VISION BASED DETECTION OF ROAD SIGNS IN VIDEOS AND IMAGES

Andrew Morato
Department of Computer Science
George Mason University
amorato@gmu.edu

Ashwin Ravishankar
Department of Computer Science
George Mason University
aravisha@gmu.edu

Suma Dixit
Department of Computer Science
George Mason University
sdixit4@gmu.edu

Janani Pargunan
Department of Computer Science
George Mason University
jparguna@gmu.edu

May 10, 2019

ABSTRACT

With the advent of autonomous driving coupled with the emerging field of deep learning, there is an explosive research interest in the vision community. In this paper we have implemented and tested four approaches to road sign detection in image datasets and video datasets. The techniques include – *color thresholding and shape approximation*, *template matching*, *hough transform with edge detection*, and *shape detection using edge thresholding*. This work mainly focuses on the image processing and vision applications to detect signs, and not on machine learning techniques.

In the **shape detection using edge thresholding** approach, signs are detected by extracting regions of interest and applying edge detection filters. The **hough transform with edge detection** technique uses hough transform to detect circle and lines using curve fitting algorithm. In **color thresholding and shape approximation**, we explore isolating colors of interest to reduce the number of contours found in an image. Heavily thresholding the contours, approximating their shapes, and removing nonsensical figures yields a set of likely traffic signs. In **template matching** – features of the template image are extracted and stored to a database. With the test image processing module, the feature extraction is done on the test image and matched to the template features to observe if a match exists.

***K*eywords** Road sign detection · Computer vision

1 Introduction and Related Work

Wouldn't it be wonderful to get into a car in California and peacefully sit without going anything till we reach New York City?

Detection and recognition of road sign is am extremely important field of research with the emerging technologies in autonomous driving. Road signs are imperative to road safety as they are used to understand the path being travelled on and serves as warning for drivers and pedestrians. Clear visibility of road signs is crucial for drivers. Many major road accidents are caused due to faults of driver not being aware of localised road safety sign - which may be due to unfavourable visibility and lighting conditions. Also the orientation of the sign could attribute to line of vision. Ideally, the sign should be perpendicular to the direction of the road. Detection of road signs is the phase where we reduce the amount of information to be processed later allowing us to easily locate regions of interest.

We refer to work done in [1] for different possible approaches in vision. Approaches to using *template matching* as a part of the road sign detection pipeline have been experimented in Amal Bouti et al [2], the image is initially processed with color segmentation to weed out the non-red colors in the sensor module and then a matching task is executed to identify the template. Rabia Malik et al [3] restricted their work to signs that have a red boundary with black symbols inside. The template matching technique is used to detect and categorize the symbols enclosed within the red boundary. Among the shape-based detection, edge-based detection plays a vital role since traffic signs have distinctive polygon shapes. According to [4] and [5], traffic signs can be detected by extracting regions of interest and applying edge detection like Laplacian of Gaussian (LoG) or Sobel approximation, which is explored in this project using the modules from OpenCV in Python and later compared with the efficiency of other techniques.

Several research works have implemented Hough Transform for detecting traffic signs. The research works in [6] and [7] have laid the foundation for the Hough Transform implementation in this project and [7] has been extended to detect other shapes as well such as hexagonal, square and rectangular shapes.

In this paper, section 3 describes the methodologies and the experimental setting for all 4 algorithms. Results of the experiments and comparative analysis is done in Section 4. Conclusion of our work is done briefly in Section 5.

1.1 Dataset

- **LISA** – Laboratory for Intelligent and Safe Automobiles released a traffic sign dataset, containing videos and annotated frames of images extracted from the video[8]. This is the image dataset used in the project.
- **Deep Drive** - This is a large scale driving video dataset of over 100K videos with diverse annotations [9]

2 Technical Approach

In our project, we experiment 4 approaches to detect road signs as demonstrated in this section.

2.1 Shape Detection Using Edge Thresholding

The steps involved in this algorithm's image processing pipeline is described below.

1. **Image smoothing**: Image smoothing is essential if we have multiple noise levels, that may make the interesting regions hard to detect. We use Gaussian blurring for this phase.
2. **Color thresholding**: Most of the traffic signs have only a few significant colors associated with them like yellow, red, white etc. So, it makes sense to segment the image based on color thresholding to detect regions of image having more color intensity for those significant colors. This is achieved by color thresholding on a RGB image converted to HSI color space, and checking if the image has pixels with intensity values within the specified range for a particular color.
3. **Contouring**: On finding regions of interest through color thresholding, we aim at finding meaningful shapes in those regions by using contouring which is used to detect shapes like squares, triangles, rhombus and other polygons which usually relate with traffic sign shapes.
4. **Edge thresholding**: Contouring is heavily dependent on the threshold value and arc length specified, which may sometimes result in detecting shapes even in regions that don't have any resulting in false positives. For ex: if we have an image of a car is moving on a road beside a yellow colored footpath and a yellow-colored traffic sign ahead, then after color thresholding and contouring both the yellow colored regions may be detected as circles/rectangles. That leaves us to classify which of the regions is actually a traffic sign for which we apply edge thresholding. Both the contour region values are stored in a temporary image array and edge detection applied. Region with traffic sign has higher number of non-zero edges since there are text/words/lines present which cannot be found in non-traffic sign contours. After this technique, we draw a bounding box around the region which satisfies the edge threshold. We have majorly used Canny edge detector from OpenCV, but using Sobel / Laplacian also should work fine.

2.2 Template Matching using Feature Descriptors

Features are that componenet of an image which gives us a better unserstanding of the contents of the image. OpenCV has several feature extraction techniques incuding SIFT, ORB, KAZE, SURF etc. We used the LISA image dataset to run this experiment.

As opposed to image pre-processing to obtain only the region of interest and then run a detection algorithm, in this project, we look at effects and effectiveness of running a feature template matching algorithms on raw images to detect signs.

The main challenge is the occlusion in the image caused in the pre-processed image dataset. The image datasets for road sign detection are originally videos from where up to 30 frames are extracted per track and annotated with the correct label. In this paper we have experimented with SIFT (Scale Invariant Feature Transformation) [10] features for matching. The SIFT features for the template image are pre-computed and stored to the database. For a series of test images, the SIFT descriptor vectors are computed and matched against the template feature descriptors to get the match result. The templates used are – red stop sign, yellow warning signs and regulatory speed limit sign. Randomly sampled 50 images per template class are are used to check for efficiency. The effectiveness of this approach is computed with metrics of accuracy.

A threshold of minimum 3 match pairs is set for an image to be classified correctly.

2.3 Color Thresholding and Countour / Shape Appriximation

This approach focuses heavily on noise removal and thresholding before applying edge detection and shape approximation. After employing several basic preprocessing strategies such as a gaussian blur and histogram equalization, we convert the image to Hue-Saturation-Value (HSV) color space before color thresholding is enforced. HSV color space is naturally favorable to color thresholding in noisy images since it allows for setting a hue and a range of saturation/brightness values to account for differences in light intensity, as opposed to BGR color space which is notoriously vulnerable to shadow or brightness. The high-level algorithm is as follows[1]:

1. Color Threshold in HSV color space for common sign colors (red, yellow, green, white).
2. Produce an edge map of the thresholded image using the Canny edge detector or Sobel masks.
3. Find external contours in the edge map (i.e. ignore interior edges).
4. Approximate the shapes of the contours to basic shapes.
   - (a) Use the Douglas-Peucker algorithm to approximate a set of vertices from a contour.
   - (b) Count the vertices to classify a basic shape.
   - (c) Refine the shape by considering figure orientation and relative vertex displacement.
5. Filter the contours to remove non-standard shapes.

2.4 Hough Transform with Edge Detection

The Hough Transform is a common technique for extraction of lines in image processing. In this paper, it has been extended to identify traffic signs of triangular, hexagonal and rectangular shapes. The first pre-processing step involves image segmentation which is performed according to color features. The RGB system is more sensitive to illumination changes than the HSI system, so the HSI color space is used for image segmentation. From the color segmented image, the edges are identified using Canny edge detection. The classical Hough algorithm can be easily extended to find any curves in an image that can be expressed analytically in the form f(x, p) = 0. The aim is to detect the straight lines intersecting with each other. The strategy is used applies Hough transform on the edge detected image and finds the intersection points of all the Hough lines detected. In order to handle cases where the detected intersection points are more than the number of vertices of a traffic sign's shape, a monotone chain Convex Hull algorithm is used for detecting the boundary points around the traffic signs. In this way, only those points that are part of the existing traffic signs in the image are detected. With one of these reference points, the bounding rectangle around the traffic sign is drawn around the traffic sign.

3 Results and Discussion

Results of the individual experiments are discussed here and a comparative study of image dataset based approaches - Shape Detection Using Edge Thresholding, Template matching and Hough transform with edge detection, is made.

3.1 Shape Detection Using Edge Thresholding

We have looked up various research that have dealt with this approach like [1], [4] and [5] and devised a simple approach for detection in images and especially for the ones taken in real time. We think the better part of this experiment was using noisy and challenging images from LISA and achieving a relatively better accuracy rate. We worked with two test

[1]OpenCV offers implementations for most algorithms used.

Figure 1: Edge Thresholding Experiment Result

Table 1: Tabular result of edge thresholding method

Dataset (#Images)	True Positive	False Positive	False Negative	True Negative
#1 - 102	95	2	3	0
#2 - 101	81	14	5	0

set - 1 set with poor lighted scene and shadows and blurred, while other had relatively brighter images with clear color contrast testset 1.

True negative is 0 for both cases since all the images had at least 1 traffic sign. False positive is a case where non-traffic signs are detected, and bounding box drawn. False negative is a case where algorithm fails to detect the sign even when there is one. As we can see, failure percentage for case 2 is higher (18%) than for case 1(4.9%).. Sample test results are plotted in Figure. 1 and tabulated in Table 1

3.2 Template Matching

A total of 150 randomly sampled images are used for testing - equally balanced over the 3 template classes: stop sign, warning sign and speed limit sign. The major challenge faced in the template matching approach with this dataset was the lack of clarity in the test image. In some images, as human experts, we were unable to identify the sign ourselves as the orientation of the sign within the image was extremely distorted. Also, there were test images that partially cut off the region of interest. Also, since the dataset was curated from videos – resulting in sequential motion of the camera, test signs at a far-off distance were not matched with the algorithm, but an image with great clarity of region of interest was matched with a great confidence score. We also came across images that had an opaque lamppost, moving car, and other objects distorting the region of interest.

A sample of correctly classified test image is shown in Figure 2.

Figure 2: Template Matching Algorithm Result

Table 2: Test result for template matching algorithm

Template Image	# Test Images	# Correctly Detected	Accuracy	Threshold
Stop Sign - Red	50	36	72%	3
Warning Sign - Yellow	40	32	64%	3
Regulatory Sign - White	40	39	**78%**	3

3.3 Color Thresholding and Shape Approximation

This approach has some difficulty detecting signs when color thresholding fails. It struggles particularly at night and when there is not much contrast between the background and the object. For example, a green sign with a tree in the background would likely not be detected. A fair amount of false positives were detected as well, especially license plates and posters by the road that are not actually traffic signs. A sample video of the algorithm tested on a subset of Berkeley's DeepDrive dataset is included in the presentation (pptx file).

3.4 Hough Transform with Edge Detection

This algorithm was tested on a dataset containing 100 images from LISA, out of which 61 images where correctly detected with traffic signs resulting in a success rate of 61%. The Hough Transform technique is computationally expensive, with run-time complexity of $O(n^2)$ and can be used for detecting traffic signs on image datasets than video datasets.

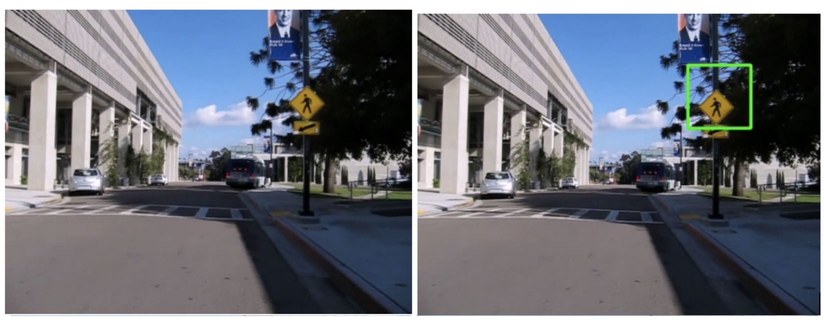

Figure 3: Visualization of Hough Transform With Edge Detection

4 Conclusion

From the results obtained from our experiments on image datasets - *shape detection using edge thresholding* works best followed by raw *template matching*. The worst performing algorithm is *hough transform with edge detection* in our test environment. Every algorithm explored in this project has its own advantages and disadvantages. With shape/edge detection, the pros are that they work better and more accurate on images takes in daytime and have more clarity. Also, it is possible to process more images/frames in small time, making it time efficient. While the drawback, can be that the case of mis-classification may be higher with poor images and is heavily dependent on threshold parameters.

5 Project Contribution

Individual contributions to the project:

- **Janani Pargunan** - Developed python module on *Hough transform with edge detection*, contributed module related content to report and presentation slides.
- **Suma Dixit** - Developed python module on *shape detection using edge thresholding*, contributed module related content to report and presentation slides.
- **Andrew Morato** - Developed python module on *color thresholding*, contributed module related content to report and presentation slides.

- **Ashwin Ravishankar** - Developed python module on *template matching*, contributed module related content to report and presentation slides.

References

[1] Neeraj Shukla B. K. Das Prachi Dewan, Rekha Vig. An overview of traffic signs recognition methods. In *International Journal of Computer Applications (0975 – 8887)*, 2017.

[2] A. Bouti, M. A. Mahraz, J. Riffi, and H. Tairi. Robust system for road sign detection and recognition using template matching. In *2017 Intelligent Systems and Computer Vision (ISCV)*, 2017.

[3] R. Malik, J. Khurshid, and S. N. Ahmad. Road sign detection and recognition using colour segmentation, shape analysis and template matching. In *2007 International Conference on Machine Learning and Cybernetics*, 2007.

[4] H. Fitriyah, E. R. Widasari, and G. E. Setyawan. An overview of traffic signs recognition methods. In *Traffic sign recognition using edge detection and eigen-face*, 2017.

[5] L. Felipe Blázquez Luis J. Miguel Rubén Laguna, Rubén Barrientos. Traffic sign recognition application based on image processing techniques. 2017.

[6] Miguel Garrido, Miguel-Angel Sotelo, and Ernesto Martín-Gorostiza. Fast road sign detection using hough transform for assisted driving of road vehicles. volume 3643, pages 543–548, 01 1970.

[7] Pilar García-Díaz, S Maldonado-Bascón, S Lafuente-Arroyo, and Rocío Montero. Triangular traffic sign detection through hough transform. pages 563–567, 01 2005.

[8] Laboratory for Intelligent and Safe Automobiles. Image dataset. `http://cvrr.ucsd.edu/LISA/lisa-traffic-sign-dataset.html`. [Online; accessed 09-May-2019].

[9] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling, 2018.

[10] Wikipedia. Scale invariant feature transform. `https://en.wikipedia.org/wiki/Scale-invariant_feature_transform`. [Online; accessed 09-May-2019].