Machine Learning Home Work2 Report

Hafeez Jimoh: hjimoh@innopolis.university

October 23, 2019

Traffic Sign Recognition with Random Forest Classifier.

1 Exploratory Data Analysis and Preporocessing of Datasets

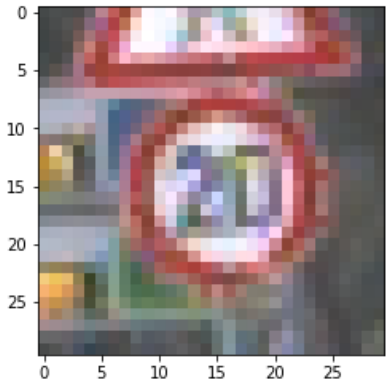

Figure 1: Image Sample

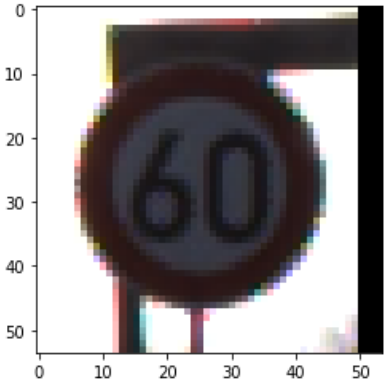

Figure 2: Image Sample

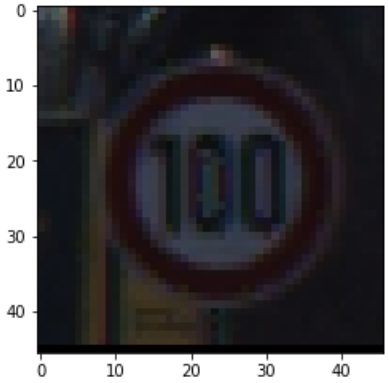

Figure 3: Image Sample

1.1 Padding and Resizing

The Images were observed to have different sizes of height and width. They were padded by adding Zeros to the shorter side to make it a square shaped. All images were equally resized to a shape of 30*30 after the padding operation was done. The result of the padding and resizing as compared to the original image shown in Fig 1-3 is shown in Fig.4 -6.

1.2 Frequency of Image Labels (Traffic Signs)

An additional step done as part of the Exploratory data analysis is to examine the frequency or distribution of our datasets. As shown , it was observed that it is a largely unbalanced dataset with different frequency of data/Image labels. A bar chart shown in Fig. 7 and 8 show the frequency distribution of the image labels.

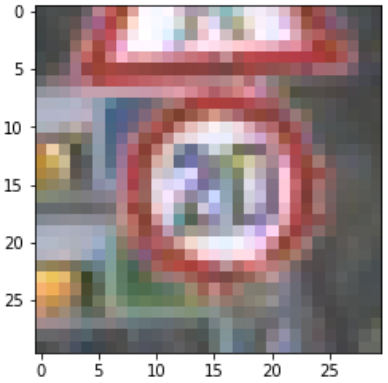

Figure 4: Resized Image

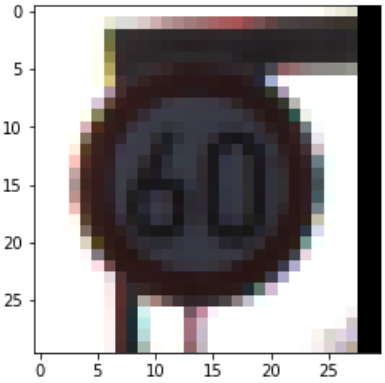

Figure 5: Resized Image

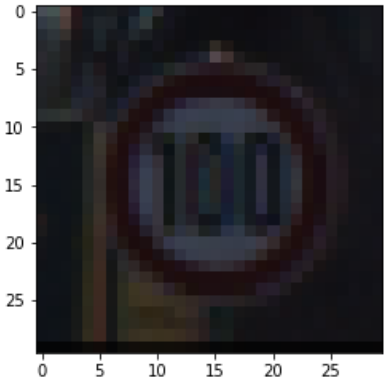

Figure 6: Resized Image

The training images from the dataset were splitted intoa new training set and validation sets. The distribution was fairly maintained so that the valiadation would have some labels to evaluate on. The result of distribution of the labels before and after split is shown in Figure 7 and 8.

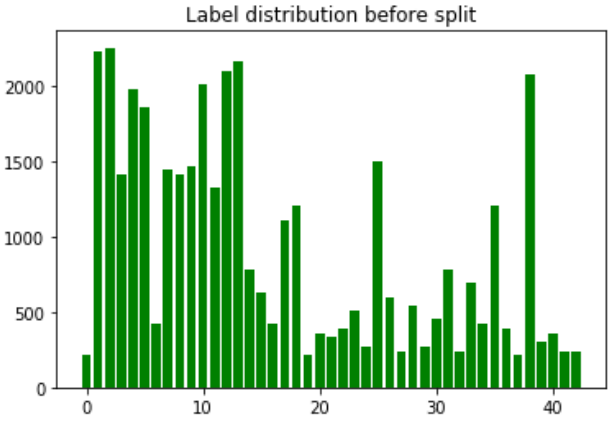

Figure 7:

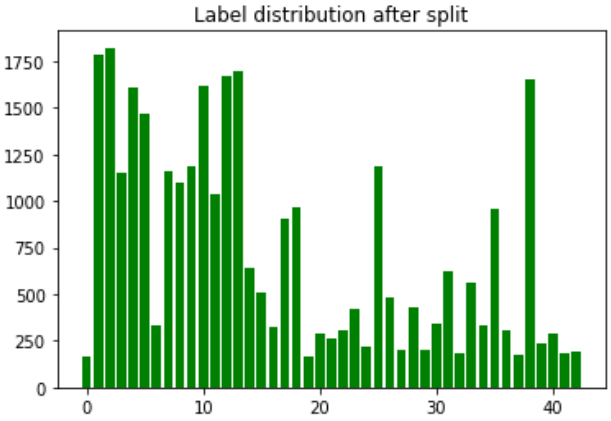

Figure 8:

1.3 Data Augmentation

Basic summary of dataset

Total Number of Training Examples = 39209
Number of training examples after validation split = 31368
Number of testing examples = 12630
Number of validation examples = 7841
Number of Training samples after augmentation : 78002
Class with highest frequency :2
Class with highest frequency :0, 37 and 19
Highest Frequency after validation split :1801

Highest Frequency before validation split :2250

There are various techniques to augment data. Augmentation in this sense means generating more training samples from the training samples available. This is done for a number of reasons. This is usually done when we have unbalanced classes and we do not want our classifier to be a majority classifier that predicts base on the majority class. We also do it when we have small dataset. Small dataset can result into overfitting. Also, training our model without augmenting when situation demands can result into overfitting. In this work, two data augmentation techniques have been applied. In other to make sure that individual augmented data are unique, some randomizations have been applied to the parameter off the function that augments the data. This prevents artificial dataset of being monotonous or correlated. The class distribution after the data augmentation process is shown in Figure 9 below.

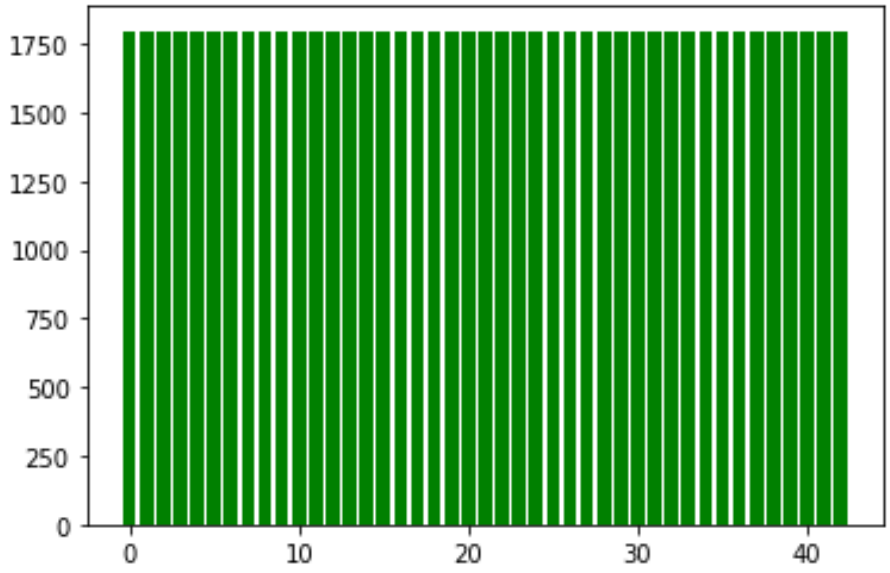

Figure 9: Frequency Distribution after Augmentation

1.4 Translation

Translation of an original image was done as a data augmentation technique. This is a simple technique that assumes for example, images are captured at different locations. While the same image would be gotten, the representation of the image on the cartesian plane would be different. To make our algorithm robust, it should be able to predict the same image captured at different positions. The images were simply shifted to the right by some random distance specified in the parameter of the function that does the augmentation process.

1.5 Rotation

Rotation was also done to the images as a data augmentation process. The images were rotated by some random degrees.
The result of the Data Augmentation of some image samples is shown in Figure 10-11.

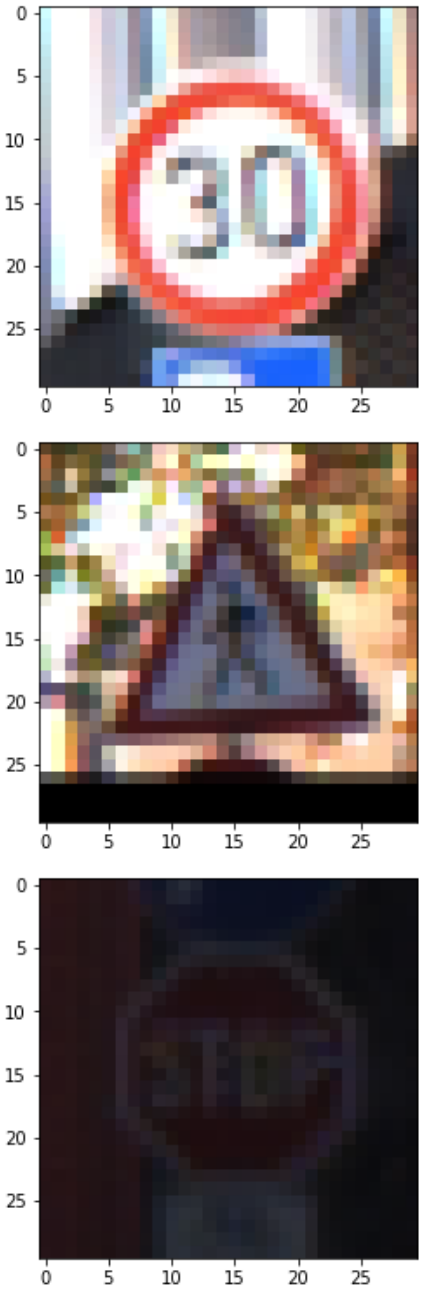

Figure 10: Original Images

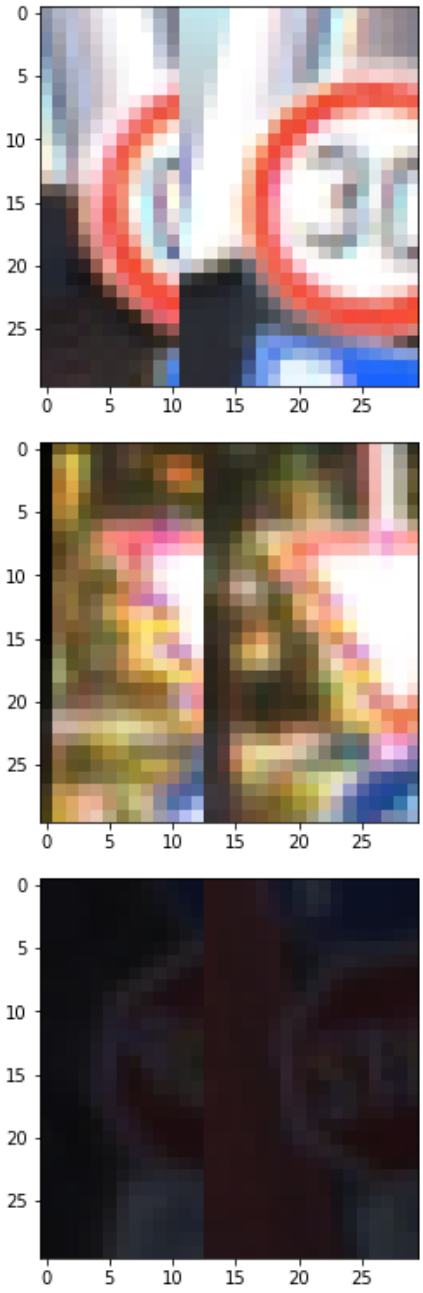

Figure 11: Augmented Images

2 Results

A random forest classifier was used to train an image of 78001*2700 using number of estimators as hyper-parameter. The model was then evaluated on the validation set and the result obtained is presented in Table 1. It can be observed that the performance of out model performs better as the number of estimators increased. Though this is at a huge cost of longer trainning time and computationally expensive if not training on a GPU machine. While it took less than less than 2 minutes to train when the number of estimators is 10, it took more than 30 minutes when the number of estimators is 300. The training was done on a Core i5 8GB ram CPU.

The estimator of 300 was choosen since it gave the best performance to predict labels for the unaugmented data of size 31368 *2700. The result obtained is shown in Table 2. It can be observed that Augmented data set performed slightly better than the unaugmented data set on validation set.

The model with estimator = 300 was finally used to evaluate the test dataset. The result obtained is presented in Table 3.

Cross Validation Result with Validation Set			
estimators(number)	Accuracy (%)	Precision (%)	Recall (%)
10	88.53	89.01	89.02
50	95.75	95.84	95.84
100	96.55	96.62	96.62
200	97.21	97.26	97.26
300	97.25	97.30	97.30

Table 1: Cross Validation Result with Validation Set(using Augmented Data to train model)

Comparison of Model Performance		
Evaluation	Augmented Data Model	Unaugmented Data Model
Accuracy (%)	97.25	86.54
Precision (%)	97.30	95.84
Recall (%)	97.30	96.62

Table 2: Evaluation result on Augmented and Unaugmented Data (estimators = 300)

Comparison of Model Performance on Test Data	
Accuracy (%)	71.28
Precision (%)	71.28
Recall (%)	72.74

Table 3: Evaluation result on Augmented and Unaugmented Data (estimators = 300)

2.1 Incorrect Prediction

A total of 3617 were incorrectly labelled which amounts to approximately 30% of the dataset. Observation shows majority of the incorrectly labelled test data are the ones with low frequency of occurrence in the test sample. For example, Figure 12-17 shows some samples of the dataset with wrong prediction. Figure 14 was incorrectly as class 11 while it is class 27. Meanwhile, class 27 only has a precision-recall area of 0.060. Incidentally, class 27 only has a frequency of occurrence of 3 in the test data which is not representative of the datasets. Thus, the resultant low metric performance.

2.2 ROC and Precision-Recall Curve

The Precision- Recall curve and ROC curve for each label is shown in figure 16 and 17. It is a probability plot that simply tells how much the model is capable of distinguishing between classes. It plots the False Positive Rate for each class against the True Positive Rate. The area under the curve is AUC (Area Under Curve). The greater the area, the higher the chance that the model would distinguish between positive class and negative class. In our case, a positive class is a class out of 43 classes and the negative class is the other 42 classes. It can be observed from the plot that classes between 19-30 have a lower AUC. Correspondingly, they also have lower precision-recall area. This classes on observation of the count of labels of test data have lower representation in the dataset.

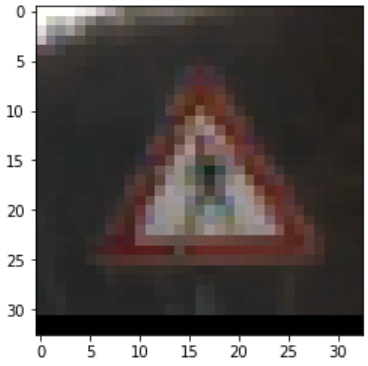

Figure 12: Incorrect Prediction

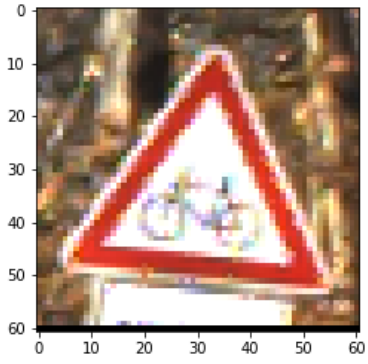

Figure 13: Incorrect Prediction

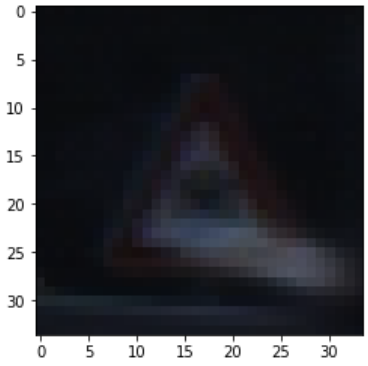

Figure 14: Incorrect Prediction

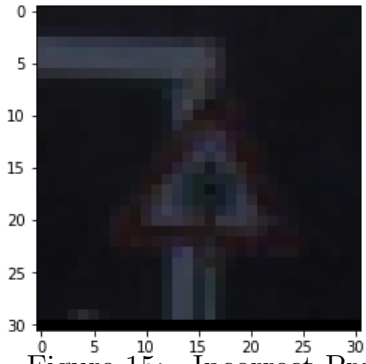

Figure 15: Incorrect Prediction

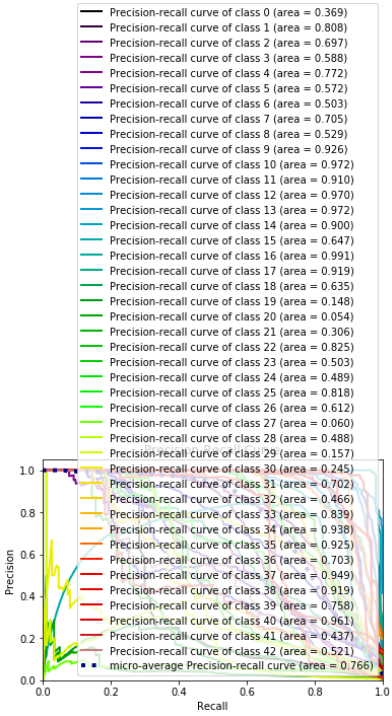

Figure 16: Precision-Recall Curve

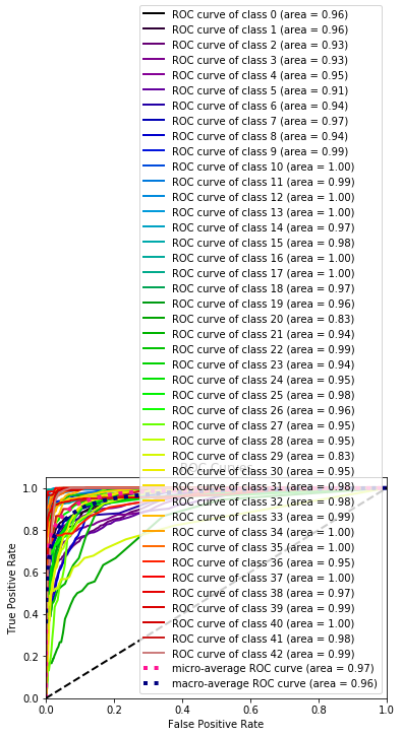

Figure 17: ROC Curve

2.3 Conclusion

The model was only tested using number of estimators as hyperparameters. While there are other parameters like criterion (gini or entropy), maximum depth, minumum number of samples and a host of others that could have been specified and apply grid search on them to know the best combination to tune our model with. This is going to be highly computationally expensive and would require longer training time even though there is a huge possibility that there would be significant improvement in our metric performance. Thus, a simple test case of tuning with estimators was employed.