-holding, grey scale conversion, color and shape analysis for detection of the signs and finally read them to the driver. This project also discusses the problems in which the sign cannot be processed to save the information without white washing it with the background. Work also must be done with the robustness of the system because time is of the essence while driving and the faster the information is provided, the more time the driver must react to the surroundings.**

***Index Terms*—Traffic Sign, Image Recognition, Connected components**

I. Introduction

Road Signs play a vital role in giving information to the drivers about the surroundings and warnings in case of any dangers. But while driving it is difficult for the driver to focus on the signs than the road. Hence implementing a system which could read the signs and read them to or display them to driver on his dashboard for safer driving conditions. Using MSER, CC and Stroke Width algorithm we could read and display the correct info to the driver without much distraction.

When driving on congested roads, it's sometimes difficult to keep your eyes everywhere at once. Checking the road ahead, oncoming traffic, what's behind you, all while trying to maintain your speed can sometimes become quite distracting. Car firms realize this and are constantly looking to introduce new technologies, which make things easier. With the introduction of Traffic Sign Recognition systems, the chances of not noticing a change in speed limit, or the warning of a potential hazard ahead have been vastly reduced. In the latest of our technical pieces, we explain how this clever system could benefit you. As with most of the best technology, it's a relatively simple principle which has the potential to reduce the hassle for road users. Essentially, the system consists of a forward-facing camera, which scans the road ahead for traffic signs. This camera is connected to character recognition software, which then makes a note of any changes described by the signs and relaying it onto the car's instrument panel. The information stays there until any change occurs, so if a driver is unsure of the current speed limit all they must do is check the information that the car has noted. (Carwow, 2018)

Traffic-sign recognition (TSR) is a fundamental component of a driver assistance system. It enhances safety by notifying the driver of speed limits or possible dangers such as imminent road works or pedestrian crossings. The traffic signs have two utmost properties are their 'Color' and 'Shape' using which the real-time system becomes capable to detect and identify the Traffic Signs e.g., red-rimmed circular prohibition signs, triangular warning signs, and blue information signs. The simplified pictograms make them easily perceivable and understandable. The detection and recognition of these traffic signs may face one or more of the following difficulties:

(i) Faded signs
(ii)The presence of obstacles in the scene
(iii) Similar objects in the scene or similar background color
(iv) Damaged signs.

To overcome these difficulties many techniques have been suggested. Traffic signs have been designed so that they are easily distinct from driving environment. The color for traffic sign are chosen in a way such that, it serves different purposes and is also distinguishable for the driver while driving. The signs are characterized by fixed shapes like triangle, circle, octagon, and rectangle. The combined feature of shape and color are used by the driver to distinguish a traffic sign. Hence an autonomous system also uses the same principle of the color and shape of the traffic signs. With respect to the road the traffic signs are positioned at well-defined locations so that the drivers can expect the position of the sign. The road sign may contain text as a string of characters, picture or both to represent the meaning of the sign. They are characterized by using fixed text fonts and character heights. There are many traffic signs in India categorized as WARNING, COMPULSORY, REGULATORY and INFORMATORY. This makes a total of 92 traffic signs all together. These signs are mainly characterized by color and shape. Figure 2 shows the different types of Indian traffic sign. (Karthiga.PL, 2016)

The sign which is placed at the side of roads to impart information to road users is known as road signs or traffic signs. There are four types of traffic signs that are shown in the traffic code:

a) warning;
b) prohibition;
c) obligation; and
d) informative.

Depending on the form and the color, the warning signs are equilateral triangles with one vertex upwards. They have a white background and are surrounded by a red border. Prohibition signs are circles with a white or blue background and a red border. Both warning signs and prohibition signs have a yellow background if they are in an area where there are public works. To indicate obligation, the signs are circles with a blue background. Informative signs have the same color. Finally, there are two exceptions: a) the yield sign, an inverted triangle; and the stop sign, a hexagon. To detect the position of the sign in the image, we must know the two properties i.e., color and shape. The applications and the difficulty of road sign detection make road sign detection an interesting problem. In terms of applications, road sign detection is quite important for the road sign recognition problem, since it is the most important step for a road sign recognition system. So far, the researchers have mainly focused on the road sign recognition problem, in which the task of finding road sign in an arbitrary background is usually avoided by either manual segmentation of the input image, or by capturing faces against a known uniform background. In the last decade, road sign detection has attracted great attention, as road sign recognition system requires automatic road sign detection as a first step, especially for images with cluttered background. Road sign detection also has potential applications in human computer interface and surveillance systems. Road sign detection is difficult due to three main reasons. First, there is a large component of non-rigidity and textural differences among road sign. Second, road sign detection is also made difficult because of additional features, such as dust, which can either be present or totally absent from a road sign. All these additional features increase the variability of the road sign patterns that a road sign detection system should handle. Third, the presence of unpredictable imaging conditions in an unconstrained environment increases the difficulty of the task. A change in light source distribution can cause a significant change in the appearance of the road sign image. All these things should be taken into consideration when designing a road sign detection system. (Md. Safaet Hossain, June 2015)

II. BACKGROUND

The Road Sign Recognition (RSR) is a field of applied computer vision research concerned with the automatic detection and classification of traffic signs in traffic scene images acquired from a moving car. The result of the RSR research effort can be used as a support system for the driver. When the neighborhood environment is understood, computer support can assist the driver in advanced collision prediction and avoidance. Driving is a task based almost entirely on visual information processing. The road signs and traffic signals define a visual language interpreted by drivers. Road signs carry many information necessary for successful driving - they describe the current traffic situation, define right-of-way, prohibit or permit certain directions, warn about risky factors etc. Road signs also help drivers with navigation. Two basic applications of RSR are under consideration in the research community - driver's aid (DSS) and automated surveillance of road traffic devices. It is desirable to design smart car control systems in such a way which will allow the evolution of fully autonomous vehicles in the future. The RSR system is also being considered as a valuable complement of the GPS-based navigation system. The dynamical environmental map may be enriched by road sign types and positions (acquired by RSR) and so will increase the precision in the vehicle positioning. Problems concerning traffic mobility, safety and energy consumption have become more serious in most developed countries. The endeavors to solve these problems have triggered the interest toward new fields of research and applications such as automatic vehicle driving, in which new techniques are investigated for the entire or partial automation of driving tasks. A recently defined comprehensive and integrated system approach referred to as intelligent transportation system (ITS), links the vehicle, the infrastructure, and the driver to make it possible to achieve more mobile and safer traffic conditions by using state of the art electronic communication and computer-controlled technology. Over time the ITS research community expects that intelligent vehicles will advance in three primary ways: in the capabilities of in-vehicle systems, in the sophistication of the driver-vehicle interface and in the ability of vehicles to communicate with each other and a smart infrastructure. An example of this can be found in the research by DaimlerChrysler. A vehicle detects ice on the road and notifies this to a radio transmitting station which broadcasts this information to all other vehicles approaches this area. The vehicle can also transmit the information to vehicles moving behind and approaching this area. The first car may have to break, and it will warn approaching vehicles of this intention so that it is not hit from behind.

Smart vehicles will be able to give route directions, sense objects, warn drivers of impending collisions (with obstacles, other cars and even pedestrians), automatically signal for help in emergencies, keep drivers alert, and may ultimately be able to take over driving. ITS technologies may provide vehicles with different types and levels of "intelligence" to complement the driver. Information systems expand the driver's knowledge of routes and locations. Warning systems, such as collision avoidance technologies, enhance the driver's ability to sense the surrounding environment, help the driver in sorting and understand all the information passed to him via road signs and other types of road markings. In the last two decades government institutions have activated initial explorative phases by means of various projects world-wide, involving many research units who worked co-operatively, producing several prototypes and solutions, based on rather different approaches. In Europe, the PROMOTHEUS project (Program for a European Traffic with highest Efficiently and Unprecedented Safety) started this exploration stage in 1986. The project involved more than 13 vehicle manufacturers and several research units from governments and universities of 19 European countries. Within this framework several different ITS approaches were conceived, implemented and demonstrated. In the United States a great deal of initiatives was launched to address the mobility problem involving universities, research centers and automobile companies. After this pilot phase, in 1995 the US government established the National Automated Highway System Consortium (NAHSC) and launched the Intelligent Vehicle Initiative (IVI) right after in 1997. In Japan, where mobility problem is even more intense and evident some vehicle prototypes were also developed within the framework of different projects. Similarly, to the US case, in 1996 the Advanced Cruise-Assist Highway System Research Association (AHSRA) was established amongst many automobile industries and research centers which developed different approaches to the problem of automatic vehicle guidance. The ITS is now entering its second phase characterized by a maturity in approaches and by new technological possibilities which allow the development of the first experimental products. Several prototypes of intelligent vehicles have been designed, implemented and tested on the road. The design of these prototypes has been preceded by the analysis of solutions deriving from similar and close fields of research and has produced a great flourishing of new ideas, innovative approaches and novel ad hoc solutions. Robotics, artificial intelligence, computer science, computer architectures, telecommunications, control and automation and signal processing are just some of the principal research areas from which the main ideas and solutions were first derived. Initially underlying technological devises, such as head up displays, infrared cameras, radar and sonar derived from expensive military applications but, thanks to the increased interest in these applications and to the progress in industrial production, today’s technology offers sensors, processing systems and output devices at very competitive prices. To test a wide spectrum of different approaches these automatic vehicles prototypes are equipped with many different sensor and computing engines (Johansson, 2015)

III. STRUCTURE OF THE ALGORITHM

Step 1: Detect Candidate Text Regions Using MSER

The MSER feature detector works well for finding text regions. It works well for text because the consistent color and high contrast of text leads to stable intensity profiles.

Use the |detectMSERFeatures| function to find all the regions within the image and plot these results. Notice that there are many non-text regions detected alongside the text. (Chen, 2011)

Step 2: Remove Non-Text Regions Based on Basic Geometric Properties

Although the MSER algorithm picks out most of the text, it also detects many other stable regions in the image that are not text. You can use a rule-based approach to remove non-text regions. For example, geometric properties of text can be used to filter out non-text regions using simple thresholds. Alternatively, you can use a machine learning approach to train a text vs. non-text classifier. Typically, a combination of the two approaches produces better results. (Neumann, 2012) This example uses a simple rule-based approach to filter non-text regions based on geometric properties.

There are several geometric properties that are good for discriminating between text and non-text regions (Gonzalez, 2012) (Li, 2012), including:

* Aspect ratio
* Eccentricity
* Euler number
* Extent
* Solidity

Use |regionprops| to measure a few of these properties and then remove regions based on their property values.

Step 3: Remove Non-Text Regions Based on Stroke Width Variation

Another common metric used to discriminate between text and non-text is stroke width. _Stroke width_ is a measure of the width of the curves and lines that make up a character. Text regions tend to have little stroke width variation, whereas non-text regions tend to have larger variations.

To help understand how the stroke width can be used to remove non-text regions, estimate the stroke width of one of the detected MSER regions. You can do this by using a distance transform and binary thinning operation. (Li, 2012)

Step 4: Merge Text Regions for Final Detection Result

At this point, all the detection results are composed of individual text characters. To use these results for recognition tasks, such as OCR, the individual text characters must be merged into words or text lines. This enables recognition of the actual words in an image, which carry more meaningful information than just the individual characters. For example, recognizing the string 'EXIT' vs. the set of individual characters {'X','E','T','I'}, where the meaning of the word is lost without the correct ordering.

One approach for merging individual text regions into words or text lines is to first find neighboring text regions and then form a bounding box around these regions. To find neighboring regions, expand the bounding boxes computed earlier with |regionprops|. This makes the bounding boxes of neighboring text regions overlap such that text regions that are part of the same word or text line form a chain of overlapping bounding boxes.

Now, the overlapping bounding boxes can be merged together to form a single bounding box around individual words or text lines. To do this, compute the overlap ratio between all bounding box pairs. This quantifies the distance between all pairs of text regions so that it is possible to find groups of neighboring text regions by looking for non-zero overlap ratios. Once the pair-wise overlap ratios are computed, use a |graph| to find all the text regions "connected" by a non-zero overlap ratio.

Use the |bboxOverlapRatio| function to compute the pair-wise overlap ratios for all the expanded bounding boxes, then use |graph| to find all the connected regions.

Step 5: Recognize Detected Text Using OCR

After detecting the text regions, use the |ocr| function to recognize the text within each bounding box. Note that without first finding the text regions, the output of the |ocr| function would be considerably noisier.

IV. Results

Below is the first set of results,

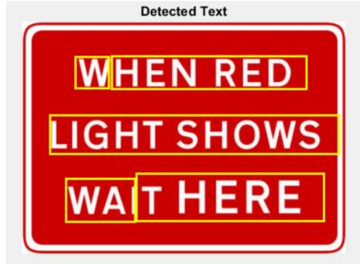
As we can see in the following detected text. All the area of the road sign is detected and all the data on the road sign is read. In the correct order as well.


```
ans =

    '

     SPECIAL PLATE
     REQUIRED
     UNAUTHORIZED
     VEHICLES
     MAY BE TOWED
     AT OWNERS
     EXPENSE
```

Moving on to the next set of image.


```
ans =

    'HEN RED

     WAIT HERE

     UGHTSHOWSI

     '
```

In these images we can see that there are some mistakes as to how the images are read. There are almost words being formed there. So we go ahead and try to tweak the parameters in the code further in order to obtain more desirable results.




```
ans =

    'HEN RED

     \T HERE

     LIGHT SHOWS I

     '
```

In this next set of same images we tried to alter a few parameters in order to read the full text from the road sign. But it just made things worse. We could read LIGHT correctly than before but still it doesn't make sense. We lost the words WAIT and WHEN.


```
ans =

    'SCHOOL||

     WHEN
     FLASHING

     SPEED

     LI

     '
```

In this above image we try to run the default code without any changes to the code. We were able to read the words that are close to each other and could make a sense of the road sign. But without any modification to the cde it is not possible to read all the road signs effectively.

V. Conclusions

The code is still to be tweaked with to get it best working with all types of images and backgrounds so that there is no error while reading road signs and mistaking one road sign for another or confusing the driver with wrong info displayed on his/her dashboard that will lead to an accident in the worst-case condition. We could see that the first image was read without a mistake because the code was worked on keeping that image as a reference. But as we go ahead with the testing of the code to other images it becomes evident that we also need to consider all the conditions of the backgrounds, Size of the road sign, Surrounding colors. Many things will matter to reading the road signs correctly. As we move to the second image we can see that the code recognizes the white edge of the image to be 'I' which is not the case if we can see it correctly. Therefore, the code is not conditioned to be work perfectly for all the images.

Once we try to tweak the code for better reading for the second image, we can see minor improvements in the results. The code could differentiate between 'LIGHT' as spelling from 'UGHT' from the first try. But still we end up with 'HEN' for 'WHEN' and 'T' read instead of 'WAIT'.

Then we move onto the third image where we try to capture the speed limit of the specific section of the road. Since the image was too big when compared to the other two images, the code did good to recognize some of the words on that road sign, but we ended up getting wrong results finally. After 'SCHOOL' there are two '| |' which are the edges of the road sign. It was able to read when flashing correctly in the order as well. But as the font size increases it becomes difficult for the code to understand the text written or to stitch it together. We can 'SPEED' being read but miss the 'LIMIT' which is read as 'LI'.

Concluding, the code can perform well on set of images that was used to build the code. But in the process of building the code using only one image has led to less robustness of the system and now it can only read the contents of the single image correctly.

VI. References

Carwow. (2018). *Glossary*. Retrieved from Carwow: https://www.carwow.co.uk/guides/glossary/how-does-road-sign-recognition-work

Chen, H. (2011). Robust Text Detection in Natural Images with Edge-Enhanced Maximally Stable Extremal Regions. *Image Processing (ICIP), 2011 18th IEEE International Conference on. IEEE*.

Gonzalez, A. (2012). Text location in complex images. *Pattern Recognition (ICPR), 2012 21st International Conference on. IEEE 2012.*

Johansson, B. (2015). *Road Sign Recognition from a Moving Vehicle.* Retrieved from semanticscholar.org: https://pdfs.semanticscholar.org/4bb8/56d5bc9a3153290f2049adfef2d46cdb5e1e.pdf

Karthiga.PL, S. R. (2016). TRAFFIC-SIGN RECOGNITION FOR AN. *CSEIJ*, 9.

Li, Y. a. (2012). Scene text detection via stroke width. *Pattern Recognition (ICPR), 2012 21st International Conference on IEEE*.

Md. Safaet Hossain, Z. H. (June 2015). Traffic Road Sign Detection and Recognition for Automotive Vehicles. *International Journal of Computer Applications(0975-8887)*, 1.

Neumann, L. a. (2012). Real-time scene text localization and recognition. *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.*