Machine Learning Homework

Farit Galeev

October 2019

1 Introduction

For this homework we have been asked to imagine ourselves as machine learning engineers in self-driving car project and solve the most important problem of following traffic signs. So that for this assignment we have been asked to build classifier which can recognize 43 classes of traffic signs from GTSRB dataset.

2 Padding and Resizing images

Before we start further actions we need to make all images of the same size. Because dataset contains images from 15x15 to 250x250 sizes firstly, we need pad images to the square and after that make a resizing

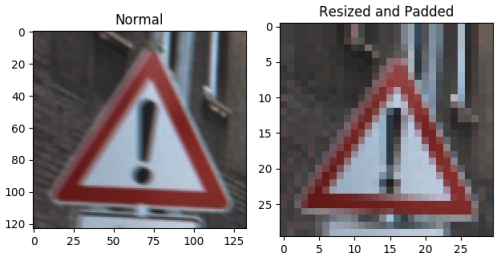

Figure 1: Padded and Resized image to 30x30

3 Data Splitting

We need divide dataset in 80-20% proportion where 80% is the training dataset and 20% is the validation dataset. The data is organized in tracks,so there are 30 photos (video frames) for each sign taken from different distances. So that,

we need to be sure that after splitting images from the particular track end up being in either training or validation split because if images from one track will be both in training split and validation split this may cause an overfitting.

4 Augmentation

After splitting we will get following distribution of classes [**Figure 2**]. So that we see that the dataset is imbalanced. This may cause a problem that the classification output is biased as the classifiers are more sensitive to detecting the majority class and less sensitive to the minority class. To eliminate this problem we will use **Oversampling** technique using data augmentation. For augmentation I used framework *albumentations* which is very powerful. What kind of augmentations I used you can see in **Figure 3**. Blur was chosen to imitate different quality of the image, rain was chosen because some of the photos may be made in rainy days, sun flare was chosen because as photo was taken in sunny day due to reflective surface traffic sign can produce a flare, as photo was taken at night flare can be produced by headlight and shift scale rotate was chosen to imitate distance between a car and traffic sign (shifting and scaling) and to imitate camera deflection (rotate).

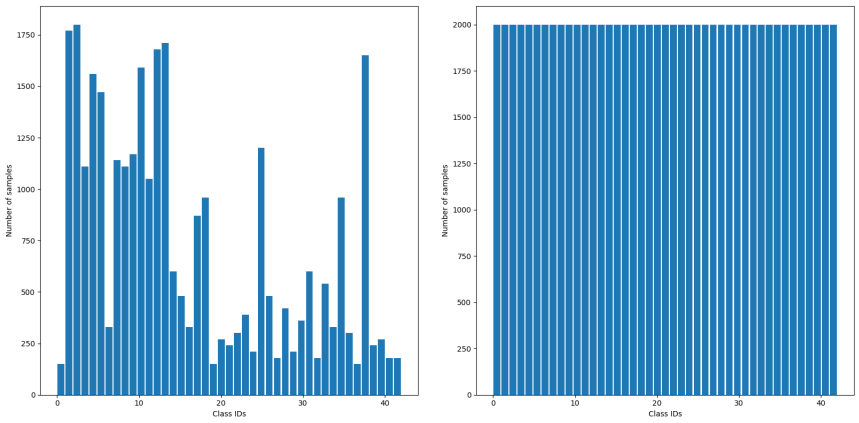

Figure 2: Histogram of 43 classes with their number of examples. Before and after augmentation

Figure 3: Different augmentations techniques

5 Testing

The overall accuracy for Random Forest Classifier applied on different image size stated in **Table 1**. So we can conclude that augmentation can give better results in accuracy (1-2%). As stated in the assignment description *overall accuracy is not very representative for all classes - some are underrepresented*, so that we need to measure precision and recall for all classes and we can see in **Figure 4** that distributions of precision and recall looks similar to distribution which is drawn in **Figure 2**. This can be explained that the model training and application process are not representative of the testing environment. The model is trained on balanced dataset and applied on an imbalanced dataset [**Figure 5**]. So that we see that precision and recall scores are mostly low for minority classes. Also in **Figure 6** we can see that accuracy depends on image size growth sublinearly and time for model training growth linearly (the -1 image size on graph means that model was training without augmentation on image size 30x30)

Examples of incorrect predicted classes you can figure out in **Figure 7**. As we can see Random Forest Classifier mostly learn shapes of the traffic signs but not the content. This may be cause because of the compression curse (pixels from high resolution image are overlapped and averaged), so that after compression image losing their contours of traffic sign content. This may visually recognized on traffic sign with speed limit 70 [**Figure 7**].

	30x30* Without augmentation	4x4	8x8	16x16	30x30	32x32
Validation score	0.6877	0.2895	0.5208	0.6519	0.7004	0.7080
Test score	0.6888	0.2907	0.5014	0.6475	0.6897	0.6944

Table 1: Comparison table of overall accuracy of Random Forest Classifier

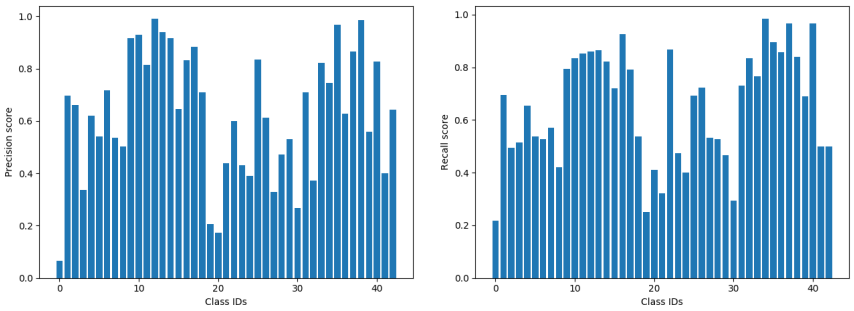

Figure 4: Precision and Recall distribution per class

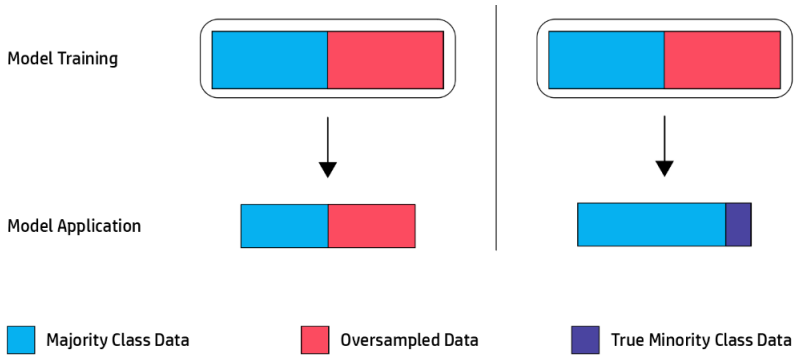

Figure 5: Applying model trained on oversampled dataset

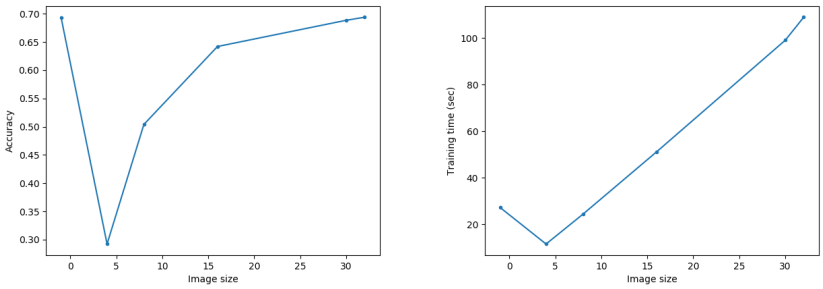

Figure 6: Dependence of accuracy and time on image size

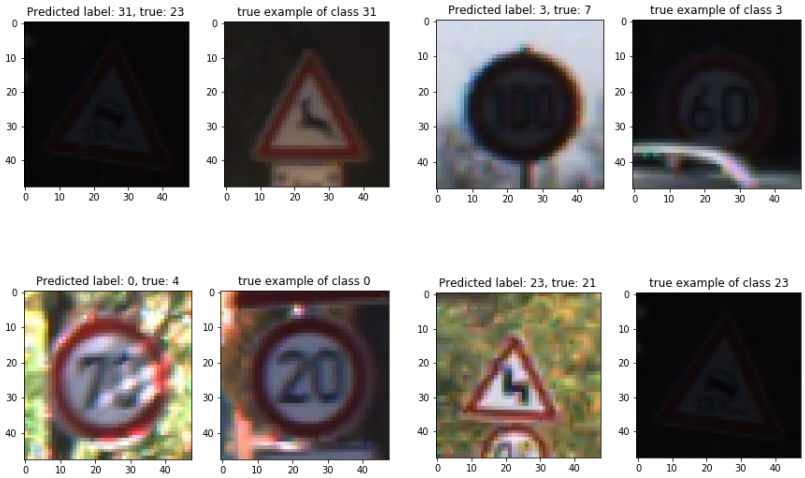

Figure 7: Incorrect classified images