

WEIGHT LOSS PERFORMANCE COMPARISON

Comparison description

In this comparison, it will be analyzed the effect of the different balancing of the two main classes (*object*, *background*) on the overall performance of the network.

To perform the balancing during the dataset loading procedure the script automatically computes the ratio between the number of background pixels and the number of objects ones, throughout the whole dataset:

$$r = \frac{\#ofbackground'pixels}{\#ofobjectpixels} = \frac{p_{BG}}{p_{OBJ}} \approx 30.$$

Then, during the MAE (Mean Absolute Error) loss computation, before computing the mean over the whole image matrix, the loss value of each object pixel (considering the *GT* as a reference to locate the object position) is multiplied by the aforementioned ratio r . By doing that we're essentially balancing an unbalanced dataset giving the same weight to both classes. This turns out to not be the best solution since it is better to reduce the disparity between the background and the object pixels. For that reason we've decided to multiply the ratio r by a factor α in the range $[\frac{1}{100}, \frac{1}{10}, \frac{1}{9}, \frac{1}{8}, \frac{1}{7}, \frac{1}{6}, \frac{1}{5}, \frac{1}{4}, 10]$.

Doing that the final MAE loss (for each pixel) is computed as follows:

$$\alpha \cdot r \cdot loss_{OBJ};$$

$$1 \cdot loss_{BG}.$$

The following are shown the results obtained by testing the network with different values of α .

Results

attempt #	α	avarage MAE (depth)	min MAE (depth)	max MAE (depth)	avarge IoU (depth)	min IoU (depth)	max IoU (depth)
1	1/100	0.0826	0.0379	0.1509	0.4837	0.4668	0.4935
2	1/10	0.1168	0.043	0.2467	0.6209	0.4691	0.8336
3	1/9	0.0698	0.0129	0.1552	0.7287	0.4752	0.9399
4	1/8	0.0585	0.0236	0.1444	0.7606	0.4735	0.9192
5	1/7	0.0632	0.0205	0.1867	0.7534	0.4764	0.9066
6	1/6	0.0777	0.0241	0.1781	0.7278	0.4671	0.8826
7	1/5	0.0807	0.04	0.2104	0.7335	0.4627	0.8773
8	1/4	0.0707	0.0273	0.2108	0.7448	0.4621	0.8901
9	10	0.3364	0.1652	0.5378	0.5414	0.4349	0.7089