

Real time Traffic Signs Recognition using Deep Learning

Shambhavi Lalsinge¹, Professor. Tejaswini Bhosale²

Abstract - These days many object detection problems are being solved using deep learning or more precisely CNN(neural networks) due to its high recognition rate and fast execution. CNN has largely influenced all the computer visionary tasks. So in this project I propose a deep network traffic sign recognition/classification model with the help of python as the base language and followed by different python libraries for training the CNN model. This model will consist of different CNN layers which will precisely classify interclass samples from the dataset which will be provided. This system will be 99% efficient for recognizing the real time traffic sign and also tell from which class a particular sign belongs.

Key Words: Deep learning, CNN, real-time, python, training, model, OpenCV, tensorflow, keras, traffic signs.

1.INTRODUCTION

Lately autonomous vehicles have been a center of attraction for research study and development. And when we speak about autonomous vehicles Traffic Signs Recognition system is the first and foremost concept to include in any autonomous vehicles. Sometimes drivers may tend to miss the traffic signs along the route this can be dangerous and very unsafe with concern to road safety. In such cases automatic process of classification of traffic signs and reduce the no of road accidents on very large scale and can ensure complete safety. Many big names in automation used this system in their cars using computer vision and machine learning approach but this was soon replaced by deep learning approach based on classifiers. Recently deep convolutional techniques have been proved to be the most effective for object detection. It proves to be advantageous to look at the traffic signs recognition/classification with the deep learning the deep learning perspective. Classification of traffic signs is not a simple task it requires a huge dataset to go through various processes in deep learning. Initially the dataset gets divided into some ratio where those many images will go through one process and other no of images will go through other process and then will show the accuracy of the classes and then further it will get recognized. This concept has notable research work history and existing work going on still there are few areas/drawbacks that are yet to overcome.

2. LITERATURE SURVEY

There is a notable amount of research work done on traffic signs recognition/classification in history and still is going on. Different types of datasets have been used to solve different types of problems which includes detection, classification, tracking etc. After trying out and testing every object detection approach researchers got to the

implementation with deep networks. In recent years with increasing technology and availability of standard datasets deep learning method is being more preferable. First CNN architecture ever used for traffic signs recognition was LENET architecture.

In the paper 'Novel Deep Learning Model for Traffic Sign Detection Using Capsule Networks' by Amara Dinesh Kumar they had stated the various approaches been tried for traffic signs recognition in early stages. Also they have stated how CNN method is effective and preferable over other methods. They have also briefly stated about the capsule techniques.

In the paper 'Traffic Sign Classification Using Deep Inception Based Convolutional Networks' by Mrinal Haloi IIT Guwahati again they have explained the novel deep learning for traffic signs detection.

In paper 'A Novel Neural Network Model for Traffic Sign Detection and Recognition under Extreme Conditions' by Haifeng Wan they have mainly focused on the traffic signs detection by autonomous vehicles under extreme weather conditions.

In the paper 'Traffic Sign Detection for Intelligent Transportation Systems' by Ayoub Ellahyani¹ they have survey regarding TSR system being installed in transportation vehicles and this paper proposes that it is a step towards intelligent transportation system.

In the paper 'Traffic Sign Detection and Recognition using Image Processing' by 1Karthikeyan D they have proposed the image processing approach towards TSR.

In the paper 'Two-stage traffic sign detection and recognition based on SVM and convolutional neural networks' by Ahmed Hechri¹ In this study, a novel two-stage approach for real-time traffic sign detection .

3. METHODOLOGY

3.1 Algorithm and Flow of the Model

In this project we are completely going to code in Python and will be using Pycharm IDE. Starting with all the imports one by one we import first of all the required libraries. Then we create one directory for the data set we are going to use for this project. After that the code is divided into two .py files one is TrafficSigns2_Main.py and TrafficSigns2_Test.py. TrafficSigns2_Main.py consist of all the required parameters, libraries splitting of data set, CNN model, pre-processing and augmentation of the images. TrafficSigns2_Test.py consists of the continuous while loop in which the traffic signs will get classified.

In TrafficSigns2_Main.py starting with importing all the libraries. Then followed by specifying some parameters for processing the data set like defining the path for the dataset and csv labels file, setting the image dimensions, epochs, validation ratio etc. Then we import the images that will basically detect the no of classes and import all the images .After that we split the data into test and validation the ratio is previously defined in the parameters. And we store the images in all the x variables and all the labels in y variable. Then we check if the no of images matches the no of labels in the data set.

Then we read our csv file and then we are going to plot and visualize and see if we are collecting the correct data and classifying it properly before we start the training process. Moving on we are preprocessing our images converting it into the grayscale and then equalizing it for the sake of standardization. And store this preprocessed images in x variable. Then at the end of it we are going to create our convolutional model. There after we compile the model and immediately the training starts. Once the training is done it will show us our plot of the trained model. It will mainly plot the validation loss and accuracy. Then we test our testing dataset and export the score.

In TrafficSigns2_Test.py we also have imported some libraries. Then we simply set the parameters like threshold and framesizes. After that we set the camera parameters. We setup our camera and then we import the trained model file. Then we again preprocess the images as we did before. Then we have our class names so that it gets displayed and coming to the end we have created a while loop that will continuously run to give us our web camera images in which it will predict the image and the class name and the probability percentage of classification being accurate.

3.2 CNN Model Architecture

In the TrafficSigns2_Main.py we have also created the CNN model .We have used the LeNet architecture for the model here with few modifications. The model originally consists of total 7 layers. The layers consists of 3 convolutional layers, 2 subsampling layers and 2 fully connected layers. First layer is the input layer then there are 2 subsampling layers also known as maxpooling layer thereafter it has the fully connected layer and then lastly the output layer. So basically CNN has layer division as 3 types namely convolutional layers(Conv2D),pooling layers(Maxpooling2D) and fully connected layers.(fig.1)

LeNet Architecture

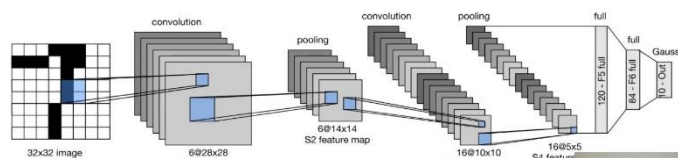


Fig.1 Standard LeNet Architecture

In fig.2 denotes the layers in a particular convolutional model. It takes the images as input then breaks down the images in small packets using the convolutional and pooling layers and then finally with the help of fully connected layers classification takes place and then we see the particular output.

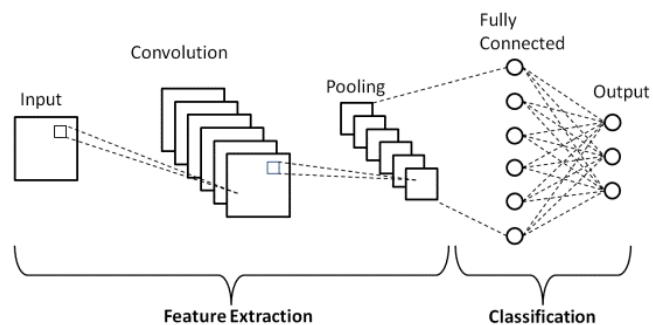


Fig.2 CNN model architecture

In this particular project we have used LeNet architecture as mentioned before. In this model we have few convolutional layers, few pooling layers then we have few dropout layers and at the end we have a dense layer which is nothing but our output layer.

4. RESULTS AND DISCUSSIONS

When we run the main.py where training,splitting and augmentation of the images is taking place.So firstly it will detect the no of classes data set has so this particular dataset has 43 classes detected and thereafter it will import each class folder one by on e and it will put all of these in one matrix with a corresponding class ID.After that it will split tha data into training,testing and validation so here certain no of images go for training certain for testing and some for validation as shown in the fig.3 it has around 22000 images for training,5000 images for validation and 6000 images for testing.

```

C:\Users\pratik>python main.py
Data Classes Detected: 43
Importing Classes...
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
Data Shapes:
Train(22072, 32, 32, 3) (22072,)
Valid(5000, 32, 32, 3) (5000,)
Test(6000, 32, 32, 3) (6000,)
Data shape: (48, 3, 32, 32, 3) (48, 3, 32, 32, 3)
    
```

Fig.3 Splitting of dataset

Once the splitting is done it will show the distribution of the training dataset in graphical form. Studying this training distribution it is important to know that we do not have same no of images for each class. So as we can see in fig.4 we have about 100 images for first class and similarly we have about 1300 images for another class it clearly implies that the distribution is not even so this may result in good classification for one class and bad classification for another class because the dataset is not evenly distributed for each class.

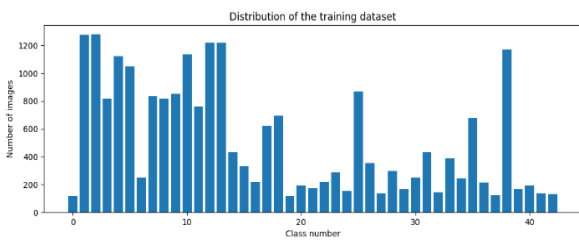


Fig.4 Training Distribution

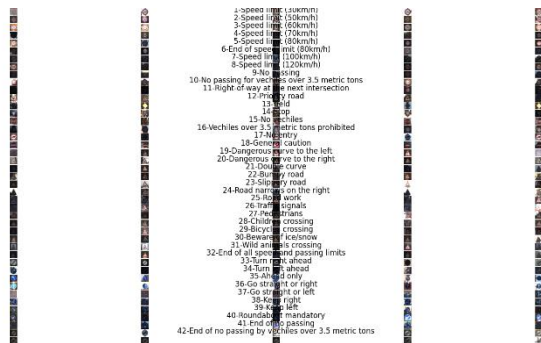


Fig.5 Each Class with Corresponding Label

Once this is done the pre-processing of the images will take place that is the images will get converted to grayscale and it will show us one example augmented image so that to check that the preprocessing is done properly.

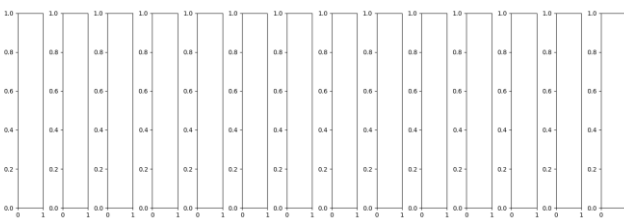


Fig.6 One Example Augmented Image

After compiling the model (fig.7) the training will start for the 10 epochs as parameter declared before and 2000 steps per epochs. So 2000 images will train for 10 iterations to give accuracy and loss of the model (fig.8).

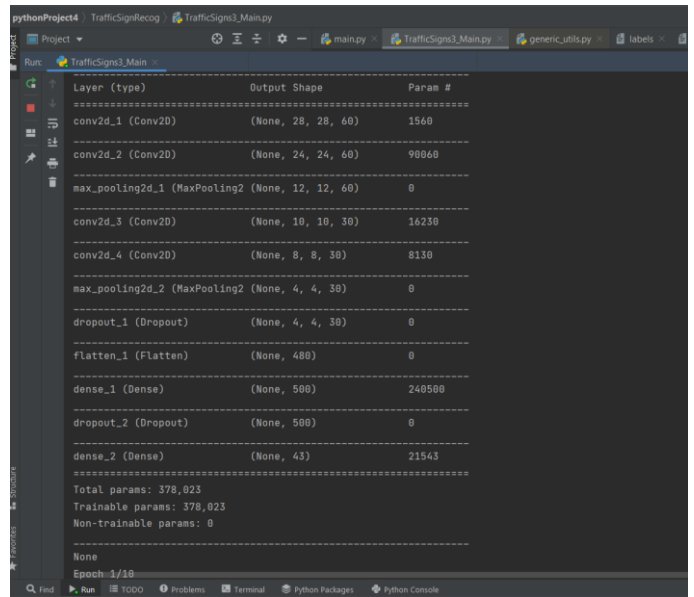


Fig.7 CNN Model

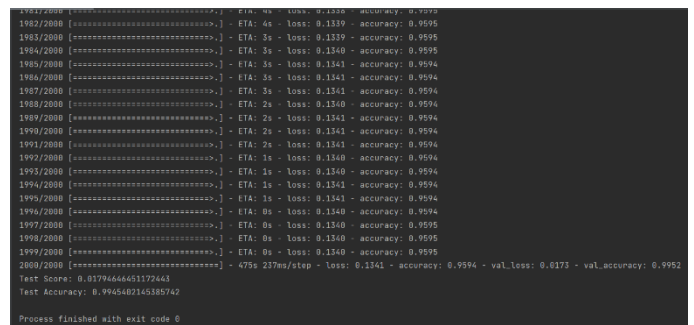


Fig. 8 Training per epoch

In fig.8 accuracy, loss, val accuracy is seen. So the accuracy it gets is around 0.95 which is fair enough to classify the images i.e it might get good results.

After that it plots the graph for accuracy and loss of the model it is getting fairly good results. For 10 epochs it is giving good results. After 4 epochs it is going at the same level. (fig.9) (fig.10)

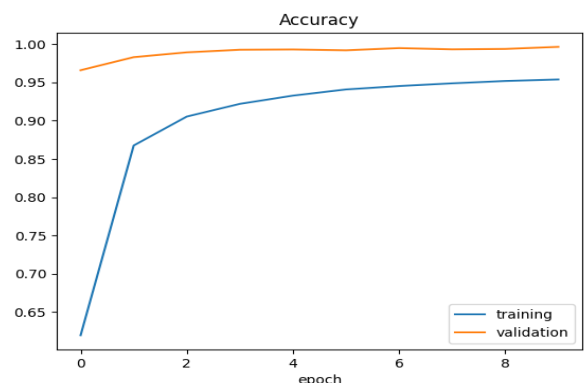


Fig.9 Accuracy Plot

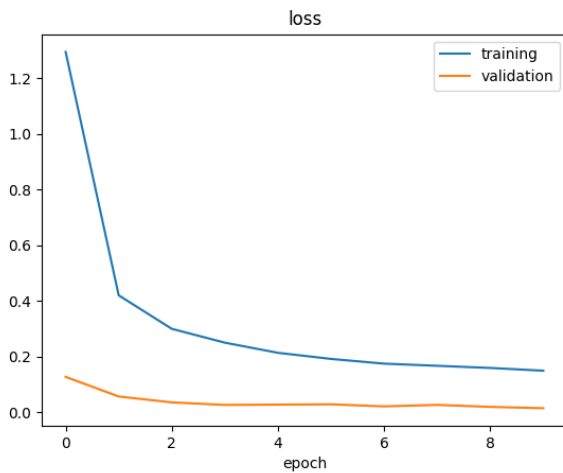


Fig.10 Loss Plot

Once training is done it shows us the real time classification of the traffic signs using OpenCV and will show the labels, class and probability of how much percent the classification is true of the particular traffic sign.(fig.11)



Fig.11 Real time classification

5. CONCLUSIONS

From this proposed system we conclude that after building a efficient CNN model and training the dataset well it acquires good classification results. And gives good accuracy results as well.

Originally to get good training distribution it requires the very big data sets but with the data sets that this system has used will also yield good results.

In some existing autonomous vehicles they have this system inbuilt but there could be many modifications and improvisations in the system to run this in every vehicle

smoothly. Installing this systems in vehicles can control, reduce upto 75% of accidents.

REFERENCES

- [1] Traffic Sign Classification Using Deep Inception Based Convolutional Networks Mrinal Haloi IIT Guwahati1.
- [2] A Novel Neural Network Model for Traffic Sign Detection and Recognition under Extreme Conditions Haifeng Wan , 1 Lei Gao , 2 Manman Su,1 Qinglong You,3 Hui Qu,1 and Qirun Sun 1.
- [3] Traffic Sign Detection for Intelligent Transportation Systems: A Survey Ayoub Ellahyani1 a , Ilyas El Jaafari1 and Said Charfi2.
- [4] Traffic Sign Detection and Recognition using Image Processing 1Karthikeyan D, 2Enitha C, 3Bharathi S, 4Durkadevi K 1 Assistant Professor, 2,3,4 UG Scholar.
- [5] Understanding Convolutional Neural Networks with A Mathematical Model C.-C. Jay Kuo.