International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 05 | May 2022 www.irjet.net p-ISSN: 2395-0072

Traffic Sign Recognition System

Saloni Pathak#, Rutuja Rane#, Geeta Chavan#, Sumedh Pundkar#

#Department of Computer Science and Technology, SNDT Woman's University, Mumbai, India

---

Abstract— *Recognition of traffic signs is an important factor in applications such as self-driving cars, traffic mapping and traffic surveillance. Deep Learning models help in automated driving for Traffic Sign Recognition. In this paper, the model is trained with pre-processed RGB images and the saved model is used to classify the image provided.*

Keywords—**Convolution Neural Network, Dropout Layer, Traffic Sign Recognition, Classification, Batch Normalization.**

I. INTRODUCTION

The advancement in technology has led to many evolutions in every field and aspect of life. One such is Convolution Neural Network (CNN or ConvNet) which is ideal and most accurate for image: processing, detection, classification. The same can be done using various architectures available namely AlexNet, VGGNet, GoogLeNet, and ResNet. The images are captured from the front-cameras of vehicles and are processed to give out instructions for the driver or assist in automated cars. Similarly, the system can scan and compare the speed of car with the traffic sign displayed, further informing driver to slow down to avoid over speeding.

Hence, we proposed a method using convoluted layers along with other layers to detect and identify the traffic signs.

II. RELATED WORK

Traffic Sign Recognition has been researched upon for the last decade. Various techniques have been employed to classify the images. Recently, deep learning techniques involving Convoluted Neural Networks have been used for image detection, classification and localization. Traditional approaches haven't proved the accuracy comparable to what humans require. Improvements have been made to increase the classification accuracy using deep learning techniques and exceed the average human accuracy.

Next we study the prior work done on CNN designed to recognise traffic signs.

A. CNNs Designed for Classification

The authors in [3] propose a model consisting of basic convolution features with supervised learning technique. It consists of multiple convolution layers and subsample layers, followed by average pooling and fully connected layers. The model is trained with gray-scale images and the Adam optimizer is used to boost the accuracy. Batch normalization is used in [1] after each convoluted layer to prevent vanishing of gradient during back propagation across various layers.

The model [2] uses a modified version of the generalized Hough transform to localize the pre processed images. Point-like noise in the image that occurs during the preprocessing was removed with the help of noise removal algorithm.

One of the most interesting results was seen when the multiple network blocks were used. In [4], the model uses a six-layers neural network that consists of two convolution layers with three subsampling layers and two branched subnets, all followed by a subsampling layer in the end. The subnets layers are trained with different parameters. The weighted cumulative sum of classifiers and alternating data augmentation during model training increases the accuracy of the model.

The authors in [5] proposed a classic LeNet-5 architecture [8] and scikit-learn pipeline framework to classify the images. HOG (Histogram of Oriented Gradient) was used to preprocess the images which allow the identification of dominant gradients in the image. The LeNet-5 architecture consists of two sets of convolution and average pooling layers, followed by flattening convoluted layers, a fully connected layer and a softmax classifier.

Taking into account all the problems faced by the above models, we propose a model based on the classic LeNet architecture. The model is trained with Adam's optimizer and once the required accuracy is reached, a confusion matrix is drawn to help retrain the model.

III. DATASET

The dataset used for this project is the German Traffic Sign Detection benchmark. The dataset consists of 43 traffic sign classes with 51,830 images each of dimension (32x32x3), 3 signifying RGB color channels. The dataset was further split into 27441 training, 12,631 validation and 11,760 test images.

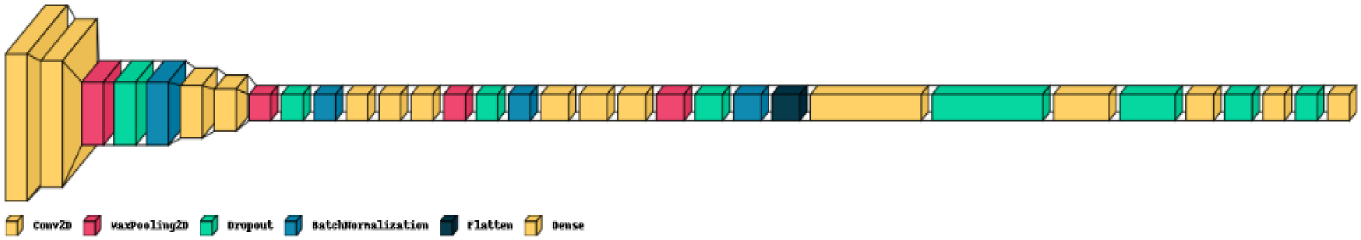


Fig.1 Network architecture

IV. Proposed Methodology

In this section, the data preprocessing techniques as well as implemented CNN based architectures alongside training details and evaluation metrics are discussed in detail.

A. *Data Preprocessing*

The proposed system for Traffic sign classification includes 32x32 RGB images of the traffic sign boards. Since the base research paper hypothesizes that raw YUV may not be an optimal input format, the dataset is pre-processed using various image processing techniques. All the images have been passed through rotation, random: translations, zooms and shearing, inverting before feeding it to CNN.

Fig. 3 depicts the augmented dataset which has been normalized after pre-processing. All of the pre-processed images are dumped into a pickle file before being reloaded for training the CNN model.

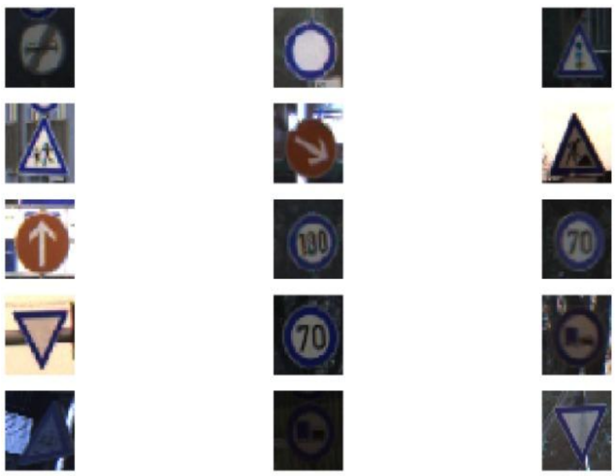


Fig. 2 Normalized Dataset

B. *Network Architecture for Training Stage*

Fig. 1 showcases the architecture of the model used. Apart from aiming for faster and better training, we had to resolve the dying ReLU issue which happens due to high learning rate or presence of a large number of negative biases during the training.

The model contains 4 VGGNet Block. Each VGGNet Block consists of 2 Convolution layers, 1 Max pooling layer, 1 Dropout layer and 1 Batch Normalization layer.

After consideration, we decided that each VGG network block contain the following elements; Batch Normalization for faster and better training; ReLU to solve dead linear rectification issue; convoluted layer. Also a fully connected layer and a pooling layer with max pooling 2D operation.

We used three VGG network blocks as follows:

- First VGG Layer extracts 32 Feature Maps
- Second VGG Layer extracts 64 Feature Maps
- Third VGG Layer extracts 128 Feature Maps
- Fourth VGG Layer extracts 256 Feature Maps

The final softmax layer has 43 outputs, corresponding to each category in GTSRB [7].

The structure of the networks and the hyper-parameters were empirically initialized based on previous works using ConvNets [8]. Then we set up a cross-validation experiment to optimize the parameters of network architecture, with details shown in Table I.

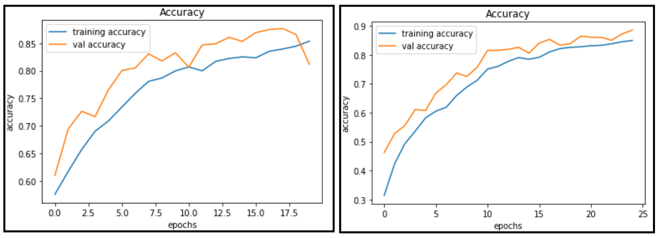
A processed input images are passed through CNN model to extract relevant representational features is adopted to predict the details of the labels [7].

Layer	Type	Feature Map & Size	Kernel
1	Input	1x32x32	
2	Convolution C_1	32x28x28	5x5
3	Convolution C_2	32x28x28	5x5
4	Max Pooling M_1	32x12x12	2x2
5	Dropout D_1		
6	Convolution C_3	64x10x10	3x3
7	Convolution C_4	64x8x8	3x3
8	Max Pooling M_2	64x4x4	2x2
9	Dropout D_2		
10	Convolution C_5	128x3x3	2x2
11	Convolution C_6	128x2x2	2x2
12	Convolution C_7	128x1x1	2x2
13	Max Pooling M_3	128x1x1	1x1
14	Dropout D_3		
15	Convolution C_8	256x1x1	1x1
16	Convolution C_9	256x1x1	1x1
17	Convolution C_{10}	256x1x1	1x1
18	Max Pooling M_4	256x1x1	1x1
19	Dropout D_4		
20	Dense (Flatten)	256	
21	Fully Connected FC_1	1024	
22	Dropout D_5	1024	
23	Fully Connected FC_2	512	
24	Dropout D_6	512	
25	Fully Connected FC_3	256	
26	Dropout D_7	256	
27	Fully Connected FC_4	128	
28	Dropout D_8	128	
29	Softmax	43	

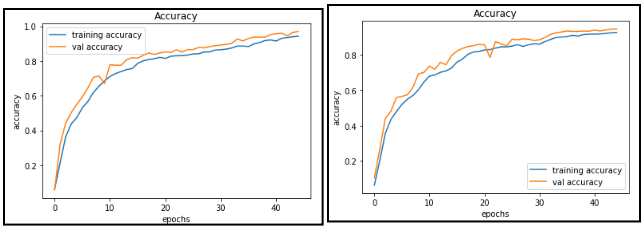
V. IMPLEMENTATION

The preprocessed data is loaded from a pickle file. The augmentation dataset is saved into two files both which contain the similar dataset but one of which has been normalized. The dataset is then loaded into the neural network for training purposes. In the next step, we test the model architecture with different hyper parameters.

We tried different training and testing parameters for our model, fig. 5 and fig. 6 shows the training and validation accuracy curve for the different number of epochs.

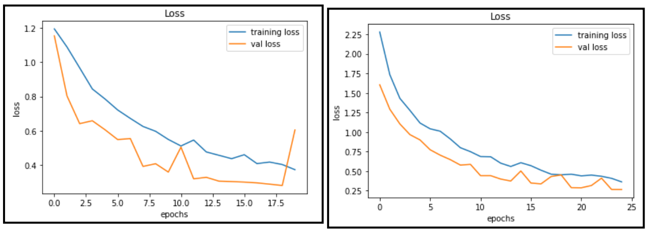


(a) 20 epochs (b) 25 epochs

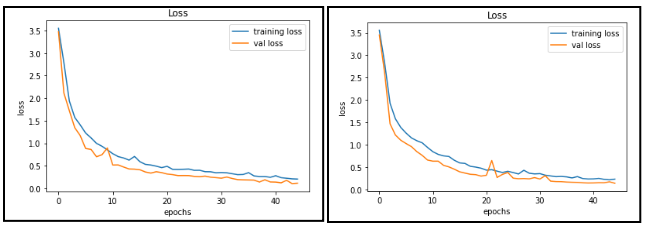


(c) 45 epochs (d) 45 epochs – normalized

Fig. 5 Accuracy curve over different number of epochs



(a) 20 epochs (b) 25 epochs



(c) 45 epochs (d) 45 epochs – normalized

Fig. 6 Loss curve over different number of epochs

All the models have been trained under similar condition with Dropout rate being 0.25 and learning rate being 0.001. Table II shows comparison of our proposed models. The final accuracy of the model is seen to be 96.9% where the Test Loss is 11.4%.

TABLE I - Comparison of Models

Model	Epochs	Training Time (secs)	Accuracy	Loss	Normalize Dataset used
1	20	2953	81.3	60.5	X
2	25	4497	88.7	26.6	X
3	45	9325	94.7	13.4	✓
34	45	5901	96.9	11.4	X

VI. RESULTS

In this paper, we tried to propose a system for the detection and recognition of traffic signs. We adopted an optimized model of a CNN architecture where we are train the model with augmented dataset where each image is augmented 24 more times, using different augmentation functionalities. The learning of our classifier was done using the customized German dataset. The CNN ensures accuracy in the achieved output. The Adam method incorporates all the aspects of CNN. The work includes processing RGB color images, in which RGB images gives more accuracy. This algorithm has a best speculation, and it can be trusted that it is used to identify more conventional traffic signs.

Thus, performance of a particular model totally depends on the problem and data at hand and it is not necessary for deeper neural networks to always surpass the rest.

VII. FUTURE SCOPE

The proposed architecture has successfully achieved the desired accuracy while reducing the probability of overfitting and internal covariant shift.

Future work will focus on extending the classifier to achieve good classification performance for traffic signs of different regions. This requires improvement in classifier design and its training methodology taking multiple datasets into consideration.

REFERENCES

[1] Shalinia G "*Traffic Sign Classification and Detection using Deep Learning*", International Research Journal of Engineering and Technology (IRJET), Volume: 06 Issue: 05, 2019.

[2] Alexander Shustanova and Pavel Yakimova, "*CNN Design for Real-Time Traffic Sign Recognition*", 3rd International Conference "Information Technology and Nanotechnology", ITNT-2017, 25-27 April 2017, Samara, Russia.

[3] Sepehr Foroughi Shafiei, Hafiza Rimsha Rauf, Yashdeep Singh, "*Traffic Sign Detection Using Deep Learning*", 2020.

[4] Sudha Natarajan, Abhishek Annamraju, Chaitree Sham Baradkar, "*Traffic Sign Recognition Using Weighted Multi Convolutional Neural Network*", Research Article, ISSN 1751-956X, 2018.

[5] Amal Bouti, Mahraz, Jamal Riffi, H. Tairi, "*A Robust System for Road Sign Detection and Classification Using LeNet Architecture Based On Convolutional Neural Network*", 2020.

[6] Yihui Wu, Yulong Liu, Jianmin Li,Huaping Liu, Xiaolin Hu , "*Traffic Sign Detection based on Convolutional Neural Networks*", 2018.

[7] Rongqiang Qian, Yong Yue, Frans Coenen and Bailing Zhang, "*Traffic Sign Recognition with Convolutional Neural Network Based on Max Pooling Positions*", 2017.

[8] *Pierre Sermanet and Yann LeCun, "Traffic Sign Recognition with MultiScale Convolutional Networks",* 2017.