Traffic Sign Detection and Recognition using a CNN Ensemble

Aashrith Vennelakanti, Smriti Shreya, Resmi Rajendran, Debasis Sarkar, Deepak Muddegowda, Phanish Hanagal
Affiliation: Qualcomm India Private Limited, Bangalore
aashu1337vennela@gmail.com, smritishreya7@gmail.com, resmir@qti.qualcomm.com
dsarkar@qti.qualcomm.com, mdeepak@qti.qualcomm.com, phanagal@qti.qualcomm.com

***Abstract*-- In today's world, almost everything we do has been simplified by automated tasks. In an attempt to focus on the road while driving, drivers often miss out on signs on the side of the road, which could be dangerous for them and for the people around them. This problem can be avoided if there was an efficient way to notify the driver without having them to shift their focus. Traffic Sign Detection and Recognition (TSDR) plays an important role here by detecting and recognizing a sign, thus notifying the driver of any upcoming signs. This not only ensures road safety, but also allows the driver to be at little more ease while driving on tricky or new roads. Another commonly faced problem is not being able to understand the meaning of the sign. With the help of this Advanced Driver Assistance Systems (ADAS) application, drivers will no longer face the problem of understanding what the sign says. In this paper, we propose a method for Traffic Sign Detection and Recognition using image processing for the detection of a sign and an ensemble of Convolutional Neural Networks (CNN) for the recognition of the sign. CNNs have a high recognition rate, thus making it desirable to use for implementing various computer vision tasks. TensorFlow is used for the implementation of the CNN. We have achieved higher than 99% recognition accuracies for circular signs on the Belgium and German data sets.**

***Keywords*: Advanced Driver Assistance System, Traffic Sign Recognition, Convolutional Neural Network, Ensemble, TensorFlow, Image Processing**

I. INTRODUCTION

In an attempt to focus on the road while driving, drivers often miss out on signs on the side of the road, which prove to be dangerous for everyone. This problem can be avoided if there was an efficient way to notify the driver without having them to shift their focus. Traffic Sign Detection and Recognition plays an important role here by detecting and recognizing a sign, thus notifying the driver of any upcoming signs.

Many a times, drivers face a lot of challenges on unknown roads and in such scenarios paying attention to both the road and upcoming traffic signs can be an arduous task. TSDR application not only ensures road safety, but also allows the driver to be a little more at ease while driving on tricky roads.

Another commonly faced problem is not being able to understand the meaning of the sign. Even if drivers manage to spot the sign, there is a chance that they do not know what it stands for. With the help of such ADAS application, whatever sign has been detected would be presented to the driver in some form which is convenient, thus making sure that drivers no longer face the problem of understanding what the sign says.

ADAS applications are the future of automobiles and road safety. With so many recent advancements in technology, the automobile industry is enhancing what cars can do. The idea is to make comfort and road safety go hand in hand.

Our paper talks about recognizing a sign on the road in two steps: detection and recognition of the sign. There are many ways to go about detecting a sign like in [1], where the author used a modified Generalized Hough Transform to determine the exact coordinates of the sign, or like in paper [2], where Circular Hough Transform is used to determine the circular prohibitory sign. However, our approach relies on obtaining a Region of Interest (ROI) using the color and shape of traffic signs. The ROI returned is passed on to the next stage of the TSDR i.e., recognition.

In paper [2], a Support Vector Machine (SVM) was used to recognize the sign. Our paper uses a CNN ensemble to recognize the sign. In Paper [3], the authors implemented a CNN ensemble for prediction of gender from face images. The authors in this paper claim to have obtained a higher accuracy of gender prediction by using an ensemble of CNNs. They obtained an accuracy of 97.31% for gender prediction using an ensemble of 3 CNNs and an accuracy of 96.94% for the same using just 1 CNN. There was an improvement of 0.37%.

Similarly, in our paper, by using a CNN ensemble, we were able to obtain a higher recognition accuracy. The Belgium Data Set and the German Traffic Sign Benchmark were used for training and testing the proposed model.

II. PROPOSED SYSTEM

The system we propose in this paper is sectioned into two phases: detection and recognition. The detection phase simply discovers a sign from the environment. When a car is moving at a certain speed, the camera captures the environment, and our algorithm checks to see if a sign is present in that frame or not. Detecting the traffic sign is based on color and shape. In the recognition phase, the proposed algorithm classifies the detected sign. This is achieved with the help of a Convolutional Neural Network ensemble.

A. Detection

In this phase, the image obtained from the camera in the car is preprocessed before the process of detection starts. General preprocessing steps involve converting the obtained RGB image into an HSV image. For detection, the HSV (Hue Saturation Value) color space is preferred over the RGB (Red

Green Blue) color space. HSV is more similar to what the human eye actually sees when compared to an RGB image. An RGB image defines colors in terms of three primary colors, whereas HSV has a greater range of colors. An HSV image is also less susceptible to external light changes. The HSV image is equalized to adjust the contrast in the image by modifying the image's histogram. Once the HSV image is obtained, the next steps would be to detect objects based on their color followed by finding out their shape and validating the object to be a traffic sign.

1) Color Based Detection

The first and most important thing we notice in a sign is the color. Once we see the color red, we realize that the board on the side of the road is actually a traffic sign. This is the same logic we used while proposing our method of detection. From the captured frames, the proposed algorithm knows to check for a sign based on the color red. If a portion of the image falls under the appropriate threshold of red, that part is checked to see if a sign is available. Once the red threshold is verified, the contours of the red part are found.

2) Shape Based Detection

The number of edges is calculated using the contours discovered in the previous step. This is done using the Douglas-Peucker algorithm [4]. In this paper, we talk about 2 shapes: triangle, circle. Once the number of edges is found using the Douglas-Peucker algorithm, the area of the contour is also found. If the number of edges found is 3, and the area satisfies the minimum condition, it is considered to be a sufficiently large triangle. Similarly, if the number of edges found is greater than or equal to 6, and the area of the contour follows the minimum condition, the contour is considered to be a sufficiently large circle.

Once the shapes are found, finding the bounding box is key. The bounding box separates the Region of Interest (ROI) from the rest of the environment. The bounding box usually touches the outer triangle or circle of the detected contour. For a triangular sign, there are two triangles, the outer triangle, which touches the bounding box and the inner triangle which is inside the outer triangle and does not touch the bounding box.

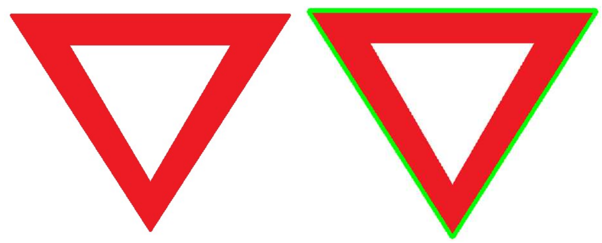
3) Sign Validation

Once the bounding box is found, the sign must be validated. The image on which a threshold filter has been applied is first inverted where the inner triangle which was previously black is now white and the outer triangle which was previously white is now black. If the white triangle touches the boundary, then it is not a sign, and if it does not touch the outer triangle and the boundary box, it is considered to be a traffic sign. This is resized to 48x48 and set to the next phase.

To indicate where the sign is, a green contour is drawn around the detected sign. As seen in Fig. 2(b), the detected sign is highlighted with a green border. OpenCV was used to perform all the above-mentioned preprocessing steps.

(a) (b)

Fig. 1. (a) Image with threshold filter applied and (b) Inversion of image with threshold filter applied



(a) (b)

Fig. 2. (a) Original Image and (b) Sign Detected Image

B. Recognition

Once the sign is detected, the sign must also be classified. With the help of Google's open source machine learning framework, TensorFlow, a convolutional neural network can be implemented.

In our implementation of the recognition phase, the first step was to take the detected sign from the previous phase and perform basic preprocessing on the image. Training and testing of the CNN is the most important part of this phase. We used the German Traffic Sign Benchmark and the Belgium Traffic Signs data set for training and testing.

A convolutional neural network is very similar to the brain in the sense that it also has neurons which in turn have weights and biases. Each neuron receives an input on which it performs some operation and the output is passed as the input to the next neuron. A convolutional neural network can have many layers, the first always being the input layer and the last the output layer. Anything else in between is called a hidden layer.

This paper proposes a feed-forward network with six convolutional layers. This also has fully connected hidden layers with dropout layers in between to prevent overfitting. The model proposed in this paper makes use of the sequential stack provided by Keras, which is an open source high level neural network library that is capable of running on top of TensorFlow. All the layers of the proposed CNN [5] have Rectified Linear Unit (ReLu) activation. ReLu activation is considered to be the most popular activation function for neural networks. The output of the 6^{th} convolutional layer is fed to a fully connected layer, which uses a flatten function to flatten the output at that point. The flattened output is given to the final layer which uses SoftMax activation. A max pooling layer is also added after every two convolutional layers to improve the speed of processing. We do not use just one CNN but an ensemble of CNNs. We have 3 CNNs whose aggregate determines the final result. This provides a much more accurate result than just a single CNN.

The loss, optimizer and metrics of the model must be mentioned. The loss of the model is set to categorical cross entropy which means that the loss of the model is calculated as a value in between 0 and 1 instead of using percentages. The optimizer uses Stochastic Gradient Descent with Nesterov momentum. The metric used is accuracy. To improve the training of the machine, epochs with a backward pass are used. Epochs increase the accuracy of the prediction.

TABLE I
LAYERS OF THE CNN

LAYERS
Convolution (32 filters of size 3x3, stride 1, ReLU)
Convolution (32 filters of size 3x3, stride 1, ReLU)
Max Pooling (2x2, stride 2) + Dropout (0.2)
Convolution (64 filters of size 3x3, stride 1, ReLU)
Convolution (64 filters of size 3x3, stride 1, ReLU)
Max Pooling (2x2, stride 2) + Dropout (0.2)
Convolution (128 filters of size 3x3, stride 1, ReLU)
Convolution (128 filters of size 3x3, stride 1, ReLU)
Max Pooling (2x2, stride 2) + Dropout (0.2)
Fully connected layer (512 nodes, ReLU) + Dropout (0.5)
Fully connected layer (SoftMax)

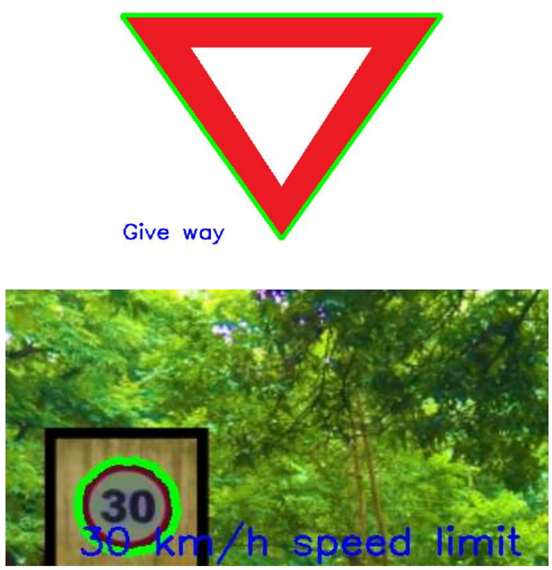


Fig. 3. Examples of a Classified Sign

Fig. 3 depicts how a traffic sign that has been classified looks like. The green border indicates the detection and the blue text below is the interpretation of the sign, thus portraying the classification.

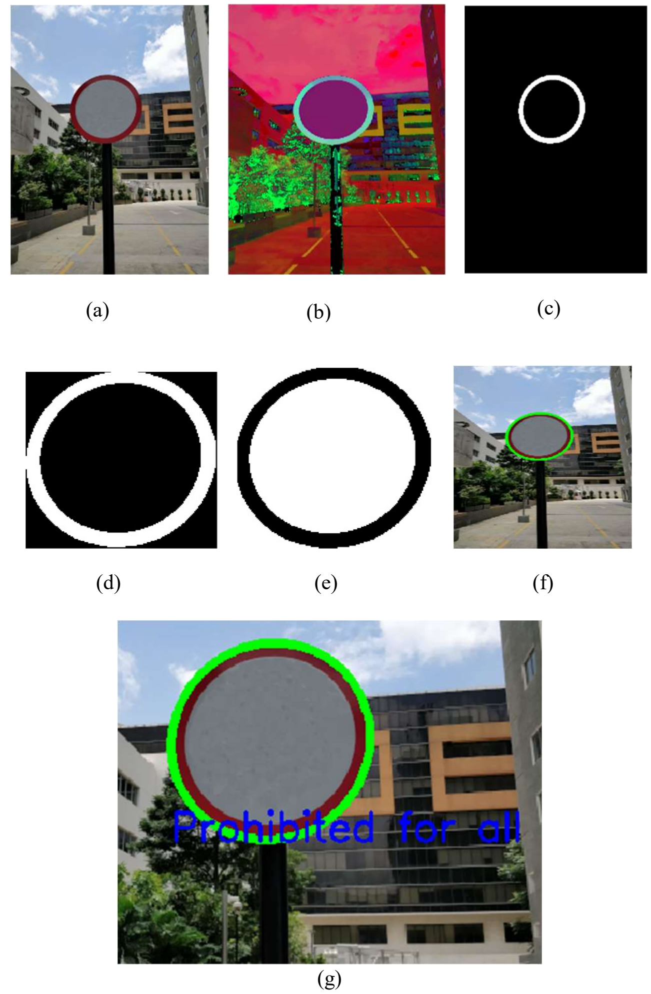


Fig. 4. Result images of the proposed model (a) Original Image (b) HSV Image (c) Threshold filter applied and contour detected (d) Bounding Box found (e) Validation through inverse threshold filter (f) Detected Sign highlighted with green contour (g) Classified Sign

Fig. 4 shows the different stages in Traffic Sign Detection and Recognition.

III. IMPLEMENTATION DETAILS

A. Data Sets

We used the German Traffic Sign Benchmarks along with the Belgium Data Sets for the training and testing of our CNN models. Our overall train set for triangles had 10582 images and test set had 3456 images. The train set for circles had 16106 images and the test set had 5277 images.

B. CNN Ensemble

To improve accuracy, we used an ensemble of 3 of the CNNs demonstrated in Table I. Each of these CNNs are trained with a random initialization of weights. Each of these 3 CNNs are later aggregated to form a single ensemble model by averaging the outputs of each CNN. The following table is a comparison of performances of 1 CNN and an ensemble of 3 CNNs. As seen in Table II, there is an increase in the accuracy of traffic sign recognition. The recognition rate for triangles increased by 2.81% and the recognition rate for circles increased by 0.39%.

TABLE II
COMPARISON BETWEEN 1 CNN AND AN ENSEMBLE OF CNNS

Model	Triangles	Circles
1 CNN	95.30%	98.79%
Ensembles of CNNs	98.11%	99.18%

An efficient way of training this model would be to train each of these models in parallel, thus reducing the time taken to train as well.

C. *Comparison with other papers*

TABLE III
ACCURACY OF DIFFERENT TSDR METHODS

Paper	Detection Process	Recognition Process	Accuracy
[1]	HSV thresholding + Noise removal + Modified Generalized Hough Transform	CNN	99.94%
[2]	Canny Edge Detector	SVM	98%
[7]	RGB color thresholding + edge detection + Connected Component Analysis	Multi Task CNN	90.2%
[8]		Fast R-CNN	94%
[9]	Color and shape	Multi Layered Perceptron	97.14% (Circles), 95.71 (Triangles)
[10]	HSV based color segmentation	CNN	97%
Ours	**HSV color thresholding and shape validation using Douglas-Peucker**	**CNN ensemble**	**98.11% (Triangles) 99.18% (Circles)**

All of the papers mentioned in Table III have different approaches and different accuracy results. As seen, our paper provides a very competitive accuracy along with a new approach. In [2], the authors have used only speed limit signs for both training and testing. The authors in paper [7], have focused only on Chinese traffic signs and have used a suitable dataset to accomplish that. Similarly, [10] focuses only on Bangladeshi signs. [1] uses the same dataset as ours, but their detection and recognition processes are different from ours.

Due to our multi step detection process involving color detection, shape detection and shape validation, a lot of sign-like images are eliminated, therefore providing us with more accurate objects to pass into the recognition phase. With the help of a Graphics Processing Unit (GPU) and training our 3 CNNs in parallel, the total time taken for training our model can be reduced by a lot, thus giving us an edge over the other papers.

IV. CONCLUSION

A. *Final Result*

Upon training our CNN ensemble, we achieved an accuracy of 98.11% for triangles and 99.18% for circles when tested.

TABLE IV
FINAL RESULT

Shape	Accuracy
Triangles	98.11%
Circles	99.18%

V. FUTURE SCOPE

While the model proposed in this paper does bring us a step closer to achieving the ideal Advanced Driver Assistance System or even a completely driverless system, there is a lot that can be improved.

For identification of a sign, this paper depends on color and shape of the sign. This is a problem if there is a reflection on the sign which impacts its color. Similarly, if the sign is chipped or cut off, the shape of the sign is impaired, thus resulting in no detection of the sign. Another important issue to consider is detection in the night. If the camera is not able to capture the environment in the night due to the darkness, the sign cannot be detected and classified. A text to speech module can also be added to this application. In the current application, the driver would have to read the text printed on the classified sign, but with the help of a voice module, more comfort is guaranteed.

The overall performance could also be improved and customized with the help of more datasets and from different countries.

VI. REFERENCES

[1] A. Shustanov, P. Yakimov, "CNN Design for Real-Time Traffic Sign Recognition," 3rd International Conference "Information Technology and Nanotechnology," ITNT-2017, 25-27 April 2017, Samara, Russia.

[2] R. Biswas, H. Fleyeh, M. Mostakim, "Detection and Classification of Speed Limit Traffic Signs," IEEE World Congress on Computer Applications and Information System, pp. 1-6, January 2014.

[3] G. Antipov, SA Berrani, JL Dugelay, "Minimalistic CNN-based ensemble model for gender prediction from face images," Elsevier, January 2016.

[4] "Douglas Peucker Algorithm" https://en.wikipedia.org/wiki/Ramer%E2%80%93Douglas%E2%80%93Peucker_algorithm

[5] "Keras Tutorial - Traffic Sign Recognition" https://chsasank.github.io/keras-tutorial.html

[6] A. Mammeri, A. Boukerche, M. Almulla, "Design of traffic sign detection, recognition, and transmission systems for smart vehicles," IEEE Wireless Communications, pp. 36-43, December 2013.

[7] R. Qian, B. Zhang, Y. Yue, Z. Wang, F. Coenen, "Robust Chinese traffic sign detection and recognition with deep convolutional neural network," IEEE 11th International Conference on Natural Computation, pp. 791-796, January 2016

[8] K. S. Boujemma, A. Bouhoute, K. Boubouh, I. Berrada, "Traffic sign recognition using convolutional neural networks," IEEE International Conference on Wireless Networks and Mobile Communications, pp. 1-6, December 2017

[9] M. Swathi, K.V. Suresh, "Automatic Traffic Sign Detection and Recognition in Video Sequences," 2nd IEEE International Conference on Recent Trends in Electronics Information & Communication Technology, pp. 476-480, May 2017, Bangalore, India.

[10] P. Dhar, Md. Z. Abedin, T. Biswas, A. Datta, "Traffic Sign Detection- A New Approach and Recognition Using Convolution Neural Network," IEEE Region 10 Humanitarian Technology Conference, pp. 416-419, December 2017, Dhaka, Bangladesh

[11] "The German Traffic sign recognition benchmark" http://benchmark.ini.rub.de/?section=gtsdb&subsection=news

[12] "keras api documentation" https://keras.io/

[13] "Belgium Traffic Sign Dataset" https://btsd.ethz.ch/shareddata/