JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1764 - 1769
https://publishoa.com
ISSN: 1309-3452

Classification of Traffic Signs using Convolutional Neural Networks

Dr. Shubhangi Vaikole[1], Makarand Bhalerao[2], Parth Nimbalkar[3], Soham Moghe[4]

[1]Computer Engineering Department, Datta Meghe College of Engineering,
Airoli, India,
shubhangi.vaikole@dmce.ac.in
[2]Computer Engineering Department, Datta Meghe College of Engineering,
Airoli, India,
nimbalkarparth12542@gmail.com
[3]Computer Engineering Department, Datta Meghe College of Engineering,
Thane, India,
soham.moghe11@gmail.com
[4]Computer Engineering Department, Datta Meghe College of Engineering,
Airoli, India,
makarand747@gmail.com

Abstract

Advancements in the field of Artificial Intelligence have optimized nearly everything in almost every industry. Automated and precise reliable models have been efficient in either assisting the humans, reducing the error of margin or completely taking over the tasks, speeding up processes and making them more efficient. Self-driving vehicles have been one of them. The following paper presents an approach towards a Traffic Sign Classifier with the help of Convolutional Neural Networks and has been tested on the standard dataset named as German Traffic Sign Recognition Benchmark (GTSRB) consisting of a total of 51839 images. Adam optimizer has been used to decrease the overall loss and increase the accuracy of the neural network. Dropout regression has been employed to prevent overfitting.

Keywords: Traffic sign classification, deep learning, image processing, convolutional neural network.

I. INTRODUCTION

Driving a vehicle is a really common capability today with nearly every person above the age of 18 being able to or actually using a vehicle to perform jobs or other chores. The automobile industry has seen a steep rise both in terms of vehicle demands and the things they are expected to do. Entire departments in the community have been dedicated to managing the consequent traffic that these vehicles create. However, although driving a vehicle seems like a fairly easy task, it is rather an intricate one, with mistakes often being fatal.That being said, advancements and automation have proven to be dependable solutions in making nearly every field, from healthcare to manufacturing both safe and efficient.

Automated and independently operable systems have become a common norm in our daily lives due to the advancements in Artificial Intelligence. One of the pioneering developments has been self-driving cars. Entirely independent self-driving cars or AI-assisted driving guidance systems are being proposed as a safe and efficient future alternative in the field of automobiles. During driving, recognition of road signs is crucial and fundamental to the very concept of autonomous driverless cars. Driving on streets, especially in residential or urban areas is a complex task that involves a variety of instructions and rules to be followed while driving. Road traffic signs act as indicators of what is expected of the driver as per these rules.Hence, recognizing them and building a precise system for traffic sign recognition is a matter of contention these days.

It has been observed that CNN i.e. the Convolutional Neural Networks has been extremely efficient when it comes to classification and recognition of traffic signs.With appropriate measures such as regularization to prevent overfitting and hyper parameter tuning, CNN models can reliably and precisely classify traffic signs. The presented paper explores an approach towards classifying traffic signs using CNN and is tested on the standard dataset named as German Traffic Sign Recognition Benchmark (GTSRB) dataset [6].

II. RELATED WORK

With the advent of unmanned automated vehicles able to drive a vehicle without a human driver, recognizing and classifying street traffic signs aptly and accurately have become a task of prime importance nowadays. There has been a lot of observed investigations for signs of traffic classification and recognition since the release of many big standard traffic sign datasets, like the BTSRB [5] and GTSRB [6] datasets. After a thorough analysis the approaches towards providing a solution to this problem statement can be broadly classified into two types namely the conventional methods

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1764 - 1769
https://publishoa.com
ISSN: 1309-3452

and deep learning methods. Since traffic signs have a definite and standard geometrical design a number of shapes based and color-based methods are used to classify them such as converting color segmentation and using Hough transform to detect shapes and finding the region of interest as demonstrated by Kedkarn et al. 2015[14]. Convolutional Neural Networks and Support Vector Machines and are the primary choices for traffic sign classification as demonstrated by Xue Yuan et al [1] and Rongqiang Qian et al [2]. M. Teague suggested a color-based segmentation method that included feature extraction using Histogram Oriented Gradients (HOG) and classification using SVM. Yi Yang et al [3] presented a integration of CNN and SVM to classify traffic signs. As traditional SVM-based methods involve manual extraction of important features and are thus computationally expensive, an inclination towards CNN has been observed. Zhu Z et al [4] created a fresh dataset of 100,000 images and proposed a CNN-based classifier on the same dataset achieving an accuracy of 84%. Recent advances in deep learning, specifically deep Convolutional Neural Networks (CNN), have yielded a slew of impressive outcomes on recognition benchmarks and object detection.

III. Proposed Work

In this study, a method based on the Convolutional Neural Network (CNN) is presented which is able to do classification of the various traffic signs in the German Traffic Sign Recognition Benchmark (GTSRB) dataset [6]. A Convolutional Neural Network can take an image as input and designate value (biases and weights) to different factors in the image, enabling them to be distinguished from one another. It comprises of several stages, each with numerous layers, namely the Conv2D layer, Dense layer, Maxpooling2D layer. Making use of these layers, we shall be able to classify a given traffic sign into one of the 43 classes present in the dataset. Entire process is explained in detail in the methodology section.

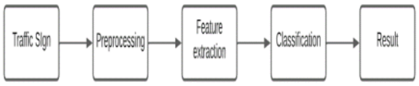


Fig. 1. Proposed model

IV. Methodology

The model has been trained and tested on the GTSRB dataset [6]. The images in the dataset are divided into 43 classes, each consisting of images from the same category. A total of 51839 images are present in the dataset of which 39209 are labeled images have been used for training and 12630 for testing.

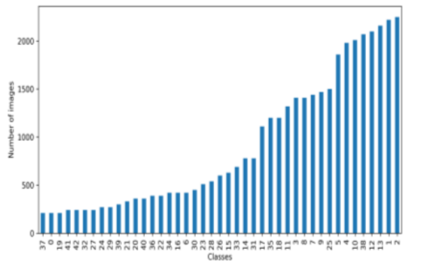


Fig. 2. Number of images per class present in the dataset

Fig. 3. Different types of traffic signs present in the dataset [15]

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1764 - 1769
https://publishoa.com
ISSN: 1309-3452

The images present in the dataset were of varying dimensions and hence preprocessing was needed before the images could be fed to the CNN model. All the images were resized to be 70*70 pixels. For the performance evaluation of the CNN model, the labeled images were randomly shuffled and split into a training set consisting of 80% (31367 images) of the images and a testing set consisting of 20% (7842 images) of the images.

The neurons present in the convolution neural network are organized in 3 dimensions: height, depth and width,, with depth referring to total number of filters. The first layer of our model is a Conv2D layer. This layer is a crucial component of the convolution process. It uses convolution to identify different features in a given image.

It works by scanning the full pixel grid and performing a dot product. Following the Conv2D layer is the Maxpooling2D layer. Max pooling represent a pooling method which identifies the maximum components from the feature map area occupied by the filter. The features are down sampled using this layer. It decreases the dimensionality of a huge image whilst preserving key elements.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 66, 66, 32)        2432

 conv2d_1 (Conv2D)           (None, 62, 62, 32)        25632

 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
 )

 dropout (Dropout)           (None, 31, 31, 32)        0

 conv2d_2 (Conv2D)           (None, 29, 29, 64)        18496

 conv2d_3 (Conv2D)           (None, 27, 27, 64)        36928

 max_pooling2d_1 (MaxPooling  (None, 13, 13, 64)       0
 2D)

 dropout_1 (Dropout)         (None, 13, 13, 64)        0

 flatten (Flatten)           (None, 10816)             0

 dense (Dense)               (None, 256)               2769152

 dropout_2 (Dropout)         (None, 256)               0

 dense_1 (Dense)             (None, 43)                11051

=================================================================
Total params: 2,863,691
Trainable params: 2,863,691
Non-trainable params: 0
```

Fig. 4. Architecture of the CNN model used

It aids in the reduction of computing time and weights. In order to ensure that the model does not overfit the data, dropout layers are also used in the model. It is a method to prevent overfitting in a model where certain neurons (selected randomly) are ignored or “dropped out”. At every updation of the training phase, the Dropout layer sets the outgoing edges of hidden units (This are the neurons present in the hidden layers) to 0. The output of pooling layers used in the model is in form of a 3D feature map, which needs to be converted to a 1D feature map before transferring data to the final fully connected layer. The flatten layer aids in this task.

The final layer used in our CNN model is a fully connected layer. This particular layer is where the actual classification takes place. It classifies the ultimate result of convolution or polling layer by layer, flattened layer by flattened layer. Weights are used to connect each input to each output. It integrates the features into additional attributes, resulting in a more accurate prediction of the classes. To calculate the loss, categorical cross-entropy method has been used since it is a reliable method useful in multi-class classification. To decrease the overall loss and increase the accuracy of the neural network Adam optimizer was used.

V. RESULTS AND DISCUSSIONS

The first experiment we did to get optimum accuracy was to determine the number of epochs for which our model was trained. The results of accuracy obtained when using a different number of epochs are given in the table below.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1764 - 1769
https://publishoa.com
ISSN: 1309-3452

TABLE I. ACCURACY OF THE METHOD

Number of epochs	Accuracy of the model
3	82.99%
6	92.56%
10	94.25%

As seen in the table above since the accuracy is the greatest after training the model for 10 epochs. The batch size is kept as 64 while training the model. Accuracy metric was used to obtain the precision of the output. As seen below accuracy increases with each epoch and loss decreases.

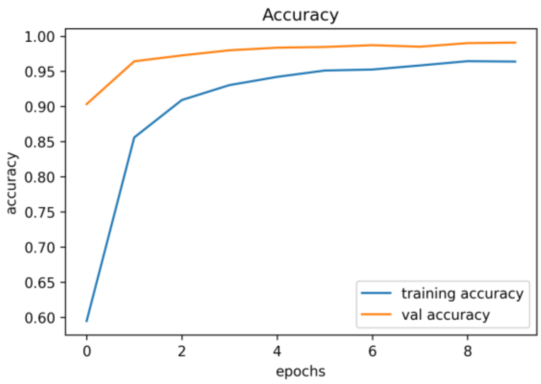
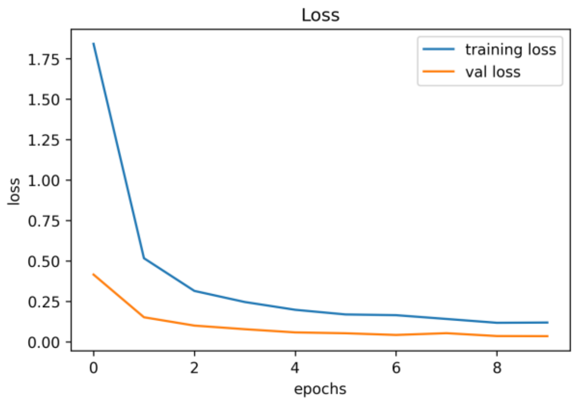


Fig. 5(a). Accuracy of model per epoch



We also tried to use different optimizers while training the model. However, the use of Adam optimizer resulted in the highest accuracy.

```
==============================
Total params: 242,251
Trainable params: 242,251
Non-trainable params: 0
______________________________
```

Fig. 6. Total number of parameters in the model

The model consisted of 242,251 parameters. After tuning various hyper parameters it was observed that the model an optimum accuracy of 94.25% after training for 10 epochs and loss stood at 19.97%

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1764 - 1769
https://publishoa.com
ISSN: 1309-3452

Fig. 7. Demonstrates how presented CNN method is able to classify signs of traffic into different categories.

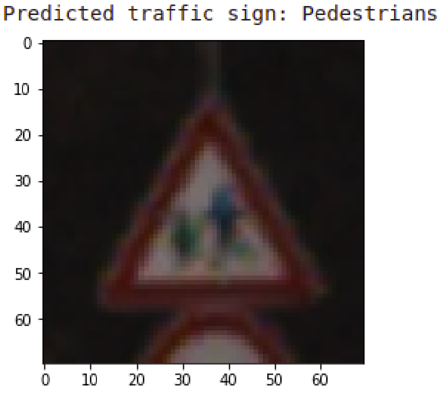


Fig. 7. Prediction on random image

VI. Conclusion

An approach to classify signs of traffic based on the Convolutional Neural Network (CNN) is proposed in this research. The design was evaluated using the publicly accessible GTSRB dataset, and the results showed that it was more accurate.

Furthermore, it employs dropout to combat overfitting by dropping part of the units from the neural network at random, and it is also thought to be the most efficient method of model averaging. Softmax is used as an activation at an output layer in the suggested network design because it calculates the likelihood of every potential class. The network Adam optimizer is used for training, and it is noted that it adapts faster than other optimizers such as SGD.

References

[1] X. Yuan, X. Hao, H. Chen and X. Wei, "Robust Traffic Sign Recognition Based on Color Global and Local Oriented Edge Magnitude Patterns," in IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 4, pp. 1466-1477, Aug. 2014, doi: 10.1109/TITS.2014.2298912.

[2] Qian, Rongqiang & Zhang, Bailing & Yue, Yong & Wang, Zhao & Coenen, Frans. (2015). Robust chinese traffic sign detection and recognition with deep convolutional neural network. 791-796. 10.1109/ICNC.2015.7378092.

[3] Yang, Yi & Luo, Hengliang & Xu, Huarong & Wu, Fuchao. (2015). Towards Real-Time Traffic Sign Detection and Classification. IEEE Transactions on Intelligent Transportation Systems. 17. 1-10. 10.1109/TITS.2015.2482461.

[4] Zhu, Zhe & Liang, Dun & Zhang, Song-Hai & Huang, Xiaolei & Li, Baoli & Hu, Shimin. (2016). Traffic-Sign Detection and Classification in the Wild. 2110-2118. 10.1109/CVPR.2016.232.

[5] https://btsd.ethz.ch/shareddata/

[6] https://benchmark.ini.rub.de/

[7] L. Chen, G. Zhao, J. Zhou and L. Kuang, "Real-Time Traffic Sign Classification Using Combined Convolutional Neural Networks," 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR), 2017, pp. 399-404, doi: 10.1109/ACPR.2017.12.

[8] V. Ciuntu and H. Ferdowsi, "Real-Time Traffic Sign Detection and Classification Using Machine Learning and Optical Character Recognition," 2020 IEEE International Conference on Electro Information Technology (EIT), 2020, pp. 480-486, doi: 10.1109/EIT48999.2020.9208309.

[9] Y. Sun, P. Ge and D. Liu, "Traffic Sign Detection and Recognition Based on Convolutional Neural Network," 2019 Chinese Automation Congress (CAC), 2019, pp. 2851-2854, doi: 10.1109/CAC48633.2019.8997240.

[10] L. Shangzheng, "A Traffic Sign Image Recognition and Classification Approach Based on Convolutional Neural Network," 2019 11th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), 2019, pp. 408-411, doi: 10.1109/ICMTMA.2019.00096.

[11] S. Mehta, C. Paunwala and B. Vaidya, "CNN based Traffic Sign Classification using Adam Optimizer," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 2019, pp. 1293-1298, doi: 10.1109/ICCS45141.2019.9065537

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1764 - 1769
https://publishoa.com
ISSN: 1309-3452

[12] R. Ravindran, M. J. Santora, M. Faied and M. Fanaei, "Traffic Sign Identification Using Deep Learning," 2019 International Conference on Computational Science and Computational Intelligence (CSCI), 2019, pp. 318-323, doi: 10.1109/CSCI49370.2019.00063.

[13] C. Timbuş, V. Miclea and C. Lemnaru, "Semantic segmentation-based traffic sign detection and recognition using deep learning techniques," 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP), 2018, pp. 325-331, doi: 10.1109/ICCP.2018.8516600.

[14] Chaiyakhan, Kedkarn & Hirunyawanakul, Anusara & Chanklan, Ratiporn & Kerdprasop, Kittisak & Kerdprasop, Nittaya. (2015). Traffic Sign Classification using Support Vector Machine and Image Segmentation. 52-58. 10.12792/iciae2015.013.

[15] Jingwei Cao 1,2 , Chuanxue Song 1,2, Silun Peng 1,2, Feng Xiao 1,2 and Shixin Song 3,*Improved Traffic Sign Detection and Recognition Algorithm for Intelligent Vehiclesdoi: 10.3390/s19184021