

CNN based Traffic Sign Detection and Recognition on Real Time Video

Mrs. Deepali Patil
Assistant Professor,
Shree L.R. Tiwari
College of Engineering

Ashika Poojari
Shree L.R. Tiwari
College of Engineering

Jayesh Choudhary
Shree L.R. Tiwari
College of Engineering

Siddhath Gaglani
Shree L.R. Tiwari
College of Engineering

Abstract - A road sign detection and recognition system is providing a comprehensive assistance to the driver to follow the traffic signs by developing a system that will automatically detect and recognize traffic sign, thus providing accurate information to the driving system. This system comprises of capturing signboards using camera installed in the vehicle. This captured image will undergo for analysis, processing and identifying the signboard. The detected sign board will be notified to the user by sending a notification. This paper proposes a system which will classifies different types of traffic signs in real time video.

Keywords - CNN; road sign; detection; recognition

I. INTRODUCTION

With the rapid development in societal and economical section, automobiles have become almost one of the convenient modes of transportation in each household. This makes the road traffic environment more complicated, and people expect to have intelligent Vision assisted system that provide drivers with traffic sign information, regulate operations of driver, or assist in vehicle control to ensure road safety. As one of the more important functions, traffic sign detection and recognition system, has become a hot research direction of researchers at home and abroad.

In the current traffic system there may be the probability of drivers missing the traffic sign because of traffic or even the drivers might ignore the traffic sign. With the continuous growth of urbanization this problem is only expected to grow worse.

A traffic sign detection and recognizing system can be applied to the vehicles where the system capture the traffic sign, detects them and recognizes the significance of the sign and informs the driver about the sign.

The efficiency of traffic sign detection and recognition is influenced by any factors like

- Color fading: This occurs due to long exposure to sun and rain.
- Vehicle motion: the vehicle motion may lead to camera jitter or blur the image.
- Weather conditions: The clarity of the image is varied by due to weather conditions such as heavy rain etc.

Traffic sign recognition is divided into two parts detection and recognition. The traffic signs have distinct color and particular shapes, which can be easily observed by drivers. Traffic sign detection is usually based on the inherent Characteristics of traffic signs such as shape and color. The color characteristics of traffic signs are more common, they usually mainly are red, yellow, and blue. Color enhancement

method is used to extract red, yellow and blue color spots, which focuses on the pixels of a given color are dominant over the other two color channels in color space of RGB. In the Lab and HSI color spaces are used to extract candidate signs. Meanwhile, the detected signs will help to extract more information, discarding the uninteresting or irrelevant area, connecting the scattered signs, and separate signs at the same location.

Detection methods based on the color characteristics have low computing, good robustness and other characteristics, which can improve the detection performance to a certain extent, but they depend on the corresponding threshold. Shape-based detection researches are generally based on the specific color and shape of the traffic signs, mainly are triangles, circles, rectangles and square. These approaches have a certain degree of robustness.

Stages involved in the system are:

- Image collection and extraction: Images of traffic sign boards are collected and extracted.
- Preprocessing: The traffic sign board image collected is preprocessed that it is cleansed. The image is cleansed by removing the noise from image. Cleansing involves Parsing, Correction, Matching, Consolidation, and Standardizing.
- Detection: This is classified into two parts color based and shape based.
- Recognition: The image collected of the traffic sign boards is compared with the image detected to match the template.

II. RELATED WORK

1) SURF is another popular method for developing a TSDR system. "Traffic Sign Detection and Recognition Using method based on feature and OCR" and "Traffic Sign Detection and Recognition Using Feature Based and SVM Method. Both use MESR and HSV for detecting traffic signs. This method does not change in motion blur and partial occlusion also. The system consist of a hardware implementation which runs at 20fps on 640x480 images.

2) Case where multiple appearances of sign: Multiple traffic sign appearing at a time and similar shape man-made object and obstacles can cause overlapping of signs and can lead to an error or false detection. The detection process can be affected by rotation, translation, scaling and partial occlusion. Li et al. in used HSI transform and Fuzzy shape.

3) This paper used SVM with a Gaussian Kernel to detect and recognize the speed limit signs, where images are extracted from an image database with 88.9% of success rate. Used shape like features for detection and CNN for classification which is especially effective in detecting speed limit signs in highways and this method is invariant in motion blur and partial occlusion also. The system includes a hardware implementation which runs at high CPU power and ram. This paper finds ROI Based on color and shape of traffic sign and also performs efficiently in low lighting conditions.

4) In this paper, the detection of the traffic signs is achieved by three main modules: pre-processing, detection, and recognition. In the pre-processing modules, the input are pre-processed to removed noise, enhance the image. In the detection phase, probable road sign are generated from the image. The image is segmented on the basis of color features. The output of this stage is a segmented image containing Regions of Interest which could be recognized as potential road signs. The segmented potential regions are extracted as input in the recognition stage. In the recognition stage, classification and recognition of detected signs is done by a Convolution Neural Network (CNN) this paper proposes a solution with 94% accuracy rate.

III. IMPLEMENTATION

Generally, traditional computer vision methods were developed to detect and recognize traffic signs, but this requires considerable and time-consuming manual work to extract important features in images. While applying deep learning to this problem, we create a model that efficiently classifies traffic signs images and learn to identify the most appropriate features for this problem by its own. In this paper, we have implemented a deep learning architecture that can identify traffic signs.

While using deep neural networks methods the model will require a large number of data and huge matrix multiplication operations which requires more computational power to tackle with this new type of algorithm that was introduced. This algorithm is called Convolutional Neural Network. It has been observed that CNN is more efficient and faster than a regular deep neural network for problems related to computer vision.

A Convolutional Neural Network are very similar to ordinary neural networks. They are made up of neurons with learnable weights and biases which is also known as supervised learning. All the basic idea learned or used for ordinary neural networks are also applicable to CNN's. The only difference between CNN and the ordinary neural network is that CNN assumes that input is an image rather than a vector. This vastly reduces the number of parameters to be tuned in a model.

Convolutional neural networks or CNN's are very important in the computer vision field. CNN help in running neural networks directly on images and are more accurate than many of the deep neural networks. Convolutional Nets models are easy and faster to train on images comparatively to the traditional models.

To train and test the model we have used German Traffic Sign Dataset (GTSRB) which contains more than 50,000

traffic sign images which is divided into 43 classes (e.g. Speed Limit 30km/h, No entry, Stop sign, etc.). This dataset is big enough which will help train model more accurately and help us achieve better results.

Steps Involved –

- Getting Data

The German traffic sign dataset can be downloaded from (<http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset>). We have used the Numpy library to calculate summary statistics of the traffic signs data set:

The size of training set is 34799

The size of the validation dataset is 4410

The size of test set is 12630

The shape of a traffic sign image is variable

The total number of unique classes in the data set is 43

- Dataset exploration and visualization

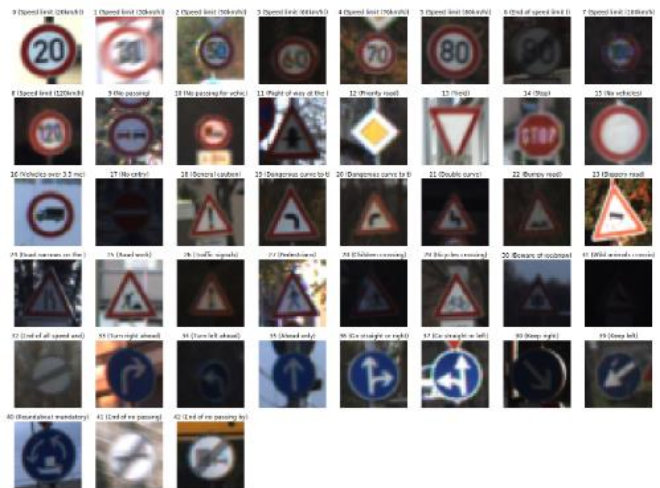


Fig 1: Example of Images from dataset

First we'll check the dimension of all the images of the dataset so that we can process the images into having similar dimensions. In this dataset, the images have a varying range of dimensions which ranges from 16*16*3 to 128*128*3 hence cannot be passed directly to the CNN model.

Next we will explore our data by plotting a distribution plot which will give us more insight on our data and number of images per class.

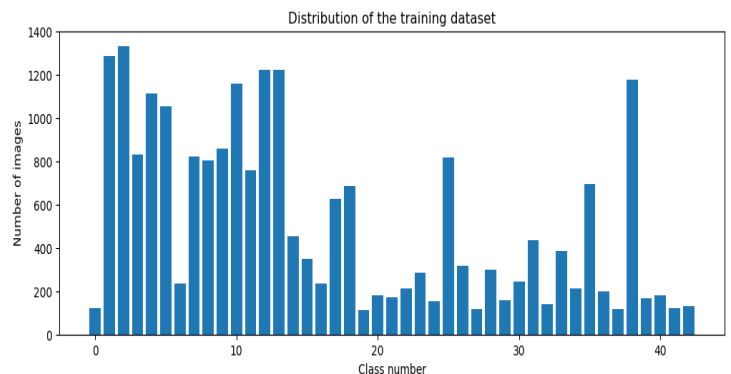


Fig 2: Distribution of images from Training data

There is also a significant imbalance across classes in the training set, as shown in the data histogram plot above. Some classes have below than 200 images, while others have greater than 2000. This means that our model could be biased towards over represented or more data in certain classes, especially when it is unsure in its predictions. Indeed, these images are samples which are extracted from real world environment. And our model have to handle all of these unusual conditions. So, it's better not to truncate our dataset in order to obtain data balance.

• Data Preprocessing

Firstly we need to compress or interpolate the images to a single dimension. To not compress much of the data and to not stretch the image too much we need to decide the dimension which is in between and save the accurate image data. So we have decided to resize every image to 32 x 32 x 3 dimension.

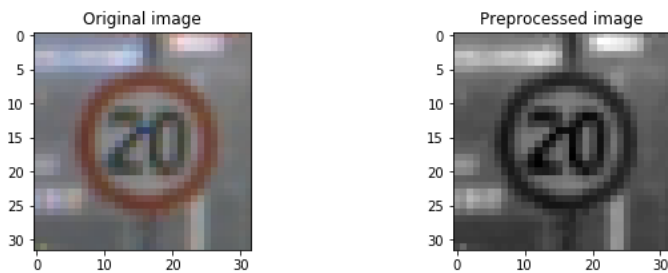


Fig 3 : Image before and after Greyscaling.

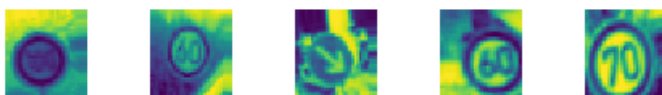


Fig 4: Augmented Images

Next we will convert this images to augmented images which will help our model to find more features in the images. Hence preprocessing is very crucial step as it reduces the amount of features and thus reduces execution time.

• Model Architecture

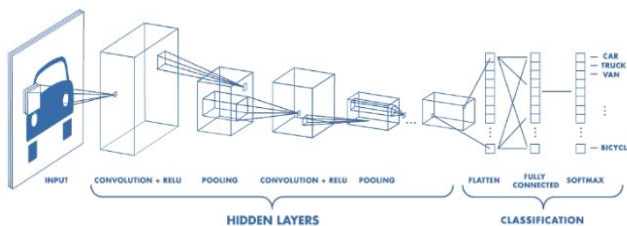


Fig 5: CNN Model

Same as the traditional neural networks convolutional neural networks is same as a sequence of layers. All the layers transforms an input image to an output image with some differentiable function that might include parameters. The CNN architecture consists of three types of layers: Convolutional Layer, Pooling Layer, and Fully-Connected Layer.

Layers	Description
Input Layer	32x32x1 images
Convolution-1	Convolution and rectified linear activation (ReLU).
Pool-1	Max pooling.
Convolution-2	Convolution and rectified linear activation.
Pool-2	Max pooling.
Local-3	Fully connected layer with ReLU
Local-4	fully connected layer with ReLU
softmax	Classification result

Table 1: Architecture of CNN

1. INPUT layer would hold the input image as a 3-D array of pixel values.
2. CONV layer will compute the dot product between the kernel and sub-array of an input image same size as a kernel. Then it will aggregate all the values resulted from the dot product and this will be the single pixel value of an output image. This process is repeated till the whole input image is covered.
3. RELU layer will apply an activation function $\max(0, x)$ on all the pixel values of an output image.
4. POOL layer will perform down sampling along the width and height of an image resulting in reducing the dimension of an image.
5. Fully-Connected layer will calculate the class score for each of the classification category.
6. In this way, CNN transform the original image layer by layer from the original pixel values to the final class values. RELU and POOL layers implement the constant function and the parameters are not trained at this layer. Parameters at FC and Convolutional layer are trained with the help of gradient descent optimizer.

• Training the model

To train the model, we have used an Adam optimizer with batch size 32 and number of epoch is 20. We followed a simple approach and ran only 20 epochs of the training and observed the validation error trying to set it on minimum level and also due to limitation of computational power. It is very important to consider mainly validation error while improving the model. Decreasing only the error with respect to training data can easily lead to unwanted model over fitting.

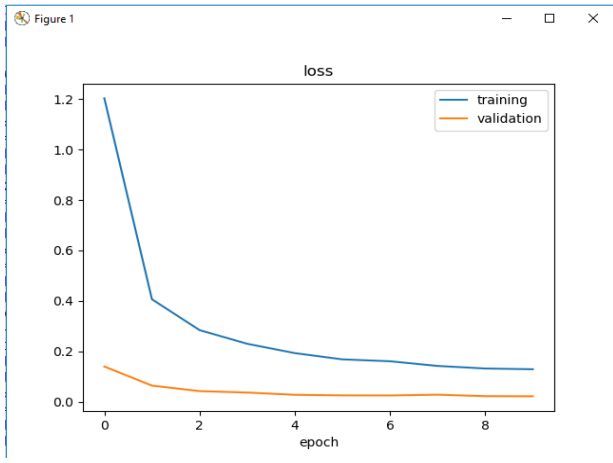


Fig 6: Loss Plot

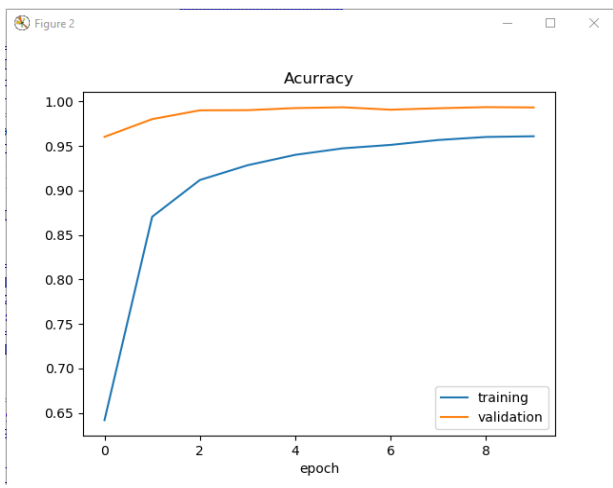


Fig 7: Accuracy Plot

After training the model up to 10 epoch with each epoch containing batch size of 2000 we are getting around 95% accuracy and low loss.

• Testing Model on new Images

In the end, we will test our Traffic Sign Recognition system on unseen traffic images. Surely, the accuracy obtained on the test set is also a very good sign that of a good model performance.

We can see that the results are very good. We have also collected some random images from google to test the model even further.



Fig 8: Testing Model on new Images

	A	B	C	D	E	F
1	ClassId	Name				
2	0	Speed limit (20km/h)				
3	1	Speed limit (30km/h)				
4	2	Speed limit (50km/h)				
5	3	Speed limit (60km/h)				
6	4	Speed limit (70km/h)				
7	5	Speed limit (80km/h)				
8	6	End of speed limit (80km/h)				
9	7	Speed limit (100km/h)				
10	8	Speed limit (120km/h)				
11	9	No passing				
12	10	No passing for vechiles over 3.5 metric tons				
13	11	Right-of-way at the next intersection				
14	12	Priority road				
15	13	Yield				
16	14	Stop				
17	15	No vechiles				
18	16	Vechiles over 3.5 metric tons prohibited				
19	17	No entry				
20	18	General caution				
21	19	Dangerous curve to the left				
22	20	Dangerous curve to the right				

Fig 9: labels.csv file (maps the class with corresponding sign)

- Testing on real time video using Camera

REFERENCES



Fig 10: Traffic sign image using webcam

Here we are testing our model on a real time video which is captured using laptop web cam or any external desktop camera. We suggest to use higher resolution camera to get better and clearer images which will eventually give us better results.

After we get traffic sign images we are processing every frame of the video (up to 30 FPS) and preprocess it first then input each image to our pickled model.

Our model will output the predictions and will show us the class name in which that image belongs to. Our system will also show the probability of the correctness of prediction.

IV. CONCLUSION

We have successfully implemented a Convolutional Neural Network to the Traffic Sign Recognition task with greater than 90% accuracy on average. We have covered how deep learning can be used to classify traffic signs with high accuracy, employing a variety of pre-processing and visualization techniques and trying different model architectures. We built a simple easy to understand CNN model to recognize road traffic signs accurately. Our model reached close to close to 90% accuracy on the test set which is good considering limitation of computational power and with a fairly simple architecture. There is still much work to be done, link including modern Deep Learning systems which use more recent and more complicated architectures like Google Net or ResNet. But obviously this comes in more computational cost, on the other hand.

- [1] Kaoutar Sefrioui Boujemaa, Afaf Bouhoute, Karim Boubouh and Ismail Berrada, "Traffic sign recognition using convolutional neural networks" in International Conference on Wireless Networks and Mobile Communications (WINCOM) 2017 IEEE
- [2] Amal Bouti, Mohamed Adnane Mahraz, Jamal Riffi, Hamid Tairi, "Road sign recognition with Convolutional Neural Network" in International Conference on Intelligent Systems and Computer Vision (ISCV). 2018 IEEE
- [3] Prashengit Dhar, Md. Zainal Abedin, Tonoy Biswas, Anish Datta, "Traffic Sign Detection- A New Approach and Recognition Using Convolution Neural Network" in IEEE Region 10 Humanitarian Technology Conference (R10-HTC) 2017 IEEE.
- [4] Yann Lecun, Leon Bottou, Yoshua Bengio, Patrick Haffner, "Gradient-Based Learning Applied to Document Recognition" in PROC. Of the IEEE, November 1998.
- [5] Faming Shao, Xinqing Wang, Fanjie Meng, Ting Rui, Dong Wang, and Jian Tang Real-Time Traffic Sign Detection and Recognition Method Based on Simplified Gabor Wavelets and CNNs
- [6] Shopa, P., Sumitha, N., & Patra, "Traffic sign detection and recognition using OpenCV" in International Conference on Information Communication and Embedded Systems (ICICES2014).
- [7] Faming Shao, Xinqing Wang, Fanjie Meng, Ting Rui, Dong Wang, and Jian Tang, "Real-Time Traffic Sign Detection and Recognition Method Based on Simplified Gabor Wavelets and CNNs" in PMC Sensors (Basel) 2018 Oct.
- [8] Mrs. P. Shopa, Mrs. N. Sumitha, Dr. P.S.K Patra, "Traffic Sign Detection and Recognition Using OpenCV" in ICICES2014 - S.A.Engineering College, Chennai, Tamil Nadu, India 2014 IEEE.
- [9] Hung Ngoc Do, Minh-Thanh Vo, Huy Quoc Luong, An Hoang Nguyen, Kien Trang, and Ly T.K, "Speed Limit Traffic Sign Detection and Recognition Based on Support Vector Machines" in 2017 International Conference on Advanced Technologies for Communications.
- [10] Xiong Changzhen, Wang Cong, Ma Weixin, Shan Yanmei, "A Traffic Sign Detection Algorithm Based on Deep Convolutional Neural Network" in 2016 IEEE International Conference on Signal and Image Processing.