

Recognition of Traffic Sign Using CNN and Deep Learning

Pisati Mahipal Reddy¹, Arrabothu Vishal Reddy², Sowjanya Jindam³, Ubaidullah Mohammed Sayeed⁴,
Arrabothu Vivek Reddy⁵

^{1,2,4,5} Student, Department of Information Technology at Maturi Venkata Subba Rao Engineering College, Osmania University, Hyderabad, Telangana, India

³ Assistant Professor, Department of Information Technology at Maturi Venkata Subba Rao Engineering College, Osmania University, Hyderabad, Telangana, India

Abstract: An application of traffic sign recognition is proposed on the basis of the convolution neural network (CNN). A CNN is an artificial neural network that is used to process and recognize the image that focuses on processing pixel data. A dataset is trained, tested, and saved in order for the application to be able to detect and classify the image considered from the dataset. A Graphical User Interface (GUI) is designed for the user to try and use the application which will load the image from the dataset and classify the image as per its requirement. In the German traffic sign recognition criterion, an accuracy of 98% is obtained from the model used. Traffic Sign Recognition plays an integral part in the intelligent transportation system and has driverless vehicles and assisted driving systems are some of the applications of it [1]. The self-driving cars needed to identify each and every detail that are present on the road that includes vehicles on the road as well as pedestrians walking on the sidewalk with extreme accuracy and precision. There were no challenging and publicly available datasets in the domain for a period of time but the situations had changed in the year 2011 when Stallkamp et al [2] and Larsson and Felsberg [3] introduced datasets that includes demonstrations for traffic sign detection and classification of it.

Keywords: Convolution Neural Network, Pandas, Keras, Tkinter, SoftMax, Maxpool

I. INTRODUCTION

Autonomous vehicles are in the trend. The importance of the field needs to be addressed by the proper recognition and classification of traffic signs to achieve safe driving conditions to the driverless vehicles. It is an application which can reliably serve the purpose of classifying traffic signs. It is an area in which development and research has done but still it exists due to the various complications. For instance, the traffic sign cannot be detected properly in the dark environmental conditions. In the recent years Convolution Neural Networks is being used for the purpose of achieving high accuracy in detecting objects[4][5][6][7][8]. The model used is trained and tested to attain perfection as much as 97%.

II. ARCHITECTURE

The architecture of the work is as follows, the image of size 30x30 is used which then led to a convolution 1 is a 2D layer with size of (5,5,3) where 5,5 is the size and 3 represents rgb color space, then it is down sampled (maxpool) to reduce dimensions to increase performance where stride is 1 and 2 respectively. The combination of convolution and maxpool is a layer together which is followed by another convolution with size(3x3x3x64) and with similar maxpool size then it is flatten to a single vector then it results to a fully connected layer (FC3) then it is passed through a SoftMax activation since it is a multiclass classification (value will be same as number of output classes i.e,256)which gives the predict class which is used to calculate cost function using categorical_crossentropy since there are multiple classes that needs to be optimized. The model is trained using a back propagation optimizer (Adam) since it is a multiclass which performed great, the loss uses.

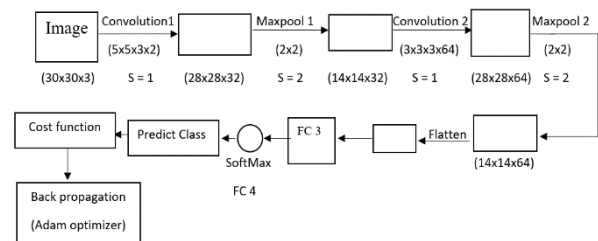


Fig. 1. Architecture of the model

III. METHODOLOGY

The methodology of the work is as follows, the images from the dataset are stored into labels which are then converted into NumPy arrays and eventually the data is split into the respective categories of training and testing which is later converted into one-hot encoding. Now, a CNN model is built and compiled which furtherly is trained, validated, and saved. The CNN generally consists of multiple layers [9] And then a GUI is built that can be accessed by the user to use and classify the traffic sign given.

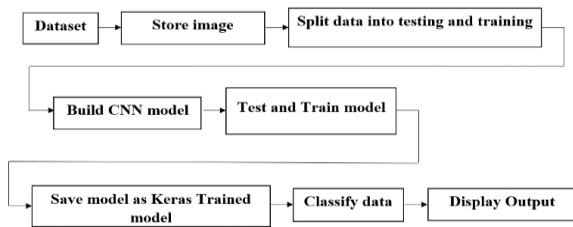


Fig. 2. Methodology for application

IV. ALGORITHM

Start

Open application

if (button clicked == upload image)

open select dialog

read image address

click open

if (button clicked == classify image)

read the model.h5 file to search for image

if (image selected found in tested model)

display the result

else

display error in compiler stating not fixed value

if (button clicked == close)

Stop

Once the algorithm is opened the given button is to be clicked in order for the application to get started and then the image address should be read to upload the image that is to be searched and classified if the selected image is found in the tested model the result is displayed or else a prompt will be displayed stating an error.

Testing and Training

After building the model architecture, model.fit() was used to train the model. The model worked great with the batch size of 32 as well as epoch size of 14 which showed fixed results and a precision of 97%. Representation of graph was done by the matplotlib library.

The classes of traffic signs that were used for classification are:

a) Mandatory



Fig. 3. Mandatory traffic signs

b) Prohibitory



Fig. 4. Prohibitory traffic signs

c) Dangerous



Fig. 5. Danger traffic signs

V. DATASET AND ITS USAGE APPROACH

A dataset (GTSRB - German Traffic Sign Recognition Benchmark) is considered by storing the images into labels and then lists are converted into NumPy arrays. Now, the data is split into testing and training respectively. And then labels test and train in one-hot encoding. A CNN model is built and compiled then trained and validated and then tested with the dataset and finally the dataset is saved into Keras trained model, which is loaded for the usage of application. A visual interface is designed for the front end of an application where a desired image is loaded, searched, and classified as per its requirement and the user gets a classified output. That is illustrated in further pages.

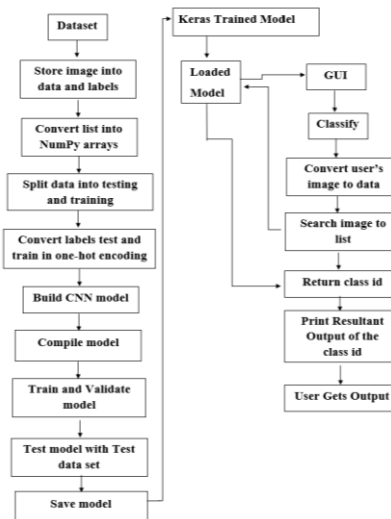


Fig. 6. Image showing flow of execution

VI. TECHNOLOGIES USED IN DEVELOPING THE APPLICATION

German Traffic Sign Recognition Benchmark(GTSRB) was used for the traffic sign dataset and various python modules are used in the implementation process. Those include Python image library, Pandas , Keras and Sklearn.

VII. DESIGNING OF GRAPHICAL USER INTERFACE(GUI)

A visual interface was built by the tk interface package in the python language for the classifier application. It mainly used the information from the trained model to extract features of the image that will be provided by the user at the front end using a GUI where it will get resized to 30x30 pixel size that gets compared to the trained model which has images in 41 classes and based on the result a forecast is made and based on it the user will be able to see the output using the same interface.



Fig. 9. Image showing output sampleoutput for Mandatory traffic sign



Fig. 10. Image showingsample output for Danger traffic sign

VIII. RESULT ANALYSIS

The graphs below represent the relationship between epochs, accuracy and epoch loss.

Graph with epoch size of 10

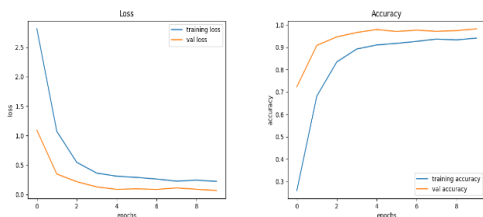


Fig.7. Graphs showing Loss,Accuracy with respect to Epoch(epoch size =10)

Table I: Values Recorded with EPOCH 10

Epoch	Time taken	Accuracy value	Loss	Accuracy value	Loss value
1	42	0.2596	2.813	0.7234	1.092
2	41	0.6816	1.071	0.9081	0.347
3	41	0.8340	0.545	0.9459	0.217
4	42	0.8918	0.363	0.9656	0.129
5	41	0.9097	0.310	0.9788	0.087
6	42	0.9171	0.289	0.9699	0.096
7	41	0.9259	0.262	0.9762	0.086
8	42	0.9362	0.227	0.9705	0.111
9	42	0.9326	0.243	0.9740	0.088
10	41	0.9405	0.222	0.9820	0.067

Test Score:0.06738291680812836

Test Accuracy:0.9820199012756348

Graph with epoch size of 15

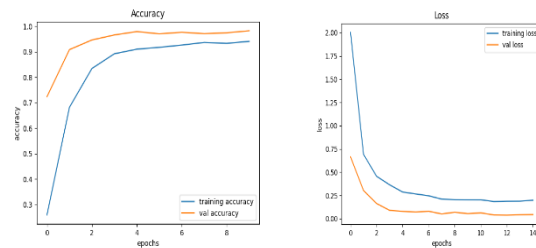


Fig. 8. Graphs showing Loss,Accuracywith respect to Epoch(epoch size =15)

Table II: Values Recorded with EPOCH 15

Epoch	Time taken	Accuracy Value	Loss	Accuracy value	Loss value
1	43	0.478	1.998	0.8082	0.687
2	42	0.800	0.691	0.9388	0.239
3	42	0.877	0.427	0.9642	0.142
4	41	0.907	0.318	0.9731	0.099
5	41	0.922	0.275	0.9719	0.099
6	41	0.923	0.270	0.9815	0.065
7	40	0.935	0.234	0.9748	0.087
8	41	0.938	0.226	0.9861	0.055
9	40	0.939	0.231	0.9776	0.077
10	40	0.943	0.218	0.9777	0.081
11	41	0.944	0.211	0.9898	0.041
12	40	0.947	0.206	0.9865	0.052
13	41	0.945	0.216	0.9907	0.034
14	40	0.950	0.201	0.9844	0.055
15	41	0.954	0.186	0.9890	0.045

Test Score:0.04512796550989151










Test Accuracy: 0.9890334010124207



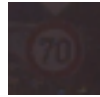
X. SAMPLE TESTING RESULTS

The output images of the program are depicted below representing the user interface which is seen by the end user.

The table below shows a sample test to check the functionality of the application.

Table III: Sampling Test Results using custom Inputs

S.NO	Given image	Expected output	Actual Output
1.		No Entry	No Entry
2.		Roundabout mandatory	Roundabout mandatory
3.		Speed limit (50 kmph)	Speed limit (50 kmph)
4.		Children Crossing	Children crossing
5.		Turn Left Ahead	Turn Left Ahead
6.		End of no passing	End of no passing
7.		Traffic signals	Traffic signals
8.		Ahead Only	Ahead Only
9.		Yield	Yield

10.		Bicycles crossing	Bicycles crossing
11.		End of speed limit (90 kmph)	End speed + passing limits
12.		Speed limit (70 kmph)	Speed limit (70 kmph)

Note images with S.No 11,12 were used to represent image conditions of dark and unclear image

XI. CONCLUSION

In conclusion, the project is used to classify the traffic signs which are not visible clearly to a human eye. It can be due to low light conditions, the distance between the observer and the road sign is far, and the quality of the road sign that was captured in an image is not of good quality.

XII. FUTURE SCOPE

The above application can also be developed furthermore that can include a voice-based assistance or it can also be programmed to drive a level 3 and above autonomous automobile driving which can lead to more safe driving conditions thus can cause less accidents and emergency vehicles could travel with traffic less roads

REFERENCES

- [1] Suisui Tang. (2014) Research on Traffic Sign Recognition Algorithm. Beijing Jiaotong University, PP. 1-7.
- [2] Krizhevsky A., Ilya S., Geoffrey E.H. ImageNet Classification with Deep Convolutional Neural Networks; Proceedings of the Advances in Neural Information Processing Systems 25; Lake Tahoe, NV, USA. 3-6 December 2012; pp. 1097-1105. [Google Scholar]
- [3] Lecun Y., Bottou L., Bengio Y., Haffner P. Gradient-based learning applied to document recognition. Proc. IEEE. 1998;86:2278-2324. doi: 10.1109/5.726791. [CrossRef] [Google Scholar]
- [4] P. Sermanet and Y. LeCun, "Traffic sign recognition with multiscale convolutional networks," in Neural Networks (IJCNN), The 2011 International Joint Conference on, 31 2011-aug. 5 2011, pp. 2809-2813.
- [5] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in Computer Vision, 2009 IEEE 12th International Conference on, 29 2009-oct. 2 2009, pp. 2146-2153.
- [6] D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber, "A committee of neural networks for traffic sign classification," in IJCNN'11, 2011, pp. 1918-1921.
- [7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in NIPS'12, 2012, pp. 1106-1114
- [8] Lai, Y., Wang, N., Yang, Y., & Lin, L. (2018). Traffic signs recognition and classification based on deep feature learning. In 7th International Conference on Pattern Recognition Applications and Methods (ICPRAM), Madeira, Portugal (pp. 622-629).

- [9] Hoo-Chang, S., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., ... & Summers, R. M. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5), 1285.
- [10] Dataset:GTSRB - German Traffic Sign Recognition Benchmark <https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign>