Detection and recognition of text-based traffic signs with convolutional neural network

Changhoon Kim, Kaihua Zhang, Yaoyang Lin, Yuanhang Luo

***Abstract*—A system of detecting and recognizing text in traffic signs has been developed in the project. Maximally Stable Extremal Regions (MSERs) operator is used to detect characters. Characters are cropped from traffic sign images and since it tends to cut character as well as traffic icons from traffic sign images, eigen space method is employed to filter out traffic icon images. For the recognition, a convolutional neuron network (CNN) is trained with Chars74K dataset to perform character recognition. The methods are evaluated and achieve an accuracy of 81.66%.**

I. Introduction

TRAFFIC signs enforce the road regulation to reduce accidents by giving the information on the speed, road condition, or sharp turns, etc. Recognizing and understanding the meaning of traffic signs is a challenging problem. Most of the current researches exist on detecting and recognizing symbol-based traffic signs[1]-[3], and a few researches are conducted on recognizing text on traffic information signs which may because of the difficulty of this task. However, understanding traffic sign plays an important role in driver assistant system and autonomous vehicle by delivering essential traffic information.

The project focuses on using computer vision techniques and deep learning methods to detect and recognize the text in images of traffic signs. The proposed system comprises two main parts: text detection and text recognition. Text detection part exploits MSERs method to find and crop characters from images of traffic signs.Then the cropped images are classified as character image and traffic icon images with Eigen space method. Once the character images are cropped and filtered, a pre-trained convolutional neuron network (CNN) attempts to recognize the text as letters and numbers.

In section two, related work are demonstrated. Section three describes the detailed methodology for detection and recognition, while the results are demonstrated to illustrate the performance of the system in section four. Discussion and conclusion is drawn in part five analyzing the results from section four and concluded our work on the project.

II. Related Work

A. Text Detection Methods

Text detection aims at locating interesting area and cropping it as input of CNN. Greenhalgh[4] used Maximally Stable Extremal Regions (MSERs) to detect candidate regions which offers robustness to variations in lighting conditions. Huang[5] combined MSERs with CNN to distinguish low-level pixel components. In our project, MSERs is used as detection method too since text on traffic signs has constant color, and a relatively clear edge which make MSERs a feasible way.

Eigen space is a another classic way of detection. L. Sirovich[10] introduced eigen face for recognition and M.Turk et al.[11] used the concept for face classification. In these papers, the space can be used for detector of non-face. T. Amano et al[12] utilize this approach to pose detection. H. Foroughi et al[13] showed eigen space approach for detecting humans falling. They use Integrated Time Motion Image (ITMI) for extracting eigen motions. These motions are used to feeding neural networks for precise motion detecting.

B. Text Recognition Methods

Text recognition takes a cropped image and recognizes the words depicted. Some researches focus on word-based recognition which extract features from the entire word area, e.g.[6][7]. While other researches explore characters based recognition relying on individual character classifier. Bissacco[9] uses PhotoOCR system to recognize characters. Goodfellow[8] has successfully employed CNN with multiple position-sensitive character classifier to recognize street number. Our method also follows characters based approach and takes the character image as input to a pre-trained CNN to recognize the characters.

III. Methodology

The system has been divided into two parts: text detection and text recognition. The flow chart in Fig.1 shows the pipelines of detecting and recognizing text-based traffic signs.

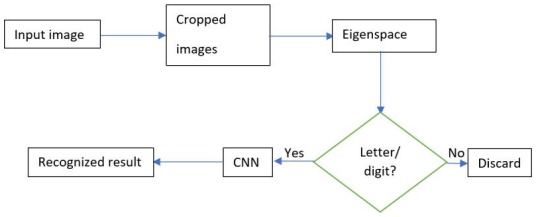

Fig. 1: Flowchart of detecting and recognizing text-based traffic signs

A. Detection

1) Maximally Stable Extremal Regions (MSERs): Maximally Stable Extremal Regions (MSERs) detection is utilized

Advisor: Professor Jason Corso, GSIs: Siyuan Chen and Byungsu Kyle Min

in this project to detect the possible candidates for classification. MSERs operator is a method for blob detection which applies different thresholds on a gray scale image and finds the connected component throughout the sequence of threshold images. Texts on road signs have constant color, and a relatively clear edge which make MSERs a feasible way to detect possible candidates, including letters, digits, and some icons that are constant in color. MSERs operator is also robust to lightning condition changes and viewpoint changes, which are very common in real life road signs.

Before applying MSERs operator on an RGB image of a road sign, some preprocessing steps are needed. To remove some noises, dilation and erosion are applied on the input image. Padding and resizing ensure the image to be square of desired size. Turning the color image into gray scale prepares for MSERs detection process.

MSERs operator tends to return many bounding boxes for a single candidate. Removing the redundant bounding boxes is necessary for cropping the image accurately. From experiments, if the bounding boxes are near to each other within an error of 10% of the width or height of the bounding box, then they should be merged. To filter out some non-text regions, simple geometric criterion is applied, i.e., assume each letter or digit has a width to height ratio between 0.1 and 1.3, where 0.1 takes care of the situation for elongated letters like "I", and 1.3 for wide letters like "W". Moreover, they are assumed to be at least 5 pixels wide and 10 pixels high, otherwise it is too vague to classify.

Another thing needs to be considered is predicting lines from detected MSERs. In our work, it is assumed that the input images are from normal driving situations, therefore the texts should be roughly horizontally arranged. To predict lines, each bounding box is stretched horizontally and squeezed vertically by a small factor. The overlapping bounding boxes merge together and create a line. This method is better than line fitting in the sense that it can detect spaces and thus separate words. The result of grouping outcome is saved to dict.txt, which contains a python dictionary with each key as a word, and its corresponding value is an ordered list of index of letters of this word.

2) Eigen Space Method: The MSERs method passes the cropped images of characters and traffic icons. Since our CNN is trained for classifying the characters, the icons should be filtered out before feeding to CNN. Eigen face detection approach[11] is employed for this purpose. The eigen face method constructs a eigenvector space from training images and compares the projections between character images and traffic icons images. 6200 images (62 classes (a-z, A-Z, 0-9), 100 from each classes) are investigated to build the eigenvector space. The original image size is 128×128 which is resized to 32×32 gray scale image for computational efficiency. Then, the mean image is calculated by averaging on vectorized images. Using mean image, the concatenated matrix is normalized and eigenvectors are calculated. However, most eigenvalues are about to zero which means most of them are noise eigenvalues. Thus, we plotted eigenvalues and chose 48 eigenvectors to make orthogonal projection matrix. 50 images from each classes are projected onto eigen space for calculating the threshold of character images. Thus, if the euclidean distance of the cropped image and its projected image were larger than threshold, it is treated as an icon and will not be passed to CNN. See Fig.2 for the denotation of projection and distances.

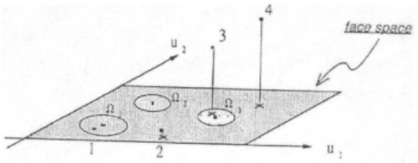

Fig. 2: Eigen space method with prejection and euclidean distance

B. Recognition

Recognition part takes the characters/numbers bounding boxes generated by recognition section as input and recognizes the text inside the bounding boxes. A multi-class classification problem has been formulated for character recognition, i.e. one class for per character. where characters c are selected in a pre-defined dictionary C according to the probability. To this end a deep CNN classfier is employed to perform classification where each character corresponds to an output neuron. The CNN contains four convolutional layers and two fully connected layers, the last fully connected layers performs classification according to the pre-defined dictionary, thus has the same number of neurons as the size of dictionary. In other words, the CNN is feed by cropped image and produces a probability of distribution over all the characters and numbers in the dictionary. The key of dictionary with maximum probability is taken as the recognition result, which can be described in the following equation,

$$c^* = argmax P(c|x)$$

in which x is the input character image.

The CNN model contains six weighty layers built with Pytorch package - four convolutional layers and two fully-connected layers. The size convolutional layers are {*filter size, filter numbers, stride, padding*}: {*5,6,1,2*}, {*5,16,1,2*}, {*5,32,1,1*}, and {*5,32,1,1*}. The inputs to convolutional layers are zero-padded to preserve dimensionality. The stride is set as 1 to capture the details of input. The fully connected layers have 120 units and 62 units corresponding to 10 numbers and 52 upper and lower letters. The capital letters and lower letters would be treated as the same letter since it has no influence on understanding the meaning of traffic signs. Each convolutional layers is followed by a 2×2 max pooling layer and every hidden layer is activated by Rectified Linear Unit (ReLU). The illustration of CNN model is in fig.3. See Table 1 for the details of CNN model.

IV. Results and Discussion

A. Detection

1) MSERs: The accuracy in this step consists of two parts: precision rate =77.35%, and recall rate=76.28%. This

TABLE 1: Details of Convolutional Neuron Network

Layer	Conv1	Conv2	Conv3	Conv4	Full1	Full2
Kernel Size, Stride, Pad	5, 1, 2	5, 1, 2	5, 1, 2	5, 1, 2		
Input Size (chan,h,w)	1, 128,128	6, 64,64	16, 32, 32	32, 16, 16	32, 8, 8	1, 120
Pool Size (h,w)	2, 2	2, 2	2, 2	2, 2		
Output Size (chan,h,w)	6, 64,64	16, 32, 32	32, 16, 16	32, 8, 8	1, 120	1, 62
Addition	ReLU	ReLU	ReLU	ReLU	ReLU	Dictionary

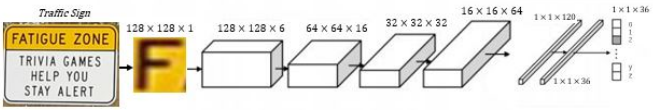

Fig. 3: Illustration of CNN model for character classification

is because some undesired regions are still detected due to its similarity to icons and texts, and some single letter is detected as two separate parts. On the other hand, some text regions are ignored due to the low contrast between background and the text itself. Moreover, due to blurring and ambiguity in edges in some images, multiple letters are sometimes detected together as one integrated part. Fig.4 shows the output with respect to steps of MSERs. See Fig.5 for some tough examples of detected characters on traffic sign images.

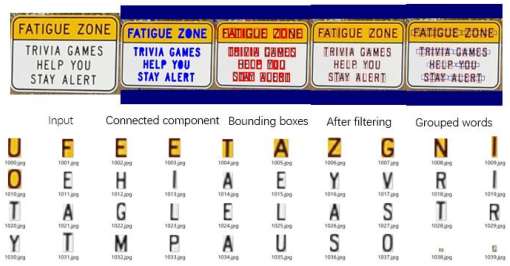

Fig. 4: Illustration of pipeline of detecting characters with MSERs (Grouped: 0: [1, 5, 4, 9, 7, 0, 3], 1: [6, 10, 8, 2], 2: [22, 34, 32, 11, 36], 3: [12, 15, 23, 33], 4: [28, 18, 13, 17, 19, 14], 5: [27, 20, 26, 16], 6: [21, 25, 24, 29, 31], 7: [30, 37, 35])

Fig. 5: Detection results of MSERs with tough examples

2) Eigen Space: To evaluate the accuracy of eigen space method, 410 cropped images from traffic signs are manually labeled as character images and traffic symbol images, which contains 388 character images and 22 traffic symbol images. The accuracy of detecting it as a character images is 95%, the details are shown in Table 2.

TABLE 2: Accuracy of Eigen Space Method

Characters Images	388
Icon Images	22
Total Images	410
Accuracy	95%

Eigen space has limited accuracy. Several reasons may count for this. First of all, for computation efficiency we decrease the size of the training images 128×128 to 32×32. However, eigen space is vulnerable to scale. Thus, it would greatly impact to the accuracy. Also, the orientation of cropped image is different with training data. Some of them are tilted because of position of camera. Furthermore, lightning condition is also different. Some of test set are taken at afternoon and the others are taken at sunset. Therefore, eigen space has some limitations for explaining training data.

B. Recognition

1) Dataset: The CNN model described in Methodology Section is trained by the Chars74K dataset which contains 62 classes (0-9, A-Z, a-z) in hand writing characters, natural image characters and computer fonts. Considering the characters on traffic signs are similar to computer fonts, a subset of computer fonts with 4 variations (combinations of italic, bold and normal) is used for training CNN model. This database consists of a training set of 56792 128×128-pixel labeled grayscale images of synthesized characters. The example training images are shown in Fig.6.

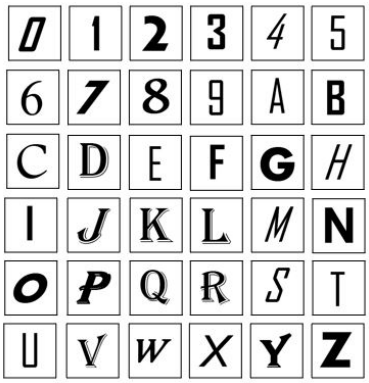

Fig. 6: Examples of training dataset

The test set cropped by detector is been preprocessed to get the required size and color for CNN model. These data are annotated by hand.

2) Preprocessing: All the cropped images from the detector are of different shapes and sizes so that preprocessing is required for testing. White boarder is added to the input image to make it square like without distortion. Then it is resized to 128×128- pixel (same as training image). Moreover, because all training images are binary, all the input image is being

thresholding to get the best result. Fig.7 shows the image before and after possessing.

Fig. 7: Examples of prepossessing

3) Data Augmentation: Some real-world traffic signs are viewed or photoed from different angle or rotated. In order to reduce overfitting on models and make the model more robust to these transformations. The training data was created by applying both projection and rotation transformations to the original training image. This stage will give the model ability to detect the tilted and distorted image from the detector. From the result of experiment, the data augmentation can improve the accuracy by 5%.The example of generated affine images is demonstrated in Fig.8.

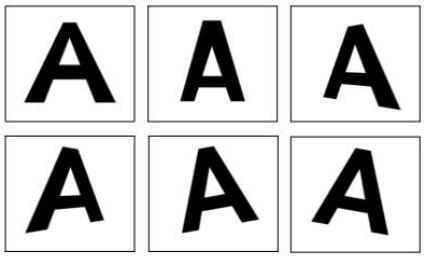

Fig. 8: Examples of generated distorted images

4) Training and Accuracy: The learning rate is set as 0.001 and batch size is 4. Loss is evaluated by cross-entropy error while the gradient degradation is fulfilled by Stochastic Gradient Descent (SGD). The training has 20 epochs and the loss degradation is shown in fig.9. The trained model is saved and used for classify characters on test images. The test images are from real-world traffic sign photos, some of them are cropped from dataset named "Real Time Detection and Recognition of Road Traffic Signs", the others are from google pictures. There are 62 traffic sign images in total and 1178 characters images are cropped from them.The CNN model recognized 962 character images correctly from 1178 character images, so the final accuracy is 81.66%. The accuracy can be influenced by the quality of cropped image and the upper and lower characters also would lower the accuracy, i.e. the "z" and "Z" is hard to tell even from human eyes which could influence the recognition accuracy.

Fig. 9: Lost change in 20 epochs of training CNN

V. Conclusion

In this project, the text-based traffic signs are understood by two steps: detection and recognition. Detection part involves MSERs methods to crop characters from traffic sign images with recall rate at 76.28% and precision rate at 77.35% . The cropped images contains noises like traffic symbol images, thus the eigen space method is exploited to filter our traffic symbol images which achieves 95% accuracy. Then the filtered character images are feed into a pre-trianed CNN model. Based on the result from test dataset, the CNN model achieves 81.66% accuracy.

References

[1] J. Greenhalgh and M. Mirmehdi, Traffic sign recognition using MSER and random forests, in *Proc. EUSIPCO*, Aug. 2012, pp. 19351939.

[2] S. Maldonado-Bascn, S. Lafuente-Arroyo, P. Gil-Jimenez, H. Gomez-Moreno, and F. Lopez-Ferreras, Road-sign detection and recognition based on support vector machines, *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 2, pp. 264278, Jun. 2007

[3] J. Greenhalgh and M. Mirmehdi, Real-time detection and recognition of road traffic signs, *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4,pp. 14981506, Dec. 2012.

[4] Greenhalgh, Jack, and Majid Mirmehdi. "Real-time detection and recognition of road traffic signs." *IEEE Transactions on Intelligent Transportation Systems* 13.4 (2012): 1498-1506.

[5] Huang, Weilin, Yu Qiao, and Xiaoou Tang. "Robust scene text detection with convolution neural network induced mser trees." *European Conference on Computer Vision*. Springer, Cham, 2014.

[6] Almazan, J., Gordo, A., Fornes, A., Valveny, E.: Word spotting and recognition with embedded attributes. In *TPAMI (2014)*

[7] Goel, V., Mishra, A., Alahari, K., Jawahar, C.V.: Whole is greater than sum of parts: Recognizing scene text words. In: *ICDAR*, pp. 398402 (2013)

[8] Goodfellow, I.J., Bulatov, Y., Ibarz, J., Arnoud, S.,Shet, V.: Multi-digit number recognition from street view imagery using deep convolutional neural networks.arXiv:1312.6082 (2013)

[9] Bissacco, A., Cummins, M., Netzer, Y., Neven, H.: PhotoOCR:Reading text in uncontrolled conditions. In: Proceedings of the International Conference on Computer Vision (2013)

[10] L. Sirovich; M. Kirby (1987). "Low-dimensional procedure for the characterization of human faces". *Journal of the Optical Society of America* A. 4 (3): 519524. doi:10.1364/JOSAA.4.000519

[11] M. Turk and A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, vol.3, no. 1, pp. 71-86, 1991, hard copy

[12] Toshiyuki AMANO Shinsaku HIURA Akashi YAMAGUTI and Seiji INOKUCHI, Eigen Space Approach for a Pose Detection with Rapge Images, Pattern Recognition, 1996., *Proceedings of the 13th International Conference*, Print ISBN: 0-8186-7282-X, Print ISSN: 1051-4651

[13] Homa Foroughi, Aabed Naseri, Alireza Saberi, An eigenspace-based approach for human fall detection using Integrated Time Motion Image and Neural Network, *Signal Processing*, 2008.ICSP 2008. 9th International Conference