# Deep Learning-Based Traffic Sign Recognition system

**Oisharya Dhar**
Department of Software System Engineering
University of Regina
oisharya.dhar@gmail.com

## Abstract

Automatic traffic Sign Recognition is an important part of automatic driving or self-driving vehicles. Proper traffic signs give important information about the nature of driving on a particular road. Moreover, even in old-fashioned vehicles, automatic traffic sign recognition can play a vital role in avoiding any kind of collision and it can make smooth traffic flow. And this traffic sign detection problem can be classified under object recognition and classification task. As part of deep learning, Convolutional Neural Network(CNN) has achieved very high accuracy in object recognition and classification task. To classify and recognize traffic signs accurately a deep learning-based model is used. A supervised learning-based algorithm with CNN and ReLU and SoftMax activation function is used to identify traffic signs of different classes. Data preprocessing also plays an important part to improve the accuracy of the model. The initial experiments show the model performs very well and achieves an accuracy above 90% for traffic signs with only 43 classes.

## 1 Introduction

From the beginning of the computer vision era, detecting and classifying images or objects held a massive role in the whole improvement of the Artificial Intelligence field as in deep learning. One particular part of deep learning which is the convolutional neural network(CNN) has shown very promising performance in the field of image classification and speech recognition etc. Recently, in self-driving or automatic driving vehicles, artificial intelligence (deep learning) is being used for many tasks. And its gaining more and more popularity day by day. After the use of machine learning in this field, it receives a great deal of attention to solve many existing problems. One of them is traffic signs detection and recognition which is very important in automatic driving. Traffic sign recognition is considered to be one of the most important parts of the automatic driving system as it is necessary to detect traffic signs before they can be identified.

Different traffic signs have different shapes, colours and instructions as they provide different information about route direction, traffic rules, road conditions, safety rules and warnings. The main object of designing an efficient automatic driving or advanced driver assistance system is to reduce accidents and to increase road safety.

Traffic signs can be divided into different classes with many subclasses depending on shapes and appearance though they hold different details. The whole process can be divided into two parts, detection and classification. With the advancement of deep learning the detection of traffic signs on the road become quite efficient and thus the next task becomes about classification which determines which specific kind of sign it is.

After the publication of the German traffic-sign benchmark(GTSDB) dataset [1], research in this area boosted heavily both detection and classification. Till today one of the main problems in classifying the traffic signs in the wild is a different kind of natural condition such as rain, snow, different angles mainly the illumination problem. In this project, this illumination problem of traffic sign classification is considered heavily.

## 2 Related Work

In general, traffic signs can be considered as the posters that are placed alongside the streets and these signs can be divided into 3 groups: regulatory, warning and informative [2]. With the advancement of the automobile, the automatic detection of traffic signs become one of the top priorities of researchers in that field. Detecting and recognizing traffic signs become very popular in recent researches. There are fixed shapes and colours of traffic signs which provide valuable information while driving in sense of different road conditions, traffic rules and route directions for the safety of the passengers and the road. This kind of work for detection and recognition is another sub-part of object detection that is very crucial when using advanced driving assistance systems and also in the use of autonomous vehicles [3]. Recently, many kinds of research for improving the functions of autonomous vehicles have been carried out mostly in the field of detection, segmentation and recognition[3], where their main focus has been on traffic, traffic signs, vehicles road signs [4] that are painted on the road. As new problems are arriving in this field every day, new researchers are trying hard and soul to solve all the incoming problems, such as insufficient illumination, partial rotation, blurring effect and serious deformation, making traffic detection a challenging problem.

With the launch of the German traffic-sign detection and classification benchmark dataset [5], a revolution started in this field of recognition of traffic signs. In Real-time Traffic Sign Recognition System with Deep Convolutional Neural Network, Jung et al. [6] emphasize the fact that different types of traffic signs should be recognized to get more detailed information about the road. For that reason, they only used 6 types of traffic sign images in training their model. They have used the LeNet-5 convolutional neural network architecture for training light-weight colour-based segmentation algorithm and Hough transform algorithm is applied to extract candidate regions of traffic signs for detection. The accuracy of their model is near to real-time accuracy but the main drawback is that the sign this system can recognize is very few.

In CNN Design for Real-Time Traffic Sign Recognition, Shustanov and Yakimov [7] proposed an algorithm for traffic signs localization in real-time. To extract coordinates of the data images Shustanov and Yakimov [7] have used the modified Generalized Hough Transform (GHT) algorithm. And for that reason, a simple classification algorithm can be used later. With this architecture, their model achieves 97.3% accuracy of traffic sign recognition. The drawback of their model is that this detection technique does not work well in low-resolution images and therefore in foggy or rainy weather the detection rate is very low.

In Traffic-Sign Detection and Classification in the Wild, Zhu et al. [8] have prepared a traffic-sign dataset that works as one of the benchmark datasets nowadays consisting of 100000 Tencent Street View panoramas. Using this dataset, they have proposed an algorithm using two Convolutional Neural Networks (CNN) which is robust and can simultaneously detect and classify traffic signs.

In Recognizing Traffic Signs Using a Practical Deep Neural Network, Aghdam et al. [9] an architecture for detecting and recognizing traffic signs by introducing a new CNN which is more accurate than the previously proposed. To achieve this more robust and effective method, they have used the Leaky Rectified Linear Units (ReLU) [10] activation function. They have also added a trainable linear transformation layer to the network in the input step which increases the efficiency. But they also faced the same problem which led to the wrong classification and that is because of occlusion or low-quality of the images.

## 3 Preliminaries

### 3.1 Network Structure

The whole method's structure can be dived into three parts: Dataset, Model and Evaluation Metrics. These three are the backbone of the whole project.

#### 3.1.1 Dataset

The dataset is downloaded from Kaggle[12] which contains images of a traffic sign with different illumination and blurring. The whole dataset is divided into three categories: training, validation and testing. The whole distribution of the can be seen in table 1.

| Class | Number of images |
|---|---|
| Training | 50000 |
| Testing | 12000 |
| Validation | 4000 |

Table 1: Details of the dataset

Figure 1, illustrates some training examples of training datasets that are affected by different kinds of blur and illumination factors.


Figure 1: Some Examples of training data

### 3.1.2 Model

The deep neural-based model contains different kinds of layers and functions such as a convolutional layer, fully connected layers as dense layer, max-pooling and so on. A brief overview of these functions are given below,

- Convolutional Layer works as a filter that strides through the whole input image (striding depending on filter size) and generates a feature map. The height and weight of the filters are smaller than the input image.
- Dense Layer or Fully connected layers are those neural layers where each neuron or node gets the input from all the neurons of the previous layer (flatten). Here, all the nodes are fully connected.
- Max pooling is the way of taking the maximum value from the patch of the feature map where the max-pooling kernel is set.
- Adam optimizer is an adaptive learning method. It computes individual learning rates for different parameters.
- LearningRateSchedular: It is a built-in function of Keras which helps to set the learning rate automatically. Here in the final model as call back this function is used.[13]
- Categorical cross-entropy loss measures the performance of a classification model whose output is a probability value between 0 and 1.
- Activation function:
  1. Rectified Linear Unit (ReLU) is used as the activation function for all the convolutional layers and the first two Dense layers in this model.
  The purpose of the ReLU is to give non-linearity to the model. It set all the negative values to zero and preserves only positive values. This function helps to reduce unnecessary noises from the data. [14]
  2. Soft Max activation function is used for the last dense layer in this model. The function is used when the output value is needed to be normalized. It converts the inputs from weighted sum values into probabilities that range from zero to one. [15]

The flow diagram of the model is shown in Figure 2.

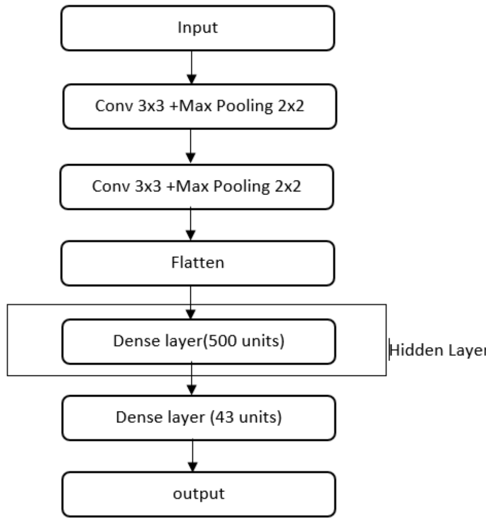


Figure 2: Flow diagram of final Model

There are two convolutional layers based on 3x3 filters with 2x2 maximum pooling. They are followed by only one hidden and dense layer of 500 neurons, and finally, the 43 neuron SoftMax layer to compute the probabilities.

### 3.1.3 Evaluation Metrics

The proposed model is evaluated using accuracy.

- Accuracy: A comparison of predicted class with correct class for all input images is done. And then the mean value among all the values is taken. On the other words, accuracy can be written as,

$$\text{Accuracy}=\frac{Number\ of\ correct\ Prediction}{Total\ Number\ of\ Predictiom}[12]$$

## 3.2 Training Process

The training process can be divided into two major steps, data pre-processing and training the model. The whole process takes approximately 10 to 12 hours of which the maximum time is taken for the training process.

### 3.2.1. Data Pre-processing

Before feeding the data to our model every image from the dataset is pre-processed. Firstly, the size of every image of the dataset was reduced to 32x32 size for ease of training. Then the dataset was divided into three parts: training, testing and validation. Then, normalization of all the images was performed. Because of this, every image got a 3-RGB channel. Thus finally, 50000 images were on the training set, 12000 images on the testing dataset and 4000 images on the testing dataset.

### 3.2.2. Training Model

The training process starts with feeding the three-channelled RGB input images as in the dataset to the convolutional layer. Both convolutional layers contain 32 filters with every filter consisting of three-channel(RGB) matching with the input image. Then ReLU activation function is then used to remove all the negative values and replace them with 0 from the pooled feature map. The pooling layer is applied to reduce dimension in both the convolutional layer. After that, a fully connected layer(hidden layer) is added with 500 neurons. The filter dimension for the convolutional layer is kept to 3x3 as it strides through the input training images. Finally, as there are 43 classes in the dataset, the output layer also consists of 43 neurons. For training, it takes 50 images at the same time. Then the testing images are fed to the trained network. The whole model is trained for 250 epochs. Finally, a built-in function from Keras is used which is the learning rate scheduler. This function updates the learning rate using current epochs and learning rate and then the updated learning rate is applied to the optimizer. Figure 3, shows the result after the convolutional layer using the 3x3 filters in the RGB channel.


Figure 3: Output of the convolutional layer before pooling

## 4 Experiment

For the experimenting purpose, the model is altered to see which specification gives the best accuracy for predicting traffic signs.

**Initial Case:** Initially in this project a very simple model to check which filter dimension gives the best accuracy. The model contains one convolutional layer, one hidden layer and an output layer. The model was then tested for different filter sizes. The filters are 3x3, 5x5, 9x9, 15x15,19x19, 23x23, 25x25 and 31x31. For the training, only 5 epochs are used as only in 5 epochs the accuracy gets quite high.

```
Model with filters 3x3, epochs=5, training accuracy=0.99331, validation accuracy=0.87256
Model with filters 5x5, epochs=5, training accuracy=0.98685, validation accuracy=0.87302
Model with filters 9x9, epochs=5, training accuracy=0.98230, validation accuracy=0.85374
Model with filters 13x13, epochs=5, training accuracy=0.97431, validation accuracy=0.83741
Model with filters 15x15, epochs=5, training accuracy=0.97092, validation accuracy=0.83333
Model with filters 19x19, epochs=5, training accuracy=0.96183, validation accuracy=0.83084
Model with filters 23x23, epochs=5, training accuracy=0.95233, validation accuracy=0.81565
Model with filters 25x25, epochs=5, training accuracy=0.95020, validation accuracy=0.82426
Model with filters 31x31, epochs=5, training accuracy=0.93776, validation accuracy=0.83628
```

Figure 4: The accuracy for every filter for training and validation

From figure 4, it is clear that filter size 3x3 and 5x5 provides the best accuracy. The validation accuracy for both of these filters is quite the same. But for training accuracy, the accuracy is higher for the 3x3 filter. Next, the time for prediction is shown to confirm which filter works the best.

```
data2 filter 3 classification time = 0.13144
data2 filter 5 classification time = 0.13647
data2 filter 9 classification time = 0.16948
data2 filter 13 classification time = 0.14723
data2 filter 15 classification time = 0.14047
data2 filter 19 classification time = 0.15163
data2 filter 23 classification time = 0.15308
data2 filter 25 classification time = 0.14198
data2 filter 31 classification time = 0.14159
```

Figure 5: Classification time for each filter

In Figure 5, it is clear that filter 5x5 takes more time than the 3x3 filter. For that reason for the next, all experiment the filter size is kept 3x3.

**Case1:** For case 1, the same model is used where there is only one convolutional layer, one hidden layer. But for the training process took 250 epochs. For the callback, the built-in learning rate function is used.

**Case2:** For case 2 experiments are done with the initial model which is exactly as same as the described 3.1.2 section. Only adding one extra convolutional and pooling layer to case 1. The training process took 250 epochs. For the callback, the built-in learning rate function is used.

**Case 3:** An extra dense layer is added to Case 2 model. All other elements remain the same as case 2. The training process took 250 epochs. For the callback, the built-in learning rate function is used.

**Case 4:** Case 4 is the same as case 3 only the early stopping function is used for callback. Here, validation loss with patience 6 is checked for stopping the training process and the training process stopped only at 7 epochs.

| Case number | Accuracy | Classification time/sec |
|---|---|---|
| Case 1 | 0.9024 | 0.058 |
| Case 2 | 0.921 | 0.048 |
| Case 3 | 0.010 | 0.043 |
| Case 4 | 0.010 | 0.039 |

Table 2: Comparison between different cases

From table 2, it is clear that case 2 achieves the highest accuracy among all the cases with relatively lower classification time. On the other hand, with added dense layer both cases 3 and 5 performs poorly and their accuracy is only 10%. Though both case 1 and case 2 achieve accuracy over 90% in both cases the classification time is less in case 2 than in case 1.

## 5. Results

From section 4, it is clear that Case 2 works the best for the given dataset. Now some results using this model are shown in figure 6. It is clear from figure 6 that most of the prediction is correct.


Figure 6: Some predicted signs with label

One major problem with this model is when the illumination is very poor or the images turn all black this model gives an unpredictable prediction without staying on only one prediction.

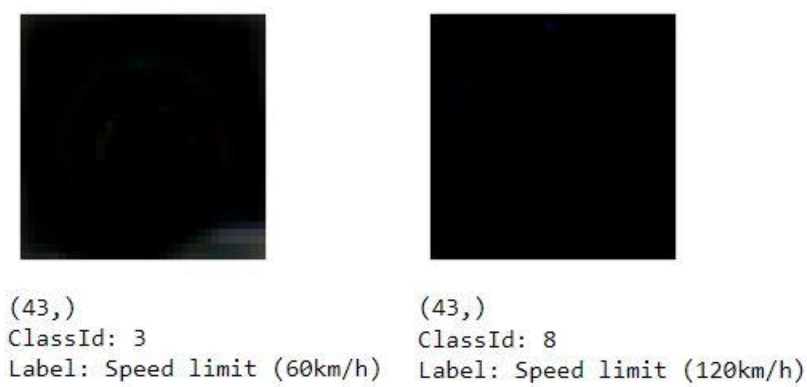


Figure 7: Wrong predictions

Figure 7 shows some wrongly predicted classes where the illumination is none. In two cases two different class was predicted.

## 6. Conclusion and Future Work

As the hype of automatic vehicles is increasing day by day, so does the need for proper and efficient techniques for driving. And to achieve this goal, traffic sign detection can play a very important role. For this reason, this paper proposed a deep learning method for detecting traffic signs in various illumination. It is evident from the result and experiences the method works efficiently in most cases and an accuracy of 92% is achieved with this model.

In future, a warning class can be added to the model if the input image is pitch black rather than predicting the wrong class. Moreover, the model can be tested on a vast dataset where the class number is even higher than this one. And finally to increase the accuracy different models with different parameters can be trained and tested.

## References


1. J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. Neural Networks, (0):–, 2012.
2. .J. E. Zafra, R. J. Moreno, and R. D. Hern´andez, "Comparison between backpropagation and cnn for the recognition of traffic signs," International Journal of Applied Engineering Research, vol. 12, no. 17, pp. 6814–6820, 2017.
3. T. Ahmad, D. Ilstrup, E. Emami, and G. Bebis, "Symbolic road marking recognition using convolutional neural networks," in 2017 IEEE intelligent vehicles symposium (IV), pp. 1428– 1433, IEEE, 2017.
4. G. L. Oliveira, W. Burgard, and T. Brox, "Efficient deep models for monocular road segmentation," in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4885–4891, IEEE, 2016
5. J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The german traffic sign recognition benchmark: a multi-class classification competition," in The 2011 international joint conference on neural networks, pp. 1453–1460, IEEE, 2011.
6. S. Jung, U. Lee, J. Jung, and D. H. Shim, "Real-time traffic sign recognition system with deep convolutional neural network," in 2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), pp. 31–34, IEEE, 2016.
7. A. Shustanov and P. Yakimov, "Cnn design for real-time traffic sign recognition," Procedia engineering, vol. 201, pp. 718–725, 2017.
8. Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2110–2118, 2016.
9. H. H. Aghdam, E. J. Heravi, and D. Puig, "Recognizing traffic signs using a practical deep neural network," in Robot 2015: Second Iberian Robotics Conference, pp. 399–410, Springer, 2016.
10. A. L. Maas, A. Y. Hannun, A. Y. Ng, et al., "Rectifier nonlinearities improve neural network acoustic models," in Proc. icml, vol. 30, p. 3, Citeseer, 2013.
11. V. Sichkar, "Traffic signs preprocessed," Aug 2019.
12. "Classification: Accuracy | Machine Learning Crash Course." Google, Google, https://developers.google.com/machine-learning/crash-course/classification/accuracy. Accessed 05-11-2021
13. L. Wen, X. Li, and L. Gao, "A new reinforcement learning based learning rate scheduler for convolutional neural network in fault classification," IEEE Transactions on Industrial Electronics, 2020.
14. N. J. Vickers, "Animal communication: when i'm calling you, will you answer too?," Current biology, vol. 27, no. 14, pp. R713–R715, 2017.
15. S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," towards data science, vol. 6, no. 12, pp. 310–316, 2017