TRAFFIC SIGN RECOGNITION SYSTEM USING CNN

Surya Teja[1]

***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. With the help of this Advanced Driver Assistance Systems (ADAS) application, drivers will no longer face the problem of understanding what the sign says. we can implement 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.**

I. INTRODUCTION

1.1 OBJECTIVE Traffic signs are devices placed along, besides, or above a highway, roadway, pathway , or other route to guide, warn, and regulate the flow of traffic, including motor vehicles, bicycles, pedestrians, equestrians, and other travellers .Since safety becomes more important for customers, Traffic Sign Recognition (TSR) becomes one of today's research subjects aiming to improve safety of driving. The theme of autonomous driving vehicles is becoming a concern for a larger and larger part of the industry. It goes without saying that the problem of autonomous automobiles is the most addressed, but some other types of vehicles benefit of this improvements, like drones. There are a lot of ideas and adaptations that are still needed for a completely autonomous, problem free vehicle, but in the meantime, some of our attention may be shifted from creating a fully autonomous automobile to the assistance of the driver. There is some pressure that can be lifted off the shoulder of the driver with the appropriate support, this contributing greatly to the driver's overall health and good disposition. One of this driver helping functionalities is represented by the traffic sign recognition and detection This project will present a method for road signs detection and classification. Its main objectives are, firstly, to recognize the meaning of the sign present in different images; secondly, to develop a method for traffic sign detection in a static image, or, preferably, in a video; lastly, to integrate both parts and provide real time traffic sign detection and identification.

[1]Department of Artificial Intelligence, Indian Institute of Technology Hyderabad, India.

[5]Department of Electrical Engineering, Indian Institute of Technology Hyderabad, India.

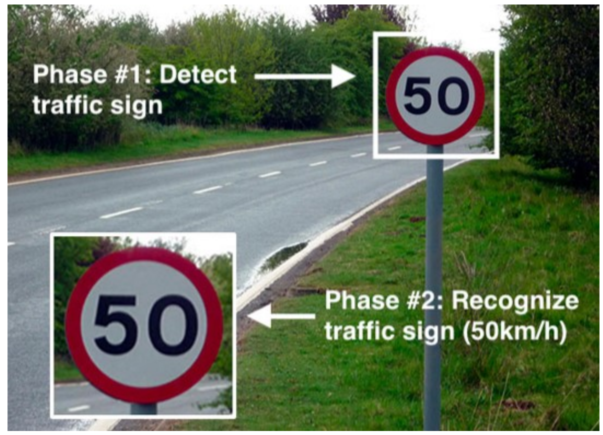


Fig. 1: Traffic Sign Detection and Recognition: (Phase-1)Detection; and (Phase-2) Recognition.

1.2 EXISTING WORKS Traffic Sign Recognition System can be divided into two sub-problems: traffic sign detection and traffic sign classification (see Fig. 1). Related work in Detection Detection methods based on color. [1] proposed a method that involves detection based on color, inside the RGB space , by thresholding. The corners are detected using a convolution of two functions: one which detects the gray levels around the corner and one which is called a "corner detector". Then the algorithm selects the points above the given threshold, finally, calculating the center of mass. If the sign is round, an approximation for rectangular sign detection algorithms is used. [2] used the HSV space , so that light doesn't contribute to affecting color detection. [3] proposed an interesting approach that is using the YUV color space, based on separation of luminance and and chrominance. The actual threshold are determined by analysing the distribution of data in the YUV color space. [4] Used High-Contrast Region Extraction , in order to extract regions of interest with high local contrast. Methods based on shape recognition, as color based recognition may have many different flaws, due to change in light and distance. [5] used Hough transform for detecting the closed contours of the traffic signs, yet this approach is sensitive to noise and occlusions. More articles develop learning based solutions, due to the recent popularity in machine learning, yet this require large amounts of data for training the models. [6] used the Viola-Jones algorithm for triangular, warning sign detection. The Viola-Jones algorithm uses increasing complexity detectors , each one being a set of classifiers based on Haar wavelet , AdaBoost being used as learning algorithm. The detector works by sliding a detection window across the image. When the window reaches the end of the image, it is enlarged and the processes is repeated. [7] used a method for traffic sign detection in which Non relevant areas are cropped if no interesting colors of the focus zone do not appear in other regions. Then, a concept called chromatic density value is defined for obtaining bounding boxes faster, with color information. There are several methods for classifying traffic signs, but the majority are based on learning algorithms. [8] used histogram of oriented gradients descriptors and distance transforms to evaluate the performance of K-d trees and random forests. [8] used Haar-like features for classifying images. It defines a function for mapping the features in images to sets of possible traffic signs. An alternative is given, so that the function is replaced by AdaBoost, working with Haar-wavelets. Deep learning algorithms enable a more precise classification, less dependent on the domain knowledge. [9], an architecture for a CNN is presented, starting from a traditional ConvNet. Multi-Scale features are presented: as opposed to a sequentially layered architecture, where output is fed only to succeeding layers, the given method is branched and fed to multiple stages. Several NonLinearities are proposed and discussed. The data used for training the models was the popular German Traffic Sign Recognition Benchmarks.

There are several methods for classifying traffic signs, but the majority are based on learning algorithms. A more difficult approach, more appropriate for experts, yet with limited representation power is based on hand-crafted features. For example, [17] used histogram of oriented gradients descriptors and distance transforms to evaluate the performance of K-d trees and random forests. In [8] Haar-like features are used for classifying images. It defines a function for mapping the features in images to sets of possible traffic signs. An alternative is given, so that the function is replaced by AdaBoost, working with Haar-wavelets. Deep learning algorithms enable a more precise classification, less dependent on the domain knowledge. In [14], an architecture for a CNN is presented, starting from a traditional ConvNet. Multi-Scale features are presented: as opposed to a sequentially layered architecture, where output is fed only to succeeding layers, the given method is branched and fed to multiple stages. Several NonLinearities are proposed and discussed. The data used for training the models was the popular German Traffic Sign Recognition Benchmarks. Several architectures were tested and the best one resulted in a performance of 98.97 percent accuracy, better than that of human performance (98.81 percent)

II. Materials and Methods

Public datasets of German Traffic Signs were used to train the deep networks CNN.The German Traffic Sign Benchmark is a multi-class, single-image classification challenge held at the International Joint Conference on Neural Networks (IJCNN) 2011. Traffic sign detection is a high relevance computer vision problem and is the basis for a lot of applications in industry such as Automotive etc. Traffic signs can provide a wide range of variations between classes in terms of colour, shape, and the presence of pictograms or text. In this challenge, we will develop a deep learning algorithm (CNN) that will train on German traffic sign images and then classify the unlabeled traffic signs

A. Data Description

For training and testing CNN models, we used the German Traffic Signs dataset[10] consists of 43 classes (Unique traffic sign images).Training Set has 34799 Images , Test set has 12630 images and the validation set has 4410 images.

B. DETECTION METHODOLOGY

The image is first preprocessed, so that the signs can be detected more easily, in a geometric manner. Preprocessing consists of -

1) CONTRAST: • Contrast is the difference in luminance or color that makes an object (or its representation in an image or display) distinguishable. In visual perception of the real world, contrast is determined by the difference in the color and brightness of the object and other objects within the same field of view. • The human visual system is more sensitive to contrast than absolute luminance; we can perceive the world similarly regardless of the huge changes in illumination over the day or from place to place. The maximum contrast of an image is the contrast ratio or dynamic range. • enhancing the contrast, so that colors can be recognized more easily;

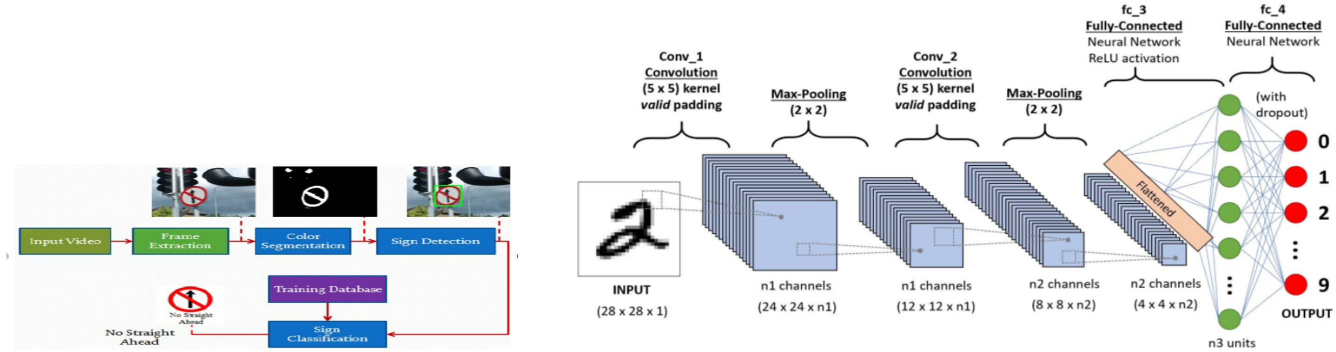
2) MASKING/FILTERING: • Image masking is the process of separating an image from its background, either to cause the image to stand out on its own or to place the image over another background. • In the old days of film stripping, it was done by cutting a physical "mask"–a sheet of material such as rubylith–in the shape of the image, and then projecting the image through it. Filtering the image in such a manner that only colors that may represent a traffic sign remain, the rest being faded to black;

3) Canny Edge Detection: The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986. Canny also produced a computational theory of edge detection explaining why the technique works.

4) BINARIZATION: Image binarization is the process of taking a grayscale image and converting it to black-and-white, essentially reducing the information contained within the image from 256 shades of gray to 2: black and white, a binary image. finally binarizing the image

5) REMOVAL OF SMALL COMPONENTS: After the image was preprocessed, the small components are removed

6) CONTOUR HANDLING: Contours handling is a process can be explained simply as a curve joining all the continuous points (along with the boundary), having same color or intensity. The contours are a useful tool for shape analysis and object detection and recognition. All the external contours are found and the sign having the greatest distance from its center to the contour is extracted from the image.



(a) (b)

Fig. 2: (a)Implentation Flow ; and (b) Architecture of CNNs .

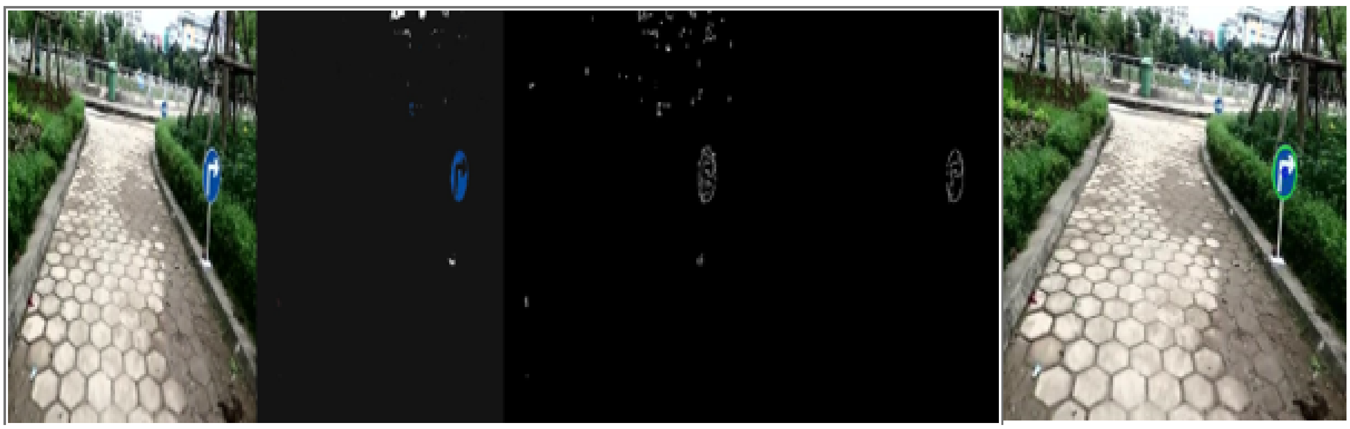


Fig. 3: Detection

7) CROP: Finally, a square image is cropped, based on the sign's contour.

C. CLASSIFICATION METHODOLOGY

One-of-many classification. Each sample can belong to one of C classes. The CNN will have C output neurons that can be gathered in a vector s (Scores). The target (ground truth) vector t will be a one-hot vector with a positive class and C1 negative classes. This task is treated as a single classification problem of samples in one of C classes.

1) TRAINING PHASE: See Fig-(5) and Fig-(6) The overall training process of the Convolution Network may be summarized as below: Step 1: We initialize all filters and parameters / weights with random values Step 2: The network takes a training image as input, goes through the forward propagation step (convolution layer, ReLU layer , pooling layer,batch normalization layer,drop layer,dense layer actions along with forward propagation in the Fully Connected layer) and finds the output probabilities for each class. Let's say the output probabilities for the image are [0.2, 0.4, 0.1, 0.3] Since weights are randomly assigned for the first training example, output probabilities are also random. Step 3: Calculate the total error at the output layer Step 4: Use Backpropagation to calculate the gradients of the error with respect to all weights in the network and use gradient descent(Adam optimizer) to update all filter values / weights and parameter values to minimize the output error. The weights are adjusted in proportion to their contribution to the total error. When the same image is input again, output probabilities might now be [0.1, 0.1, 0.7, 0.1], which is closer to the target vector [0, 0, 1, 0]. This means that the network has learnt to classify this particular image correctly by adjusting its weights / filters such that the output error is reduced. Parameters like number of filters, filter sizes, architecture of the network etc. have all been fixed before Step 1 and do not change during training process – only the values of the filter matrix and connection weights get updated. Step5: Repeat steps 2-4 with all images in the training set. The above steps train the ConvNet – this essentially means that all the weights and parameters of the ConvNet have now been optimized to correctly classify images from the training set.

2) TESTING PHASE : When a new (unseen) image is input into the ConvNet, the network would go through the forward propagation step and output a probability for each class (for a new image, the output probabilities are calculated using the weights which have been optimized to correctly classify all the previous training examples). If our training set is large enough, the network will (hopefully) generalize well to new images and classify them into correct categories.

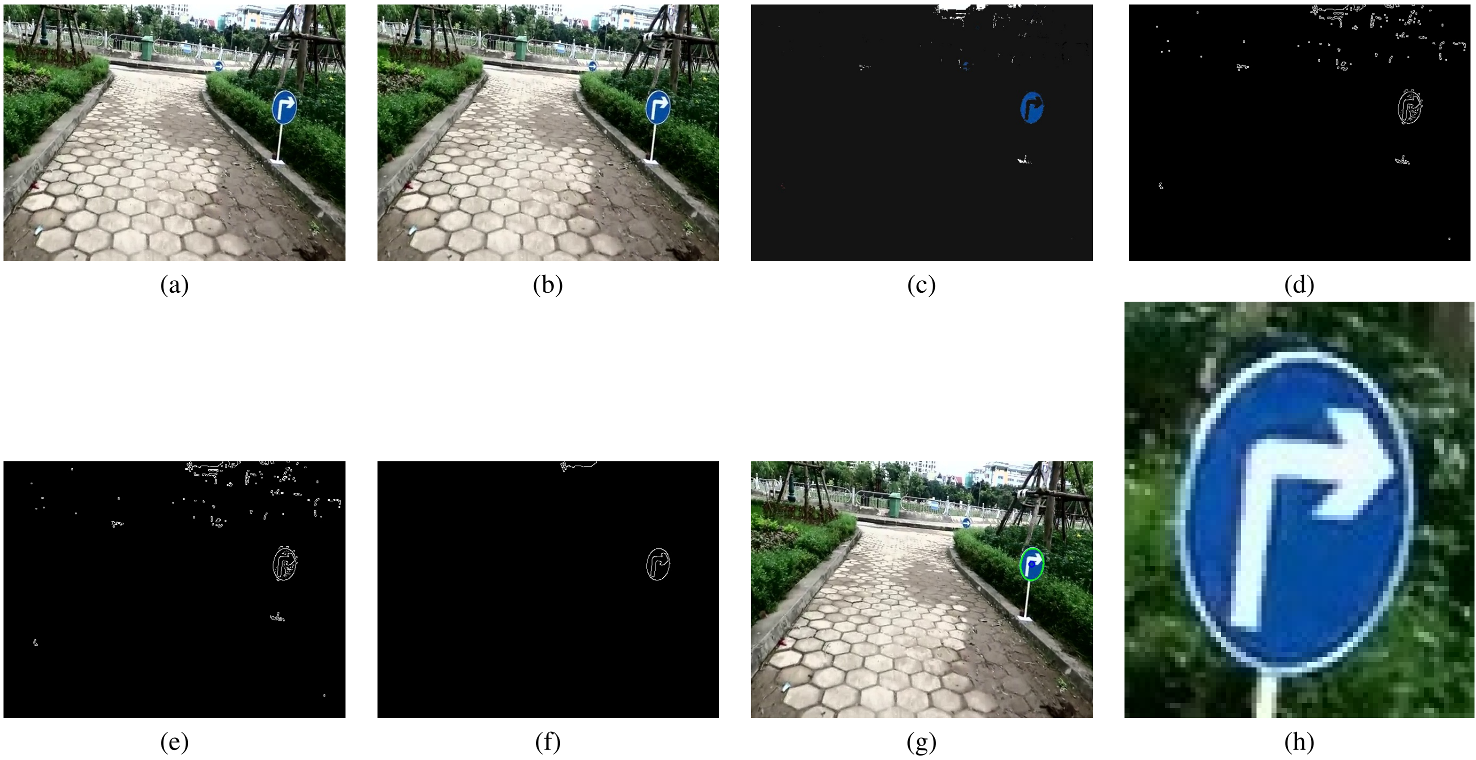


Fig. 4: (a) Original Image; (b) Contrast Image; (c) MASKING/FILTERING; (d) Canny Edge Detection;(e) BINARIZATION;(f) REMOVAL OF SMALL COMPONENTS;(g)CONTOUR HANDLING;(h) Cropped Sign Image;.

D. VOTING

When trying to classify an input image, more CNN models will be used, each of them voting three labels, based on prediction accuracy. In order for the label to be voted, the accuracy should be greater than a given threshold, by default 90 percent. The label with the greatest sum is the winner, if there exists one.

E. Outcome Prediction

Elsewhere, an ML model(CNN) has been developed based on German Traffic-Sign dataset to predict an incoming Traffic Sign Image. In this paper, we shall compare the performance of Different CNNs Architectures and Use Voting to classify Traffic-Signs.

III. Experimental Results

In this project, a traffic sign recognition system, divided into two parts, was presented. The first part is based on classical image processing techniques, for traffic signs extraction out of a video/image, whereas the second part is based on machine learning, more explicitly, convolutional neural networks, for image labeling See Fig-(5) and Fig-(6). Fig-(7) shows Evaluation Statistics of CNN Model-1 recognizing correct Traffic Sign in first choice;second choice;third choice; And Unrecognized.

When put together, the two parts work well, as the majority of the erroneous data sent by the detector is rejected by the classifier. Voting certainly helps (Used three models in this project), as more models that behave differently can classify an image and come up with the best statistical result.

Fig. 5: CNN model-1 Epoch Accuracy plot

IV. Conclusion

The Traffic sign in an image/video is predicted by the trained models (with an accuracy above 97 percentage each). In this project, a traffic sign recognition system, divided into two parts, was presented. The first part is based on classical image processing techniques, for traffic signs extraction out of a video/image, whereas the second part is based

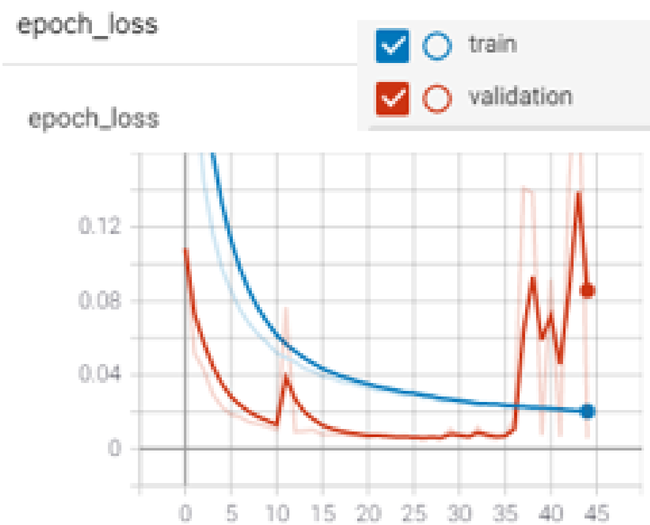


Fig. 6: CNN model-1 Epoch Loss plot

[H]

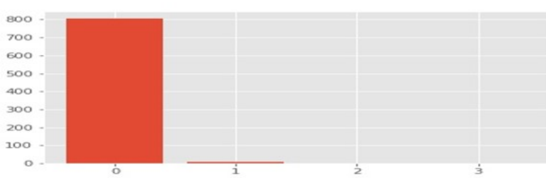


Fig. 7: Evaluation total statistics (0 - recognized first choice; 1 - second choice; 2 - third choice; 3 - others [unrecognized])

on machine learning, more explicitly, convolutional neural networks, for image labeling. When put together, the two parts worked well, as the majority of the erroneous data sent by the detector is rejected by the classifier. Voting certainly helps, as more models that behave differently can classify an image and come up with the best statistical result.

In Night conditions with flash or headlight illumination seem to be the best. The combination of low ambient noise due to lack of ambient light and the high visibility of the sign due to the reflected light makes the detector's job, and, as a result, the classifier's job too, very easy.

In Shade conditions , the project yield good results, as color is not de-natured and the masks do not cover the signs. The classifier has to reject a moderate amount of noise, most commonly rocks, a very clear blue sky or blue cars, and some patches of grass.

In Intense light conditions makes the detector skip some signs, their colors being denatured and, as a result, masked by the color mask. In that case, the classifier has no power, being able only to reject the noise it is fed with.

An improvement that can be brought to classification is a new class, consisting of different kinds of objects that are not traffic signs, so that the models may know how to handle noise better. This would be an alternative to using sigmoid.

Detection can be done using machine learning as well. Architectures like YOLO can detect and crop images from videos, as well as classify objects in those images, with outstanding speed.

References

[1] M. S. Arturo de la Escalera, Luis Moreno and J. Armingol, "Road traffic sign detection and classification. industrial electronics,," *IEEE*, 12 1997.

[2] A. Shustanov and P. Yakimov, "A method for traffic sign recognition with cnn using gpu," *SciTePress*, 2017.

[3] T. K. Jun Miura and Y. Shirai, "An active vision system for on-line traffic sign recognition," vol. E85d, p. 52 – 57, 02 2000.

[4] Z. C. C. Liu, F. Chang and D. Liu, "Fast traffic sign recognition via high-contrast region extraction and extended sparse representation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17(1), p. 79–92, 01 2016.

[5] M.-A. S. Miguel Garrido and E. Martm-Gorostiza, "Fast traffic sign detection and recognition under changing lighting conditions," *IEEE*, p. 811 – 816, 10 2006.

[6] K. Brki´c and A. Pinz, "Traffic sign detection as a component of an automated traffic infrastructure inventory system," 03 2020.

[7] F. S. R. J. R. O. W. R. st´ephane Estable, J. Schick and Y.-J. Zheng, "A real-time traffic sign recognition system," p. 213 – 218, 11 1994.

[8] B. S. Fatin Zaklouta and O. Hamdoun., "Traffic sign classification using k-d trees and random forests," p. 2151 – 2156, 09 2011.

[9] P. Sermanet and Y. Lecun, "Traffic sign recognition with multi-scale convolutional networks.," p. 2809 – 2813, 2019.

[10] "German Traffic-Sign Dataset(GTSRB)."