

Homework Report

General

Author: Ruslan Sabirov

Group: B17-DS-02

Changes

I have changed provided code for reading to support reading testing data.

I have placed Training data in the structure like `./data/Final_Training/Images/{image_class}/{image_name}`.

I have placed Testing data in the structure like `./data/Final_Testing/{image_name}`.

I have added one more picture to the class 33 track 19 so that there are 30 pics for each track.

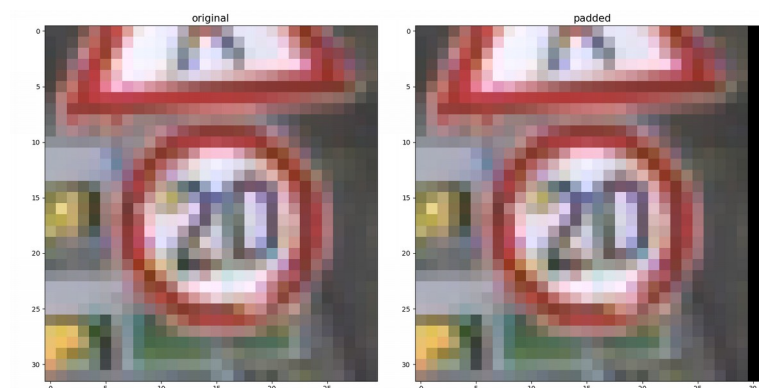
Problems

I have faced these problems:

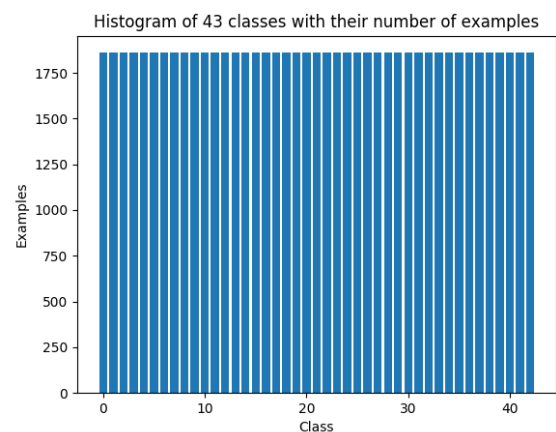
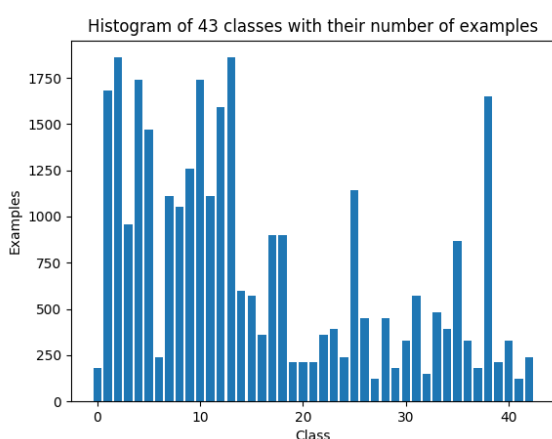
- 1) Not enough RAM to store all the data (only 8 Gb). Solved by reducing amount of proceeding data, making functions in-place and clearing lists after usage.
- 2) Code for reading files was not compatible for testing data. Added additional parameter (indicator) to the function and proceed it according to the value of the indication.
- 3) Could not found right values for `max_lenght` and `n_estimators`. Did hyperparameters tuning.

Padding

Added zero (black) pixels to make weight and height the same.



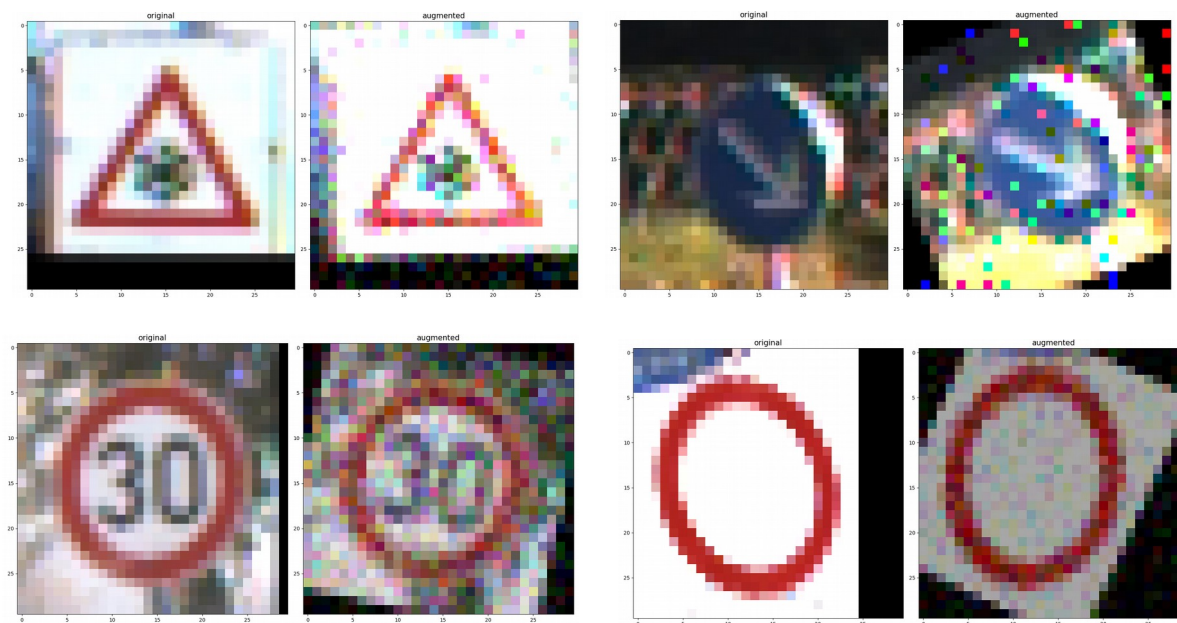
Frequencies



Augmentation

I have done 3 transformations on random sample:

1. Rotation to the angle from -25 to 25. Physical traffic signs could be really rotated because of bad binding or strong wind or some accident.
2. Noising random pixels (salt & pepper approach). This could happen when there are some problems during printing or problems with camera.
3. Changing brightness (increased or decreased from 1 to 3 times). This could happen during taking a picture because of wrong focus or problems with lighting.



Evaluation

Overall accuracy: 0.6209241848369674

This is table shown by `sklearn.metrics.classification_report()` on testing data

	precision	recall	f1-score	support					
0	0.00	0.00	0.00	1634	21	0.12	0.07	0.09	1290
1	0.54	0.70	0.61	12943	22	0.46	0.62	0.53	1806
2	0.42	0.65	0.51	12599	23	0.37	0.35	0.36	2365
3	0.36	0.35	0.35	6450	24	0.17	0.04	0.06	1204
4	0.48	0.60	0.53	11008	25	0.64	0.73	0.68	8686
5	0.39	0.43	0.41	10750	26	0.55	0.52	0.53	2580
6	0.50	0.30	0.37	2322	27	0.00	0.00	0.00	1075
7	0.48	0.51	0.49	8213	28	0.63	0.43	0.51	2709
8	0.38	0.35	0.36	7396	29	0.82	0.34	0.48	1763
9	0.82	0.71	0.76	7869	30	0.38	0.16	0.22	2752
10	0.87	0.88	0.87	11782	31	0.54	0.49	0.51	4601
11	0.76	0.81	0.79	7095	32	0.45	0.24	0.31	903
12	0.86	0.84	0.85	12470	33	0.65	0.67	0.66	3354
13	0.87	0.95	0.91	12083	34	0.74	0.73	0.73	2365
14	0.89	0.82	0.86	4816	35	0.78	0.50	0.61	6579
15	0.54	0.54	0.54	3612	36	0.76	0.65	0.70	1849
16	0.98	0.95	0.97	2666	37	0.93	0.72	0.81	774
17	0.95	0.73	0.82	6794	38	0.73	0.80	0.77	11137
18	0.48	0.42	0.45	6106	39	1.00	0.62	0.77	1720
19	0.00	0.00	0.00	1075	40	0.85	0.82	0.84	1720
20	0.00	0.00	0.00	1505	41	0.33	0.19	0.24	1118
21	0.12	0.07	0.09	1290	42	0.53	0.30	0.38	1419
					accuracy			0.62	214957
					macro avg	0.56	0.50	0.52	214957
					weighted avg	0.62	0.62	0.61	214957

Incorrectly classified



Most of the images that were misclassified had low quality, low brightness or were taken from perspective with big angle.

Experiments and analysis

I have run algorithm 10 times to make experiments. If was either augmented or non-augmented data and there were 5 kind of image sizes: 10x10, 15x15, 20x20, 25x25, 30x30.

Augmented approach performs better, but needs to much time to perform. Thus if time is limited, augmentation should be ommited or performed much better.

At the beginning as image size increases, accuracy increases much. But at the end not so much.

