# A Study on Neural Networks used for Traffic Signal Classification

ADITYA KULKARNI[1], OMKAR HASE[2], HETAVI GANDHI[3], KOPAL GANGRADE[4], SHWETA SHAH[5]

*[1, 2, 3] TE, Computer Engineering, Pune Institute of Computer Technology, Pune, India*

*[4, 5] Assistant Professor, Computer Engineering, Pune Institute of Computer Technology, Pune, India*

***Abstract- Traffic Recognition System (TRS) plays a key role in saving millions of lives every year. It recognizes the signs placed beside the road. Many a times such signs are partially covered leading to confusion. The TRS enables the drivers to identify these signs beforehand which in turn decreases the chances of an accident occurring. Technology behind the TRS has evolved over the years and Neural Networks (CNN) has been on the forefront of this evolution. CNN models are specifically used for object detection and are widely used. In this article we would analyse these CNN models along with a brief implementation of the same. The dataset used for this are highly regarded and have been the basis of several research papers.***

***Indexed Terms- Neural Networks, Computer Vision, CNN, Image Processing***

## I. INTRODUCTION

Ever since the evolution of Humans, one of the most important activities performed is that of Travelling. Humans travel from one location to other in search of comfort, better opportunities or because of pure joy. In the modern world, one of the major modes of travelling is that of Travelling by Road. Even though travelling is loved by many, most of us dread or avoid aimless travel. For helping us travel in a defined and safe manner, traffic signs were developed. But these traffic signs cannot be identified at times, due to them either broken, unclear, hidden behind objects, smeared with unwanted graffiti, etc. To make the job of recognizing traffic signs easier, the Traffic Recognition System or, TRS was developed. Upon the turn of the 20th Century, the rise of research on Autonomous Driving (ADAS) [3] has broadened the interest in the traffic recognition system. Along with the emergence of Machine Learning and its ever-widening reach, the need to automate recognition of traffic signs came at the forefront. Machine Learning has gone from single image classification to multi class classification. Not even stopping there, simpler Machine Learning methods like Boosting [4], Support Vector Machines (SVM), etc have become obsolete in the terms of object detection. The arrival of Neural Networks totally changed the way in which recognition of traffic signs was done. Convolutional Neural Networks or fondly abbreviated into CNN are today one of the topmost methods for TRS. One of the main aims of TRS is to try and prevent the number of unnatural deaths that are caused by driving accidents. Along with that due to rise of autonomous vehicles which is vying to be the backbone of the transport industry, it's important for both GPS and Traffic signs to be able to coordinate for smoother and swift transport.

This paper aims to analyse many of the previous CNN models developed on the Traffic Recognition System along with a brief implementation of a small-scale CNN which is both efficient as well as take less computational time.

## II. DATASETS

Ever since the rise of research in the domain of Autonomous Vehicles, the research done on TRS has grew exponentially. Its Enormity has been a led to creation of numerous datasets like the following: -

### *A. German Traffic Sign Recognition Benchmark*

This Dataset which is abbreviated into GTSRB is heavily used for Multi-Class Single-Image Classification purposes. It's one of the most trustworthy datasets for testing and validating traffic sign classification and detection algorithms. It was first used at International Joint Conference on Neural Networks (IJCNN) 2011.There a total of 43 various Traffic Signs which are present. These 43 signs are grouped together as Classes. It consists of over 50000 images (51,389 to be exact). The training images account to 39209 images, while Testing count is 12360.It divided the traffic signs in 3 categories: 1) Danger, 2) Prohibitory, 3) Mandatory and include Other as an irrelevant category.[1]

*B. Belgian Traffic Signs*

This Dataset is abbreviated as BTS. There are 63 classes in total. It consists of over 17000 images with the division of training images to that of testing images being approx. (10:7). The original image size is 128x128.

*C. Swedish Traffic Sign Dataset*

This is a dataset is based out of Sweden and consists of over 20000 images. There are over 3488 different signs present. It isn't as well curated as that of GTSRB where only the images of Traffic Signs are cropped out. But instead, a lot of diversified terrain including highways, forests, etc are also visible as the dataset was recorded from more than 350 km of Swedish roads.

*D. Laboratory for Intelligent and Safe Automobiles*

This dataset is developed in the USA containing a set of videos and annotated frames containing US traffic signs. 47 different US traffic signs are present. A total of 7855 annotations and 6610 frames are present in this dataset. These annotations contain both grayscale and colour images.[2]

## III. LITERATURE REVIEW

*A. Traffic Sign Classification Using Deep Inception Based Convolutional Networks*

This paper consists of model that used a modified version of the GoogleLeNet. The parent network for architecture consists of GoogLeNet with batch normalization while the Inception architecture is based on GoogLeNet. The model used the GTSRB dataset. The GTSRB dataset images were resized to 128 x 128.For activation, they employed PReLU rather than parameter-free ReLU, hence providing more accuracy. The accquired accuracy stands at an astonishing 99.81%. The main takeaway is the achievement of deformed input images.[5][12]

*B. Deep Learning for Large-Scale traffic sign detection and recognition*

This paper focuses on using a Mask R-CNN model. This Network is an extension of the Faster R-CNN method. The RCNN models are made up of 2 modules. Region-based CNN is one of the modules while the other one is called Region Proposal Network (RPN). The network architecture used in case of Mask R-CNN is ResNet(Residual Network).The Fast R-CNN model and Mask R-CNN model are merged together to create a unified network. For Mask R-CNN, the network architecture used is VGG16 while the ResNet is used for Fast R-CNN. The best result obtained is 95.5% accuracy for the Mask R-CNN with Adaptation and Data Augmentation. This model is based on the dataset called DFG Traffic Sign Data Set. This dataset is based on the data collected in the EU
country of Slovenia.[6]

*C. Indian Traffic Sign Detection and Classification Using Neural Networks*

The majority papers written on the topic of TRS use the public datasets curated from Europe (STSD, GTSRB, etc). This paper focuses on Traffic Signs procured from India. The process of detection and classification consists of 3 steps:1. Image colour segmentation, 2. Blob Detection to attain the Region of Interest, 3. Classification using Multiple Neural Networks. One of the major feats of this paper is that of Improving the technique of recognizing traffic signs in poor lighting and partially obstructed images.[7]

*D. Traffic Sign Recognition With Hinge Loss Trained Convolutional Neural Networks*

This paper uses the Hinge loss stochastic gradient descent (HLSGD) method to train its CNN model. While training the model, its observed that the rate of decrease in errors is about 35.19%. The HLSGD method gives a steady convergence rate and a better test accuracy. The main advantages of this method, is model generalization's improvement. Some Data Augmentation [8] was performed on training data which is there after added to the training set. The accuracy of the model is a staggering 99.65%.[10]

*E. Novel Deep Learning Model for Traffic Sign Detection Using Capsule Networks*

This paper focuses on a capsule network-based architecture for training its model. The proposed model was able to generate accurate results up to the tune of 97.62%. Capsules are a group of Artificial Neural Networks. These ANN's help in the better performance complicated internal computations. This model definitely consists of multiple layers. The input image which is fed into the system receives a small part of the receptive field. A vector is the respective output of a Caps Net. The model has showcased that it performs better at detecting pose and spatial variance when compared to CNN's and is deemed to be pretty reliable and accurate.[17]

## IV. CNN - (CONVOLUTIONAL NEURAL NETWORKS)

Convolutional Neural Networks is a Deep Learning Algorithm. This Algorithm takes in an Input Image and applies weights and biases to the various objects in the image. These weights and biases are used to differentiate one from the other. The amount of pre-processing required for the CNN is very less in comparison to other algorithms. CNN reduces images to a form where its easier to process without losing it's features. A CNN model is mostly preferred over a ANN model as it takes less computation and isn't sensitive to location of an object in an image.

*A. Kernel*

A kernel is defined as the extract various feature of from the given input images. A kernel is a matrix, which is slid across the image and multiplied with the input such that the output is enhanced in a certain desirable manner. Kernels are the building blocks of a filter which is the concatenation of multiple kernels. Each of this kernel is assigned to a particular channel of the input image.

*B. Convolutional Layer*

In the convolutional layer, the main objective to extract local patterns. The kernels play a major role, in this layer. The kernels are made up of a 3-D matrix along with a bias. The kernel size is equal to the height and width of the matrix. The source image has numerous channels (RGB or HSV) but in case of the output generated from the kernel, there's only a single channel. The dimensions of the input and output image remain the same. The output generated is the dot product of an input image and a bias where both of them are in a form of a matrix. Normally the aim of the first convolutional layer is to capture the low-lying features of the image which includes the edges, colour, gradient orientation. Upon the addition of the added layers the detection of even finer details is executed.

*C. MaxPool Layer*

The Pooling layer is added in an CNN to solve the issue of precise positions which differ from in training instances having same labels. The Pooling layer decreases the map's resolution.[9] MaxPooling is the type of Pooling which only selects the maximum value from the image covered by the kernel as well as discarding noisy activations. These Maximum Values are stored in the Output Image, while all the small values are dropped. This layer normally follows a non-linear layer and neither does it any borders but its doesn't even change its channel count. This helps in benchmarking the order of any CNN which is normally convolutional layer, nonlinear layer and finally the Pooling layer.[16]

## V. IMPLEMENTATION

*A. Dataset*

We made use of the German Traffic Sign Recognition Benchmark (GTSRB) dataset which contains 39209 training images and 12360 testing images. It consists of 43 classes.

*B. Model*

We propose a model consisting of 4 Convolution layers. 2 Convolutional layers are paired together after which a MaxPool layer is applied to avoid over-fitting. After all the 4 Convolutional layers, we have Flatten layer which paves the way for 2 Dense layers, in between whom we have a Dropout layer. The total params of our model is 2,511,467 while the batch size of the model was 32. For our 4 Convolutional Layers we make use of the default (3 x 3) kernel along with an application function of ReLu(Rectified Linear). The MaxPool2D layer has a pool size of (2 x 2). For our Dropout layer, our dropout rate is 0.8. Finally, our model ends with last Dense layer which has a Softmax Activation Function.

*C. Activation Layer*

In our Model we make use of 2 Activation Layers that are as follows –

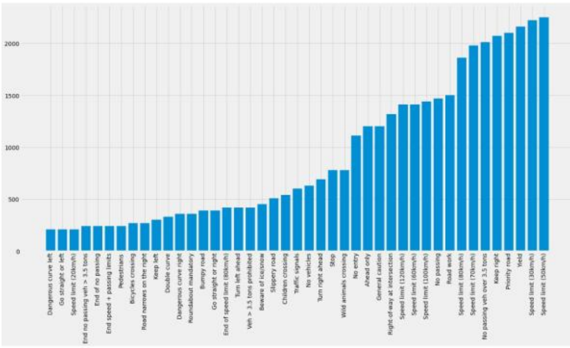

Fig. 1. Fig 0. Image Classes and its count.

1) *Rectified Linear (ReLU):* The Rectified Linear Activation Function or better known as ReLU is one of the most commonly used activation function. Its majorly used in multilayer neural network. It's a non-linear in nature. The main reason why its chosen over the sigmoid or hyperbolic tangent functions is because of its reduction in computation time. The ReLU function gives an output of 0 if the input is below 0.

While it calculates the max between 0 and the given input.[14]

2) *Softmax:* The Softmax Activation is being used in the last layer in our model. The Softmax activation is used to fit the output for each of the given class in between 0 and 1. So in our model, we try to fit each output of the 43 classes into the range of 0 to 1. Softmax layer is non-linear in nature. This main use of Softmax is when we have a multi-class classification.

*D. Loss Function*

The Loss Function used in this model is the famous Categorical Crossentropy. This function is mainly used in multi-class classification tasks. The calculated loss is a pretty good measure to distinguish between two discrete probability distributions. The Softmax function is the only activation function used with Categorical Cross Entropy Loss Function.

*E. Optimization Algorithm*

The Adam Optimizer Algorithm is used in this model. The main aim of this Optimization function is used to update network weights iterative based in training data. Adam is very straightforward when it comes to implementation. Its memory requirement is minimal and is computationally efficient. There are 4 configuration parameters:

1) *Alpha:* It's referred as the step size or the learning rate.
2) *Beta 1:* It's the exponential decay rate for the first moment estimates
3) *Beta 2:* It's the exponential decay rate for the second moment estimates.
4) *Epsilon:* This value is used to prevent any division by zero during the implementation.[13]

*F. Results*

The model was tested on different environments. We used Kaggle notebooks for developing, training and testing our

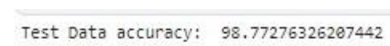


Fig. 2. Obtained Test Accuracy

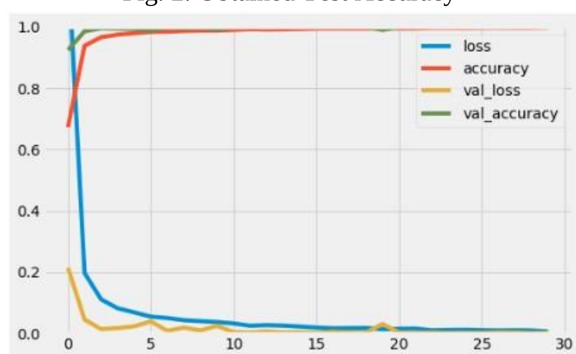


Fig. 3. Training Loss, Accuracy and Validation Loss, Accuracy

model. Along with the Kaggle Environment, we used the local Jupyter notebook for testing purpose. In the Kaggle notebook we made use of Nvidia P100 GPU chip for an efficient computational time. The Jupyter notebook was ran in environment of consisting of an Intel 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz,8 Cores alongwith a Nvidia GeForce RTX 3050 Chip and a 16GB RAM. Each Epoch took roughly 16.9 seconds. The total number epochs were 20. The training loss was converged down to 0.109

from 1.7739 while in the case of validation data it narrowed down from 0.0987 to 0.0025. The validation accuracy obtained was an astonishing 99.9%. Similarly the Training accuracy retrieved was at 99.66%. Upon running the model on the test data, we acquired an accuracy of 98.77%.

## CONCLUSION

The main motivation of this paper was to develop a CNN model used to recognize the traffic signs. The sheer volume of the dataset led to an increased the duration of training the model. The photos of the signs being blurred or being cropped, didn't make the pre-processing of the model any easier. The above model was able to learn both multiple classifiers and features. In the model we explained each layer in great detail. Each Parameter from Optimization Function to Activation Function has been meticulously explained. The training speed time has averaged out to 16.9 while the accuracy has 99.66%. The Final accuracy obtained is 98.77%. The Model was to trained on the famous German Sign Traffic Recognition Benchmark (GTSRB), which even though being pretty large resulted in easy, efficient computation as well as a stable convergence.

## ACKNOWLEDGEMENT

This undertaking could not have been completed without the participation and assistance of so many people whose names may not all be enumerated. Their contributions are sincerely appreciated and gratefully acknowledged. We would like to express our gratitude to Ms. Kopal Gangrade (Professor), Department of CE, Pune Institute of Computer Technology, for her cooperation, guidance, and supervision. Besides our guide, we would like to thank the HOD of the Computer Department, Mrs. Geetanjali Kale. We extend our gratitude to Pune Institute of Computer Technology for giving us this opportunity. We also acknowledge with a deep sense of reverence, our gratitude towards all our friends who directly or indirectly helped us to complete this research paper.

## REFERENCES

[1] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. *The german traffic sign recognition benchmark: a multi-class classification competition.* In Neural Networks (IJCNN), The 2011 International Joint Conference on, pages 1453–1460. IEEE, 2011.

[2] Andreas Møgelmose, Mohan M. Trivedi, and Thomas B. Moeslund, *Vision based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey,"* IEEE Transactions on Intelligent Transportation Systems, 2012.

[3] Yanjun Fan and Weigong Zhang, *Traffic sign detection and classification for Advanced Driver Assistant Systems*, 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 2015, pp. 1335-1339, doi: 10.1109/FSKD.2015.7382137.

[4] A. Ruta, Y. Li, and X. Liu, *Robust class similarity measure for traffic sign recognition*, IEEE Trans. Intell. Transp. Syst., vol. 11, no. 4, pp. 846–855, Dec. 2010.

[5] Haloi, Mrinal, 2015. *Traffic sign classification using deep inception based convolutional networks.* CoRR, abs/1511.02992

[6] D. Tabernik and D. Skocaj,ˇ*Deep Learning for Large-Scale TrafficSign Detection and Recognition*, in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 4, pp. 1427-1440, April 2020, doi: 10.1109/TITS.2019.2913588.

[7] A. Nandewal,A. Tripathi and S. Chandrra , *Indian Traffic Sign Detection and Classification Using Neural Networks*,2nd International Congress of Technology, Management and Social Sciences-16 (ICTMS-16)

[8] A. Krizhevsky, I. Sutskever, and G. Hinton, *"Imagenet classification with deep convolutional neural networks,"* in Proc. Adv. Neural Inf. Process. Syst., 2012, vol. 25, pp. 1106–1114.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, Proc. IEEE, vol. 86, no. 11, pp. 2278– 2324, Nov. 1998

[10] J. Jin, K. Fu and C. Zhang, *Traffic Sign Recognition With Hinge Loss Trained Convolutional Neural Networks*, in IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 5, pp. 1991-2000, Oct. 2014, doi: 10.1109/TITS.2014.2308281.

[11] Timofte, Radu, Karel Zimmermann, and Luc Van Gool. *Multi-view traffic sign detection, recognition, and 3D localisation.* Machine vision and applications 25.3 (2014): 633-647.

[12] Chaudhari, Tejas and Wale, Ashish and Joshi, Amit and Sawant, Suraj, *Traffic Sign Recognition Using Small-Scale Convolutional Neural Network*, (May 3, 2020). 2nd International Conference on Communication and Information Processing (ICCIP) 2020

[13] Kingma, Diederik & Ba, Jimmy.*Adam: A Method for Stochastic Optimization.* International Conference on Learning Representations, (2014)

[14] Agarap, Abien Fred. (2018). *Deep Learning using Rectified Linear Units (ReLU).*

[15] J. Bouvrie, Notes on convolutional neural networks, 2006

[16] D. Scherer, A. Muller, and S. Behnke,¨*"Evaluation of pooling operations in convolutional architectures for object recognition,"* in Artificial Neural Networks—ICANN. Berlin, Germany: Springer-Verlag, 2010, pp. 92–101

[17] Kumar, Amara Dinesh.*"Novel Deep Learning Model for Traffic Sign Detection Using Capsule Networks."* ArXiv abs/1805.04424 (2018): n. pag.