INTRO TO MACHINE LEARNING

Home Assignment 2: Traffic sign recognition

Sergei Bakaleinik

Innopolis, 2019

1. Downloading the data with cropping

The training data, traffic sign images, was collected in CSV files with next structure: separate packages for each of 43 classes with related images in them. Moreover, their the special "GT-0000" configuration files with data about the images as resolution and size, the position of the traffic sign on the current image and corresponding labels of classes.

For the purpose of increasing the accuracy of the model data about the location of the sign was collected and utilized during the downloading to crop the images – all unnecessary areas of images were cut out.

2. Data Preprocessing: Transformation images to equal size with padding

The padding methodology depended on the sort of the picture which should be resized. On the off chance that one side of the picture is bigger than another, we use kind of the padding which makes the type of picture quadratic. Notwithstanding, for the quadratic pictures padding was utilized as well. Additionally, this padding was utilized for all the color channels because of our aim to utilize the color of the pictures for progressively exhaustive training of the algorithm. Besides, the calculation copies the outskirt pixels to make padding pixels.

For the resizing images to the equal size of 30 x 30 the Pilow open source library was used. Its .resize() method is able to simultaneously expand small images and shrink big one. Moreover, its functions and methods are effective in collaboration with numpy arrays.

Next images resent how the algorithm preprocess incoming image in standard and applicable one.

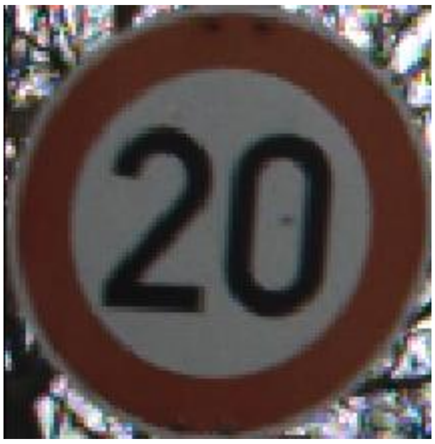


Figure 1 – Image before the preprocessing (first class)

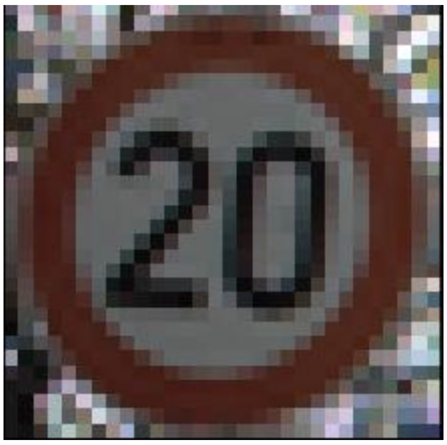


Figure 2 – Image after preprocessing

3. Data splitting (train/test) with augmentation from albumantation

The algorithm goes through all the classes to do the split and augmentation processes. In each class it takes every one of the tracks and the greater part of them (80%) will be added to the training part and 20% for test part. Note that tracks have huge contrasts among themselves in one class – every one of them speaks to various true traffic sign and this is the reason the decent variety among them is huge. For another situation, isolating the elements if the tracks, prompts excessively high precision due to overfitting and makes the application pointless.

Furthemore, the algorithm of splitting use different strategies for training tracks and test ones due to the problem of unbalanced data. On the one hand, for better performance, precision and impartiality of training we need to somehow normalize the training data. On the another hand, we need to not to change the test data, because in real-world the signs are not appear in equal manner and unbalanced structure of the test simulates the real circumstances better than balanced one.

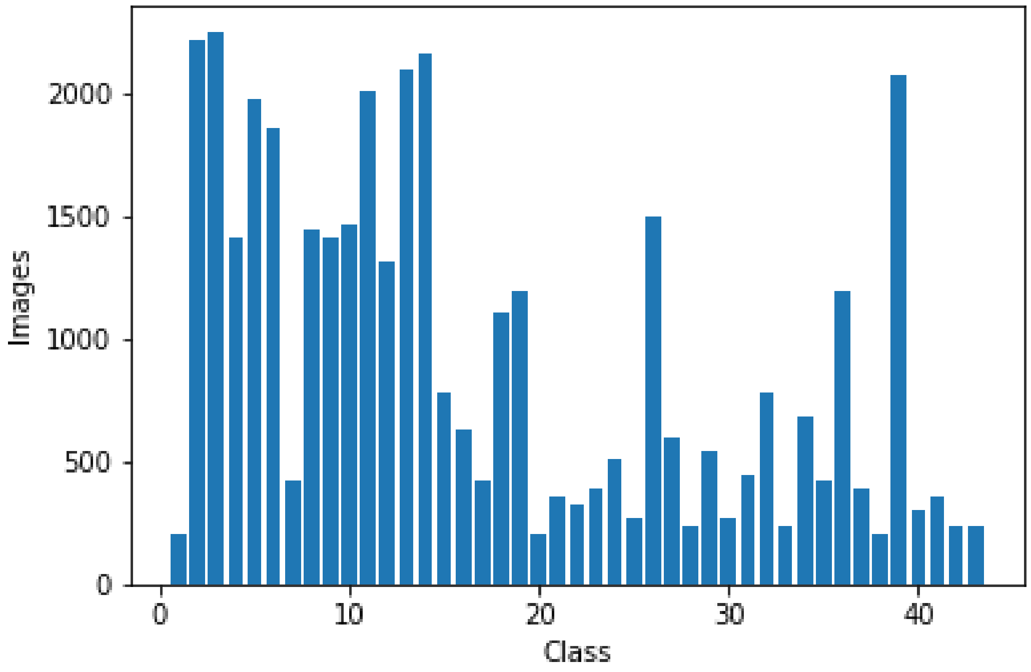


Figure 3 – Current distribution of number of pictures among the classes

So far, for finding the decision to the problem of unbalanced training data, algorithm shrinks the classes, where the number of pictures is too high. It is important to note that algorithm doesn’t cut down whole tracks, but only the number of pictures in the tracks – diversity between pictures in one track is lower than diversity among the tracks. The current threshold for cutting is 750 entities. If a class contains more, the number of pictures in tracks will be shrieked to 15 or 10 based on how many total pictures it has. In addition, the entities, which are will be destroyed, are chosen randomly.

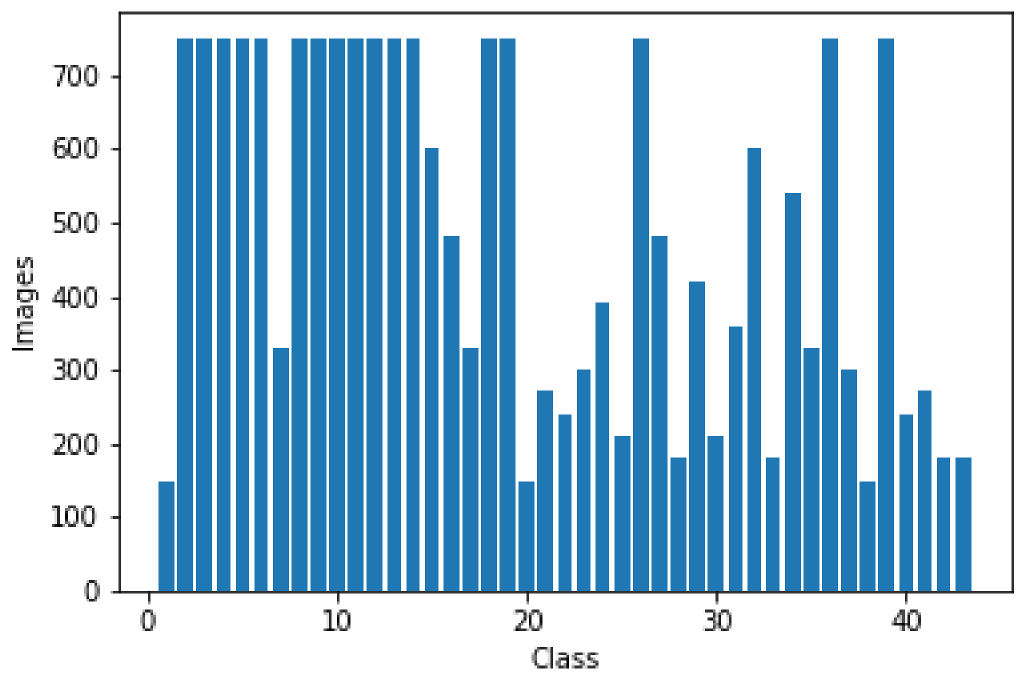


Figure 4 – Result of shrinkage of the training data

Augmentation with albumantation – explanation, justification and examples

During the latest conference (“Stachka) in the Innopolis University one of the authors of albumantation library described how its project works and provide examples and justifications that his library is one of the best. This is why in the algorithm of augmentation the functions from this library is used. Each of augmentations was analyzed by several criteria. First one is the effectiveness of augmentation – is this augmentation makes the picture really different from original one? Second one is how randomly the augmentation can work – is the difference among the results high enough? Third one is problem of quality of the image. augmentation for the training data is used before the resizing for the reason of the quality of augmentation process and this is why the need of checking the impact on the quality is appeared. If augmentation damages the quality of image and makes it harder to recognize the image it will be deleted.

Next twenty images demonstrates the work of augmentations. Examples are provided in pairs – original with original resolution (for the demonstration purposes) and the result of applying each of the augmentation. The list of the augmentations: ElasticTransform, RandomGamma, GridDistortion, RGBShift, Rotate and RandomBrightness.

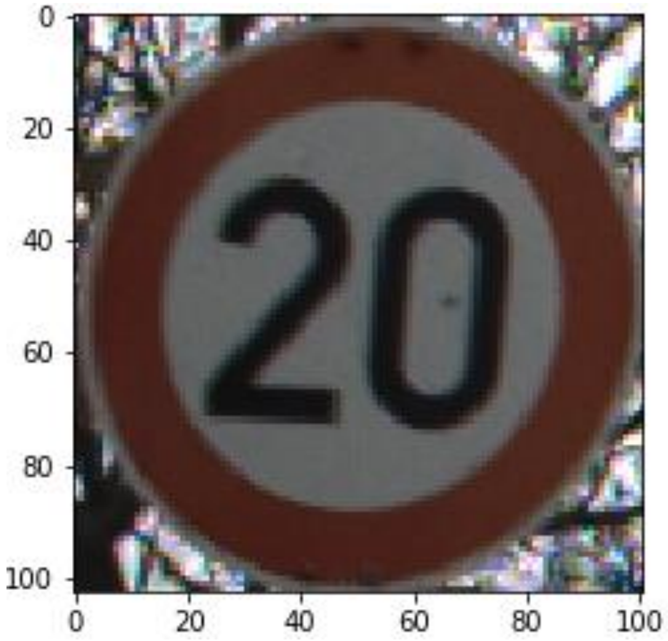
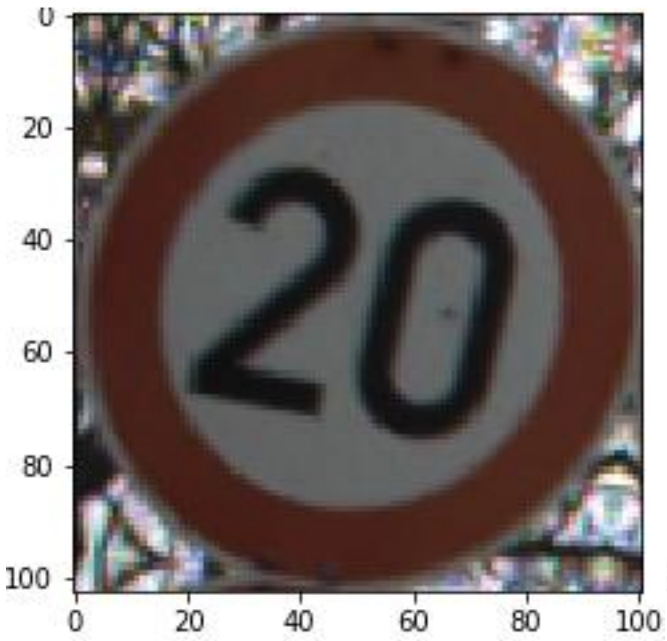
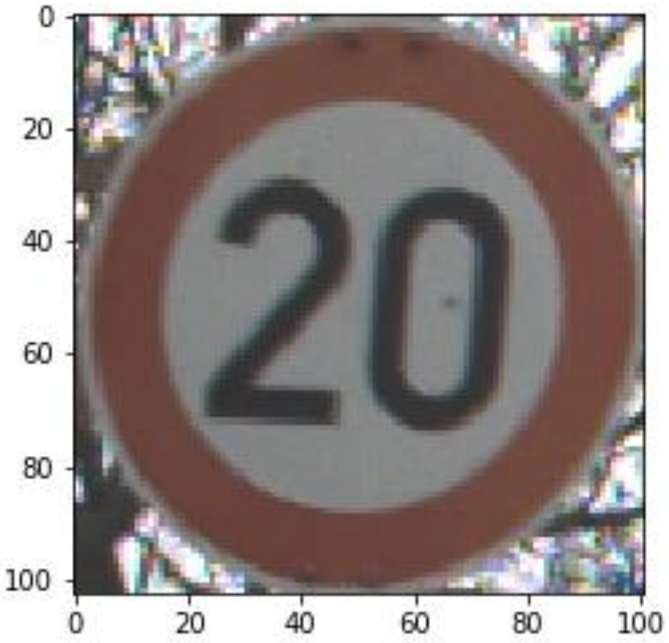


Figure 5 – Original Image





Picture 6 – Six different augmentations

The algorithm of applying the augmentation of the pictures depends on the how much augmented images are needed. If number is significant, the algorithm needs to apply several augmentations on each picture of the class. The augmentations, which will be applied, are chosen randomly from all that the algorithm has, but it doesn’t apply the same twice. If the number is not so big, the algorithm randomly takes the track and element from it and repeat this choosing as much as needed.

4. Image normalization (from 255 to 1) and color issue

Next part of the algorithm makes the normalization of the color channels from 0-255 degree to 0-1. It makes the calculations slight faster and precise due to usage od more precise double parameter instead of rough integer one.

During the implementation the idea of going to the grey scale was decline due to the purpose of improving the accuracy of the algorithm. Firstly, The color channels help to identify the sign structure among in the whole picture. Secondly, it will help to distinguish images more effectively. For example, 80 speed constraint is showed in two different colors – red, white and black (6) and grey one (7). The restriction of joint travel of freight and passenger transport has the same properties (11, 43). Vertical signs like 19 and 27 really hard to distinguish in grey scale too.

5. Evaluation

Next information provides time periods, which are needed to algorithm to compute each of the stages.

Time for reading from csv and cropping: 623.721363 s

Time for splitting test and train data, shrink the train under some threshold and augmentation process: 65.689474 s

Time for random forest fitting under the augmented training data with 30 x 30 size images: 341.306502 s

Time for model predicting: 341.306502 s

Total accuracy: from 73 to 75 percentages due to randomness of the algorithm stages.

Processor in the laptop, which was used to calculations: AMD A9

Nevertheless, the estimation of an accuracy of algorithm isn’t so representative for all classes what we have – some of them are poorly represent. This is why the confusion matrix was calculated and demonstrated for the purpose of identifying the problems areas firstly and analyzing them after.

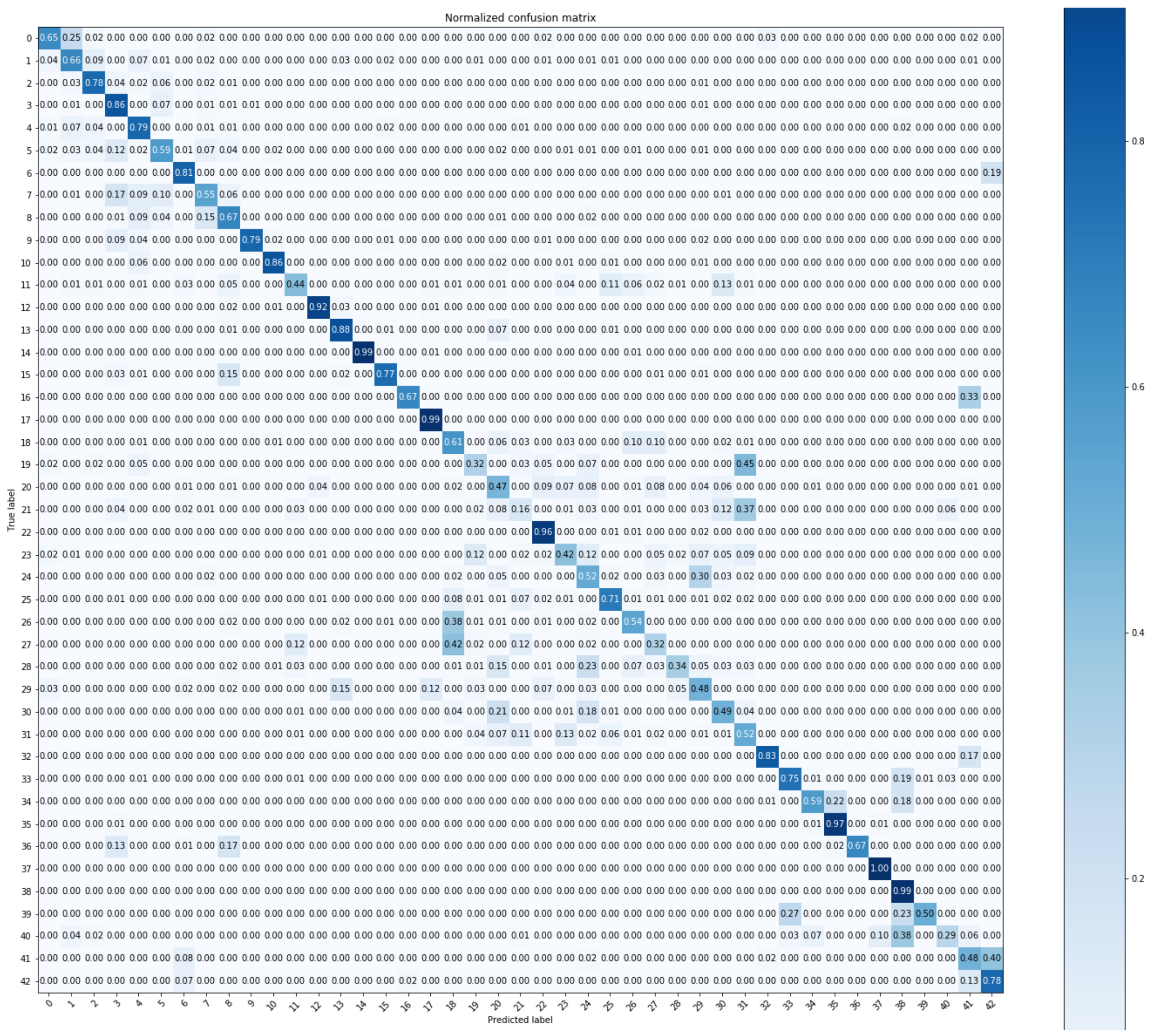


Figure 7 – Confusion matrix

The situation with the pictures that were mixed up with each other complements the situation with the choice in favor of color pictures. On the one hand, this solution helped, on the other hand the speed limit signs (the first 8 pictures), turns, signs that are a mirror image of each other, differ only in color or a crossed line are still confused with each other.

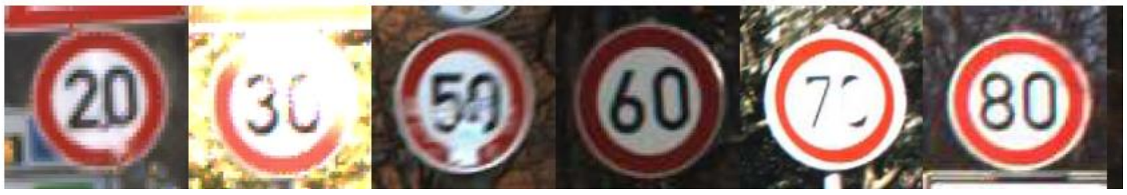


Figure 8 – first group of confused images

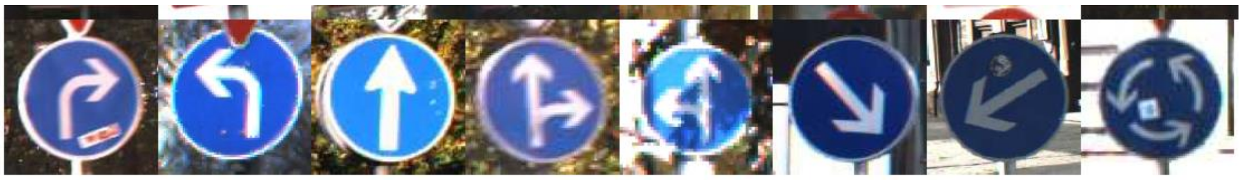


Figure 9 – Turns, second group of confuse. The mirror images are included in this group too

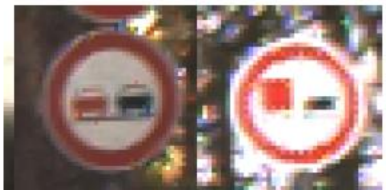


Figure 10 – These pictures are close to each other

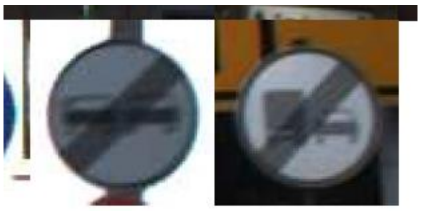


Figure 11 – These pictures differ from previous group only color and crossed line

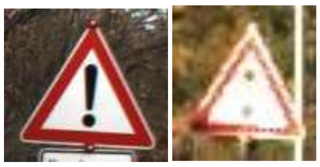


Figure 12 – They are close in both shape and color

Experiment -> Analyses -> Conclusions

The compare between augmented and non-augmented showed that the performance doesn’t change significantly (it takes only 30 to 40 seconds) but increase the accuracy quiet well compare to consuming time – from 3 to 5 percentages (variance is existing due to random in many stages of algorithm). This is why the effectiveness of this approach is approved.

During the experiment part number of sizes were used. However, the results are not good. In case of increasing the size of picture to 50 x 50 for example, we received the 2.7 takes bigger arrays for calculations and huge grow in computational time due to Big O complexity of random forest, and only several percentages in accuracy.

In other case, then algorithm tries to use smaller images, the accuracy go down **rapidly** and mostly it is explained by loss the structure of the signs, shapes and color differences too.

In such case, using another sizes of images is useless and can lead only to huge decreases in quality ore huge increases in computational time.

6. References

1) The German Traffic Sign Recognition Benchmark: A multi-class classification competition – Official paper of competition, provide the sense of the competition, information about data structure and how it was collected.

2) Open source augmentation project from well-known champions of Kaggle competition. https://github.com/albu/albumentations with examples and documentation

3) Non-Technical Guide to Writing a Report. https://hackmd.io/J89FQKJdRWmZCtku1pP04w?view – Guide for writing the report in the right way