

KW45 NiftyNet

Results of the training are here.

Inference Settings:

- Full CT (full, half, quarter size) no foreground selection is done
- border (16,16,16) for 96x96x72/1 trained images
- border(8,8,8) for 48x48x48/8 trained images
- Checkpoints from 12000 to 49999 every 2000 iterations

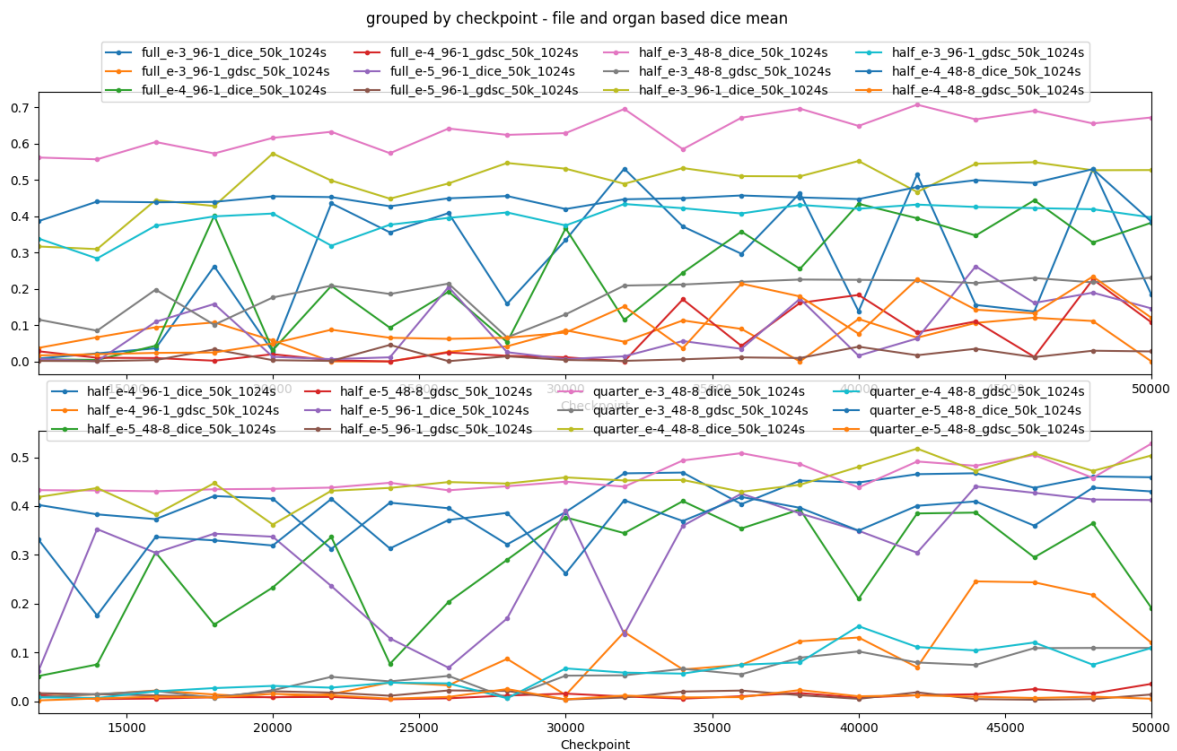
Maybe still to less iterations 25 images a 1024 samples (label ratio estimated - should still be calculated but around 0.01 for half of the samples, the other half does not contain info) = 25000 iterations to see every image when batch size is 1!!

Results generated with codebase at ibia git - param_tuning_v1

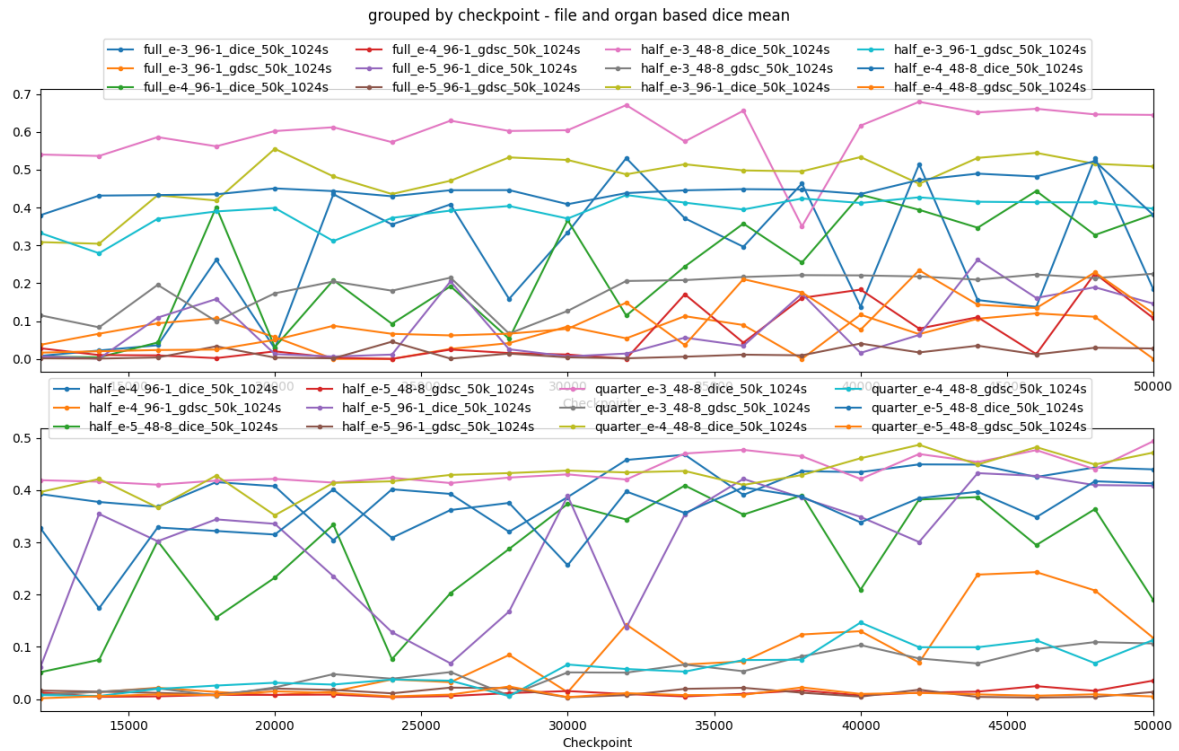
Results are collected with the collect_results.py script, using the PairwiseMeasurement script from Nifitynet and written to a csv file with the following header:

Model,Checkpoint,File,Organ,Dice

From the csv the evaluate_results.py script graphs are created:



First evaluation. Reduced size was used to calculate the dice, which is **wrong** as the full sized results (**upsacle after inference**) has to be used to get the correct results!!!

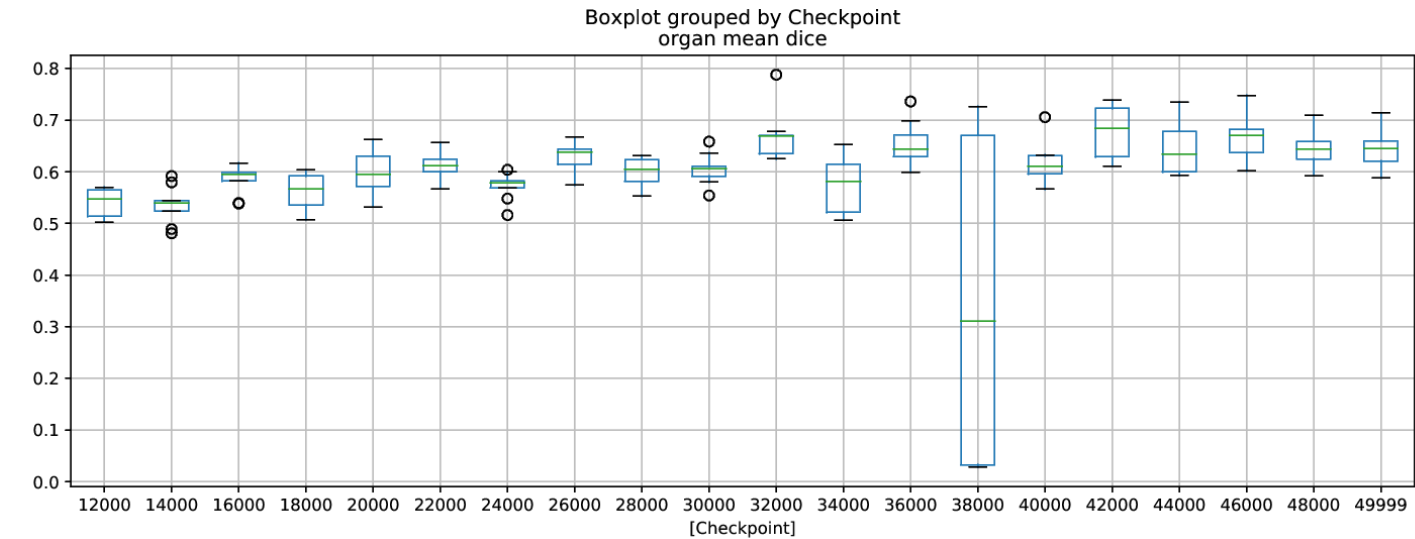


Second evaluation. Upscaled results are used.

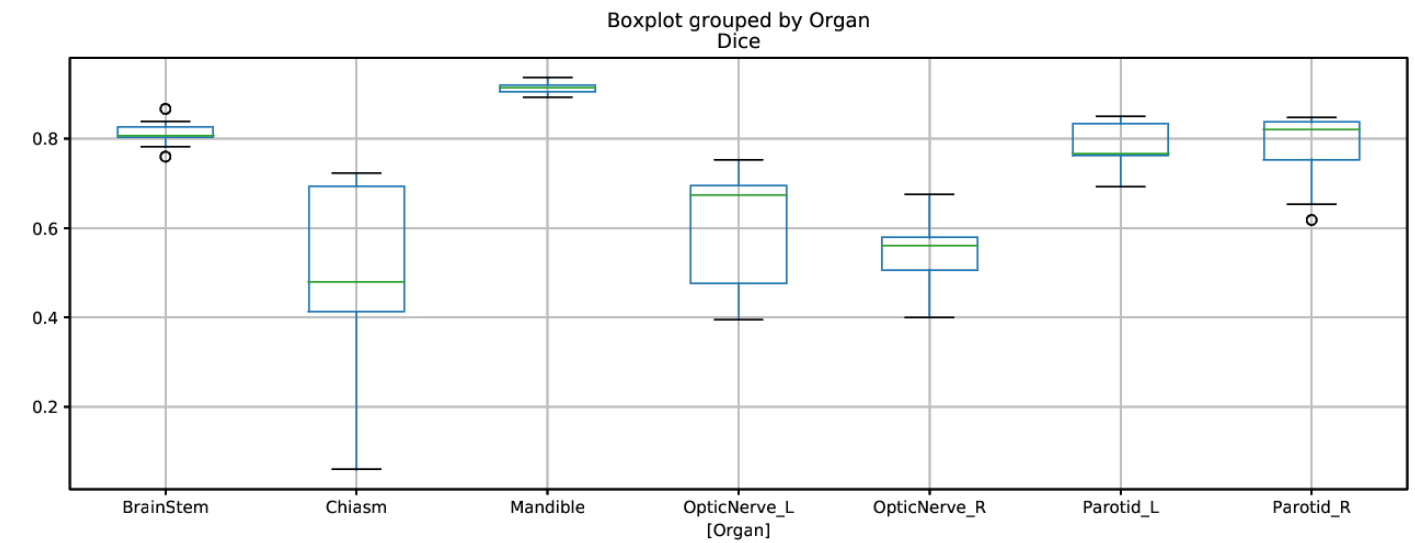
The figures shows the test performance (dice) for each checkpoint averaged over all organs and files.

Best test performance:

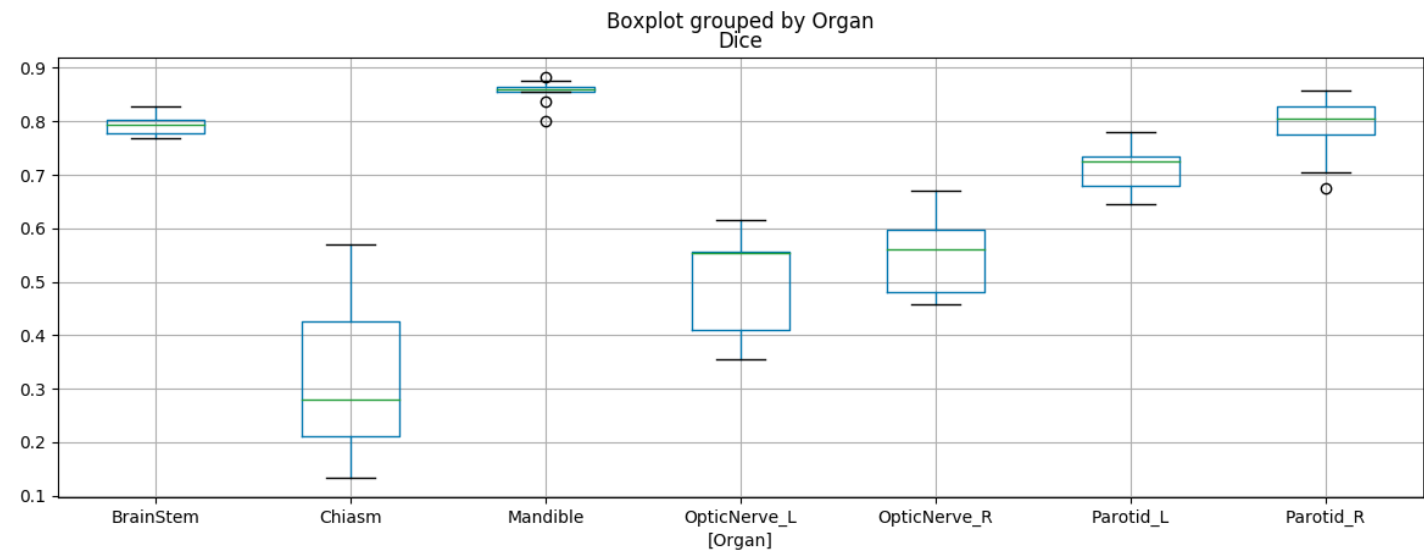
half_e-3_48-8_dice_1024s



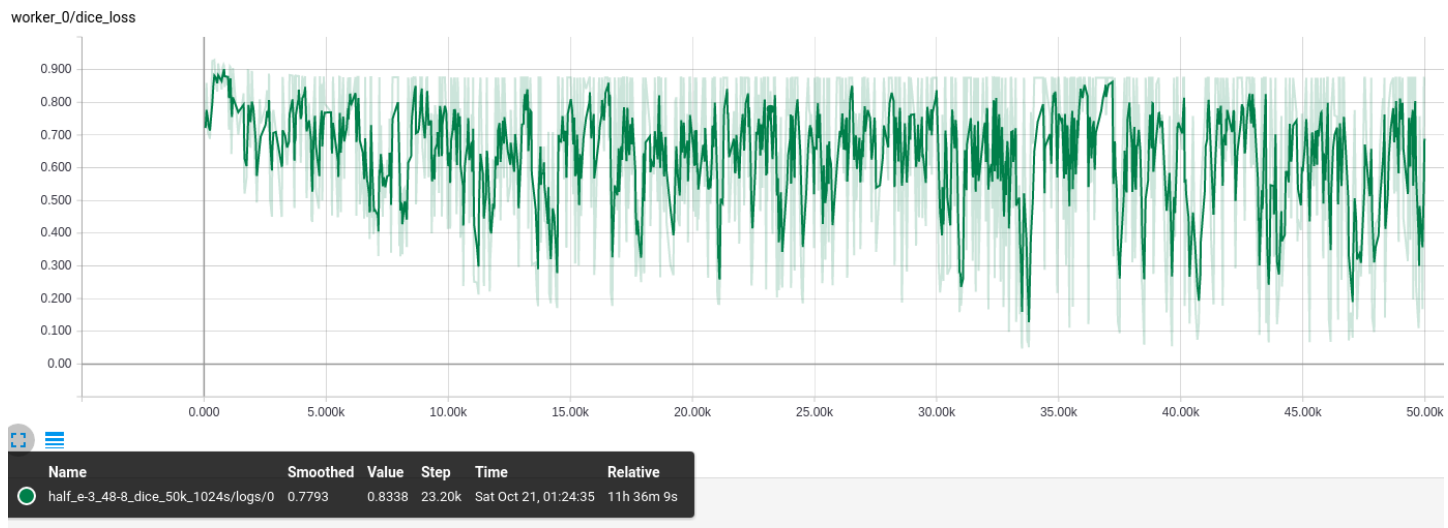
Test result of half_e-3_48-8_dice_1024s averaged over all organs for each checkpoint.



Detailed results for iteration 42000 -> best over all organs average!!



Detailed results for iteration the last iteration 49999!!



Training performance!!

Interpretation Test performance:

- best model on average organ and file performance: half_e-3_48-8_dice_1024s
- generalized dice generally worse results (maybe due to foreground background ratio)
- Loss functions all not really converging (see tensorboard)
- Dice loss function gets best results with a learning reate of 0.001

Todo:

Find out how to calculate foreground-background ratio of random sampling process after training?? Maybe that can explain some results better?

Apply random sampling process to all volumes of all sizes with the different patch sizes to get an estimate of the imbalance ratio!!

Remake figure to get a better overview and understanding of the results:

- group by parameters
- bad results can be removed
- just print bigger sized and smaller organs separately
- reinterpret

Officially ground truth for the test set are not available