TRAFFIC SIGN RECOGNITION SYSTEM IN SELF-DRIVING CARS USING DEEP LEARNING (CNN) AND MACHINE LEARNING (SVM & RF)

Student Name: Vijay Govardhan Pasula *Supervisor Name: Carolyn Devereux*

INTRODUCTION

- The goal of this project is to build models with high accuracy that detects and classify the traffic sign boards in self driven cars.
- For this purpose I have chosen 3 image classification models (CNN, SVM and Random forest) to compare the results with one another.

Fig 1: Source: https://images.wapcar.my/file1/7ea571f7ee1e488baa14273469fdbeb4_800.jpg

LITERATURE REVIEW

- Nazmul Hasan[1] has proposed that the traffic sign recognition system has two parts- localization and recognition. In localization part - traffic sign region is located and identified by creating a rectangular area. In recognition part the rectangular box provided the result for which traffic sign is located in that particular region.
- I found an interesting comparison by Annamraju, A.k[2] where they compared the accuracies for different sign categories like speed limits, prohibitions, danger signs etc. And their future work is focused on extending the performance of the classification for traffic sign in different geographical regions.

METHODOLOGIES

1. Convolutional Neural Network (CNN)

Data preprocessing:

I imported the data directly from Kaggle to the google colab. And I retrieved the images and it's labels. The data was split into 80:20 ratio for training and testing purposes. Then I checked the shape of the data and converted the labels to one-hot encoding.

Building the model:

I built a sequential model by adding a few convolutional layers, max pooling layers, dense layers, and a few drop outs.

- *Activation function = Relu*
- *Optimizer = Adam*
- *Metrics = Accuracy*
- *Loss = categorical cross entropy.*

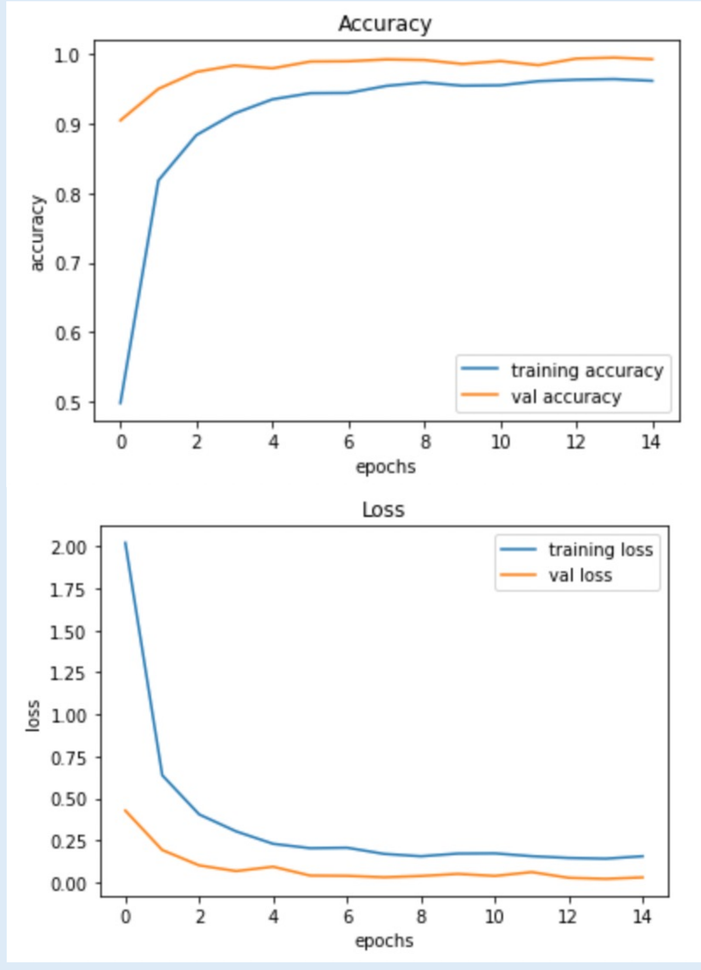
⟺ Compiled the CNN model with these settings

Training the model:

I trained the CNN model for 15 epochs and batch size of 64, which gives 93% training accuracy and 97% validation accuracy.

TRAINING SPEED: 32 minutes
TESTING SPEED: 20 seconds

RESULTS OF CNN MODEL



Training accuracy of CNN model is 96%

Validation accuracy is 99%

Training loss of CNN model is 15%

Validation loss is 3%

Fig 2: Plotting the accuracy and loss of CNN model for 15 epochs

2. Support Vector Machine (SVM)

In SVM, after data preprocessing (which is similar to CNN), I normalized the data and defined the SVM model and then I trained the model. Using SVM classifier, predicting the values for the test data.

Metrics used:

a. Recall
b. Precision
c. F1 score
d. Accuracy

Classification report of SVM

	precision	recall	f1-score	support
0	1.00	0.92	0.96	38
1	0.96	0.96	0.96	496
2	0.94	0.97	0.95	450
3	0.89	0.93	0.91	280
4	0.95	0.98	0.96	418
5	0.90	0.92	0.91	364

And also calculated the macro average and weighted average..!

accuracy			0.96	7842
macro avg	0.97	0.95	0.96	7842
weighted avg	0.96	0.96	0.96	7842

WHAT NEXT?

3. Random Forest

Am working on Random forest model and then finally I'll compare all the three models in terms of accuracy, training speed and test speed.

References:

1. Hasan, N., Anzum, T., Jahan, N., (2020) 4th Int. Conf. Inventive Comms. and Comp. Tech., 'Traffic Sign Recognition System (TSRS): SVM and Convolutional Neural Network' pg.5
2. Natarajan, S., Annamraju, A.k., Sham Baradkar, C., (2018) "Traffic sign recognition using weighted multi-convolutional neural network" pg.2.

Data Source:

https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign?datasetId=82373&searchQuery=svm2.