Classification of Images of Road Traffic Signs using CNN

Introduction

Image classification has been one of the major applications in Computer Vision space. It is used for identifying objects in a picture and classify them accordingly. This practice became increasingly popular due to advancement in technology in the recent years.
In this report, data is adapted from the modified version of the Belgium Traffic Sign Classification Benchmark and the goal is to classify the images based on the based sign and the shape-type.

Overview of the classification process

- Data Loading: Image paths and their respective labels(both shape and sign) are stored in dataframe.
- Data Exploration: The distribution of labels with respect to specific label type and shape type are observed. It shows an imbalanced distribution of labels for both the categories.[Fig. 1]
- Data preparation: The data set is divided into three partitions – train, validation and test sets.
- Data Modelling: Model adapts VGG16 architecture for this classification task as it has been proven to classify images of multi-class efficiently and appropriately[1]
- Hyper-Parameter tuning: Augmented data, dropout methods, class weights parameters are adjusted for this experiment.
- Individual evaluation: images are sourced from multiple sites such as Kaggle, Belgium Traffic sign benchmark website and google image search results.

Algorithm Approach

Image classification can be achieved using traditional machine learning algorithms. However, latest technology such as deep learning with the help of Convoluted neural networks are more suitable for image related problems as they are efficient in processing un-structured data compared to the latter[2].

In [1], they suggest VGG16 model with 8 layers as one of the candidate models among five suggested for traffic sign classification. The model used for this assignment is inspired from this architecture, implemented with slight modifications. The algorithms are tested with base line models initially, and the hyper parameters are tuned to achieve the best result for both the models (sign-type and shape-type). Below are the details of the hyper parameters used

- **Image Augmentation**: Images are rotated by max to 20 degrees, shifted width, height, zoomed and flipped horizontally. Each of these operations are applied to an image randomly by ImageDataGenerator by Keras.

- **Class weights**: This parameter has been set so as to weights are given importance during back propagation, proportionally as there is imbalanced data set.

Shape type classification

- Baseline model with 8 layers is trained without any changes made to the data[1].
- Although, 97% accuracy is obtained for validation data [Fig. 2] tell that the model is overfitting as train results show that training set has better performance than the test set.
- In order to overcome over-fitting, class weights are introduced[3] and also with the help of augmented, better results are obtained.
- Best model is with augmented data and class_weights set which provides generalized classifications.

Sign type classification.

- Baseline model with 8 layers is trained without any changes made to the data[1].
- 95% accuracy was obtained but there is clear evidence of over-fitting.
- Validation data has more accuracy than the train data with accuracy of 95%[Fig. 4].
- Test data provides results with 96% , which is unseen data.

Independent evaluation

- Challenges
    - Images were sourced from various sites and had different formats, different sizes
    - Some of the images are close representation of the required image, this behaves like a noise.
- The test accuracy for sign type was 65%
- The test accuracy for shape type was 79%

Ultimate Judgment

- Various Models have been tried with VGG16 as the base architecture.
- Data Augmentation is required in imbalanced dataset in order achieve consistent generalized results.
- Hyper parameter tuning is required in order to get the better generalized model and to prevent over-fitting scenario.
- However, further improvement can be brought into the model by tuning further using keras tuner library.

References

[1] https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8558481

[2] https://hackernoon.com/deep-learning-vs-machine-learning-a-simple-explanation-47405b3eef08

[3] https://machinelearningmastery.com/introduction-to-regularization-to-reduce-overfitting-and-improve-generalization-error/

Appendices

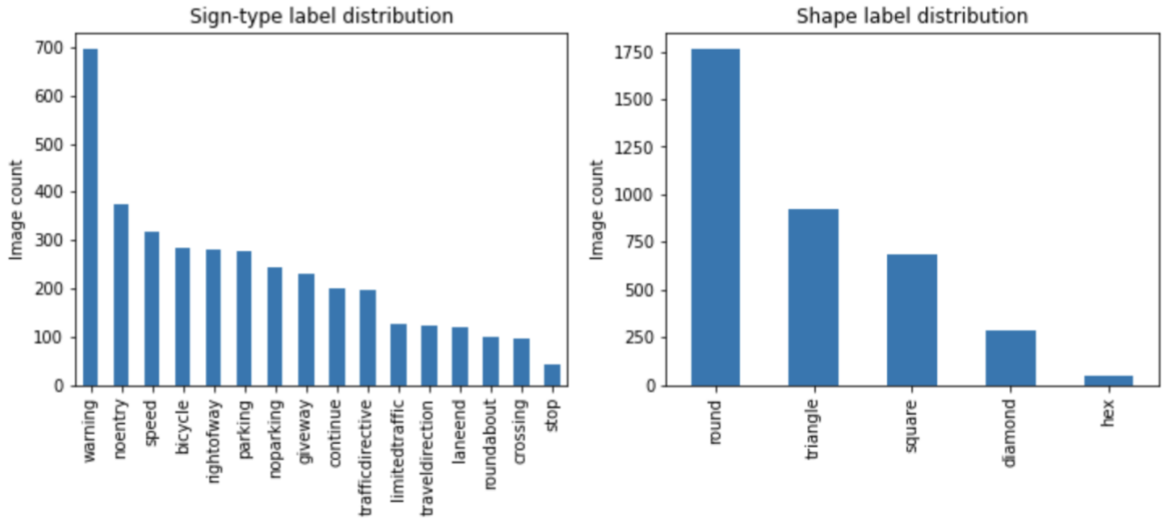

Fig. 1 : Distribution of images based on sign type(left); based on shape(right).

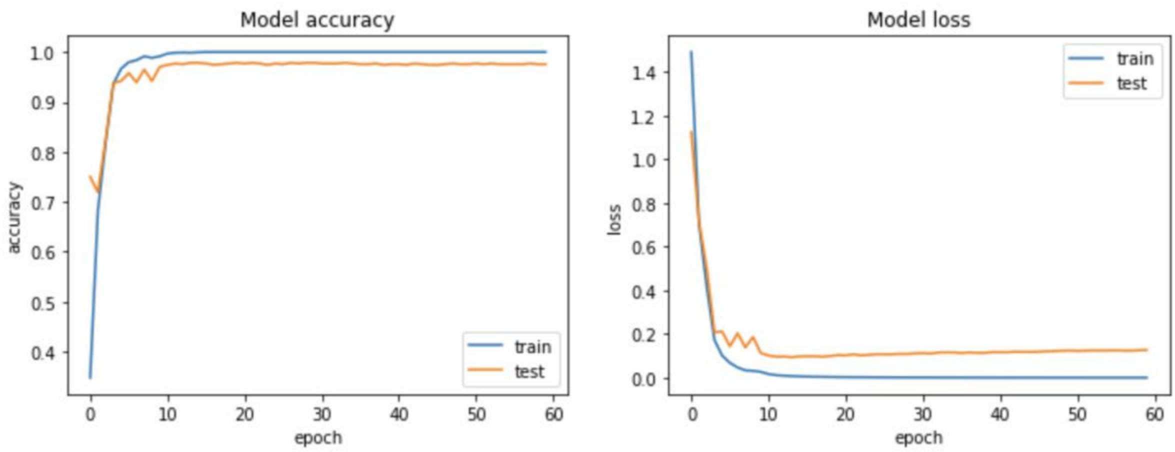

Fig. 2: Model accuracy plot(left) and Model loss plot(right) for shape-type classification.

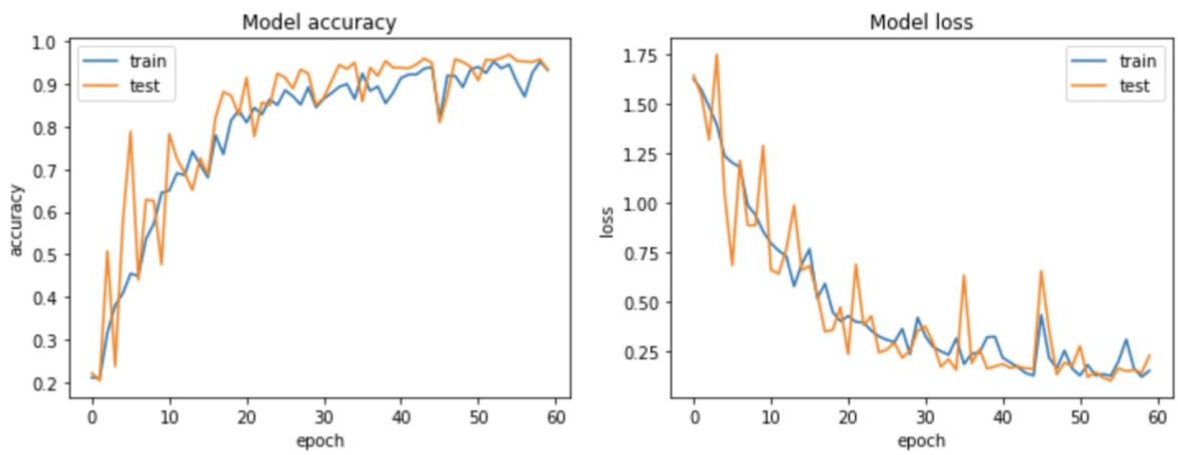

Fig. 3: Model accuracy plot(left) and Model loss plot(right) for shape-type classification with augmented data.

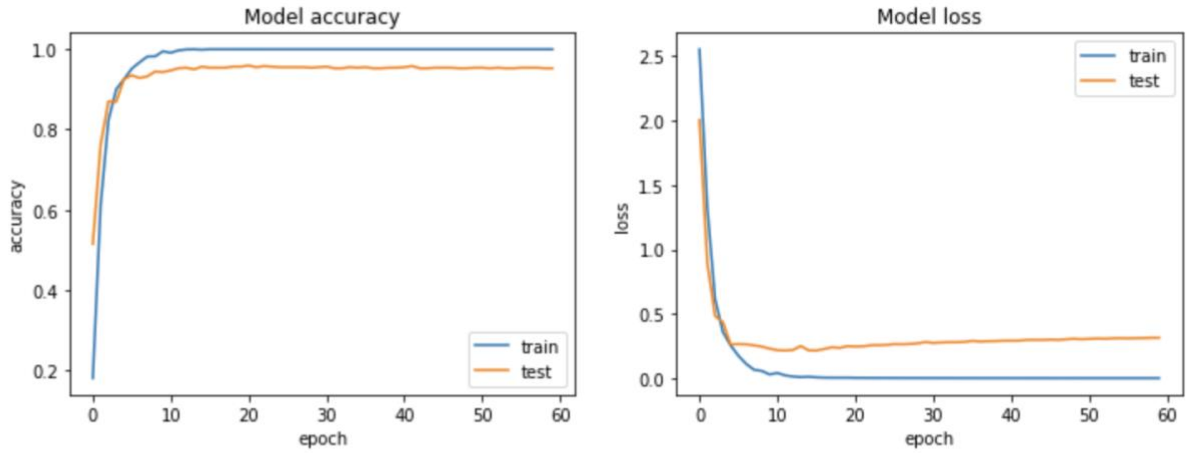

Fig. 4: Model accuracy plot(left) and Model loss plot(right) for sign-type classification.

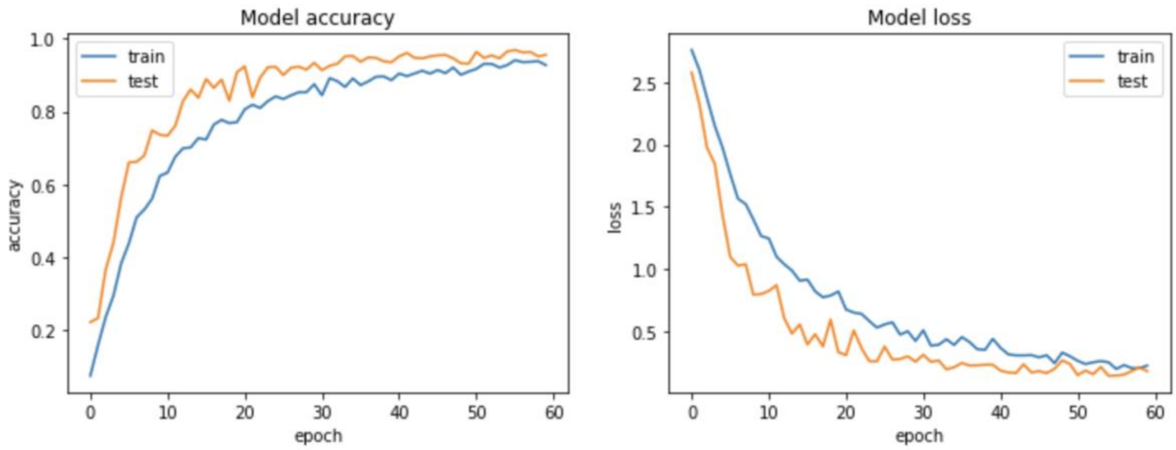

Fig. 5: Model accuracy plot(left) and Model loss plot(right) for sign-type classification with augmented data.