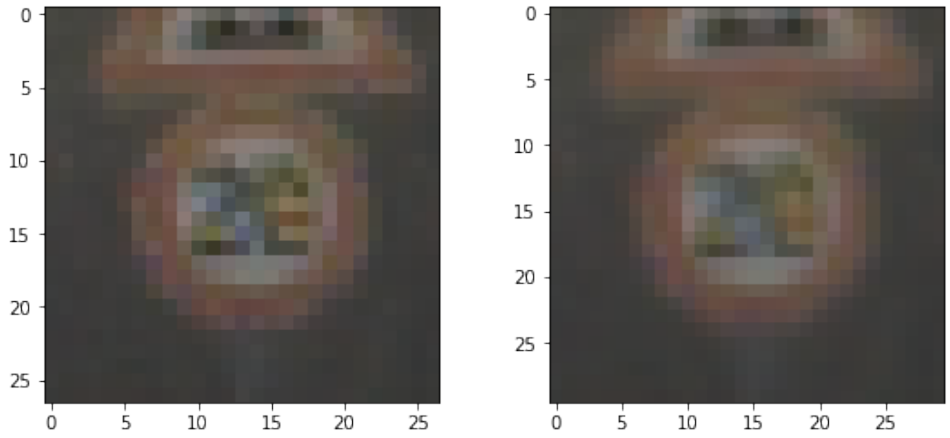
The goal of the report is simple: to inform what I have done and what my results are.

I am trying to build a model, which will classify traffic signs. This is one of the most important task for self-driving cars, because they need to recognize signs and follow them.

I built the model following the **next steps:**

1. Downloading dataset, which includes 39.119 images. One image was missed, that is why I copied one similar image and pasted. Now I have 39.120 images of traffic signs.

2. Dataset includes images with different shape and size, in order to produce the same features from all images I added zeros pixels to the shortest size and made reshape them to one dimensionality - 30 30 pixels. Example before reshaping and after (Pic 1).

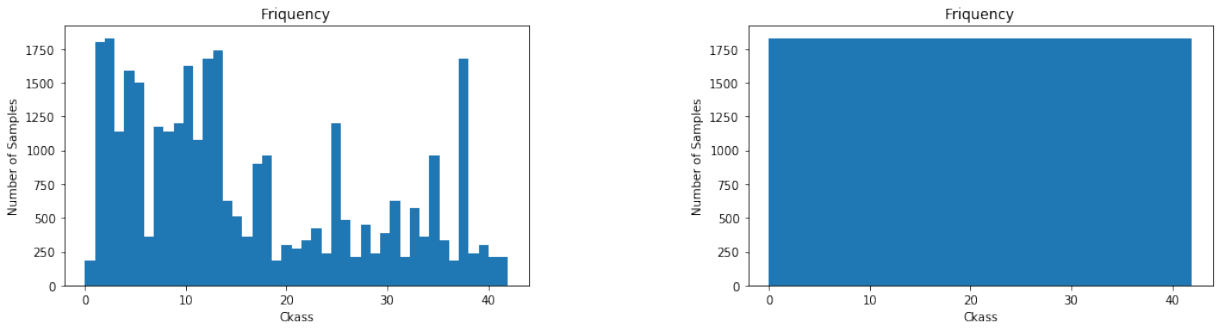


Picture 1 (Before - Left / After - Right)

3. Splitting data on train and validation sets (80% - 20%). Actually the splitting was on 81.56% and 18,44%.

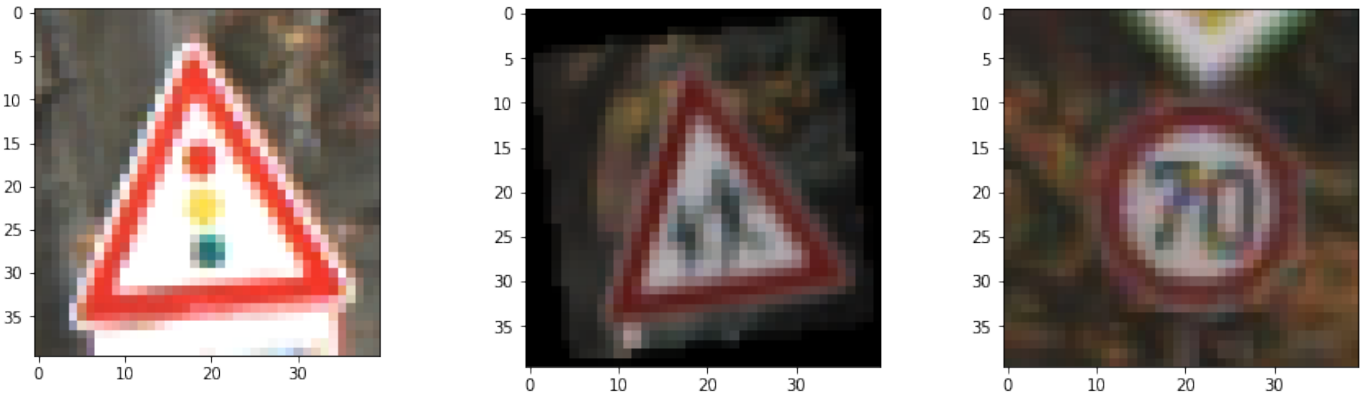
4. Building histogram in order to find class frequencies. In one class I have more examples and in another less, so the classes are highly imbalanced. I need to produce synthetic images from the existing ones and add them to the training set. I decided to rotate images on small angle and add noise. I can not rotate on 180 degrees and flip images, because if I do

that the image with left arrow becomes right arrow and so on. There are histograms before and after adding new pictures (Pic 2):



Picture 2

Examples of augmentations (Pic 3):



Picture 3

5. I normalized every image separately (all pixel values are between 0 and 1), ravel each picture matrix as 1- dimensional vector with the same size and stacked those vectors to matrix.

6. Training the model using RandomForest.

Results:

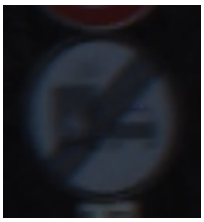
I get accuracy on training set 82%

Overall and precision for each class were close to 0,8 - 0,95. The less value was for 18 and 27 class (Pic 5).

Samples that were classified incorrectly (Pic 5 and 6):

Picture 5

Probably this 2 classes were classified incorrectly, because they both red on the edge and black figure on the black background.



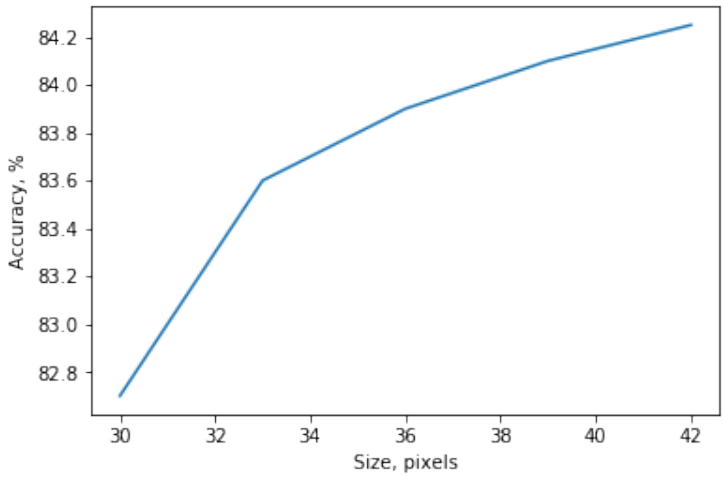
Picture 6

Maybe because most of them were dark in dataset.

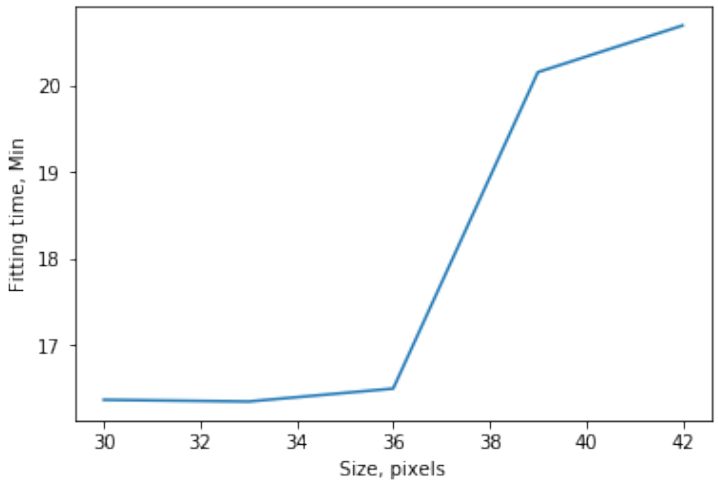
Analize:

If train model on non-augmented data I get 61% accuracy. That is not enough for out task.

What if I change the size of images to different value? Below the graphs of how accuracy and fitting time depend on image size (Graphs 1 and 2)



Graph 1



Graph 2

I changed image size from 30 to 42. Fitting time increase from 16 to 21 minutes. Accuracy from 82% to 84% it is not hight growth, but for cars on the road is very important to classify traffic signs. Even 2% - 3% play an important role.

Conclusion

As a result I got model and this model can recognize 43 different traffic signs with 84 percent accuracy. Model will work even if signs are dirty or it is dark outside. If the future we should use different machine learning approaches and make analyze which method works better for this task.