

Method

Architecture. Faster R-CNN with FPN with ResNet-50 as a backbone. The weights were initialized from the provided baseline model. The experiment is implemented on a GeForce RTX 2080 Ti GPU.

Anchor. Anchor sizes were set to [16, 32, 64, 128, 192] and aspect ratios were set to [0.8, 1.0, 1.25].

Augmentation. Before training, we copied the positive cases three times in the images file to make the balance of the positive and negative cases (using `training_utils/copy.py`). During training, we adopt RandomHorizontalFlip and Multi-scale sampling (shorter side=800~1333) for all cases and RandomCrop for positive cases. RandomCrop first randomly cropped a patch from the input image, which included all the corresponding bboxes. Then, cropped image were zoomed to the input size. During testing, the results from RandomHorizontalFlip and Multi-scale sampling (shorter side=800, 1333) were mixed by Weighted-Boxes-Fusion. Note that we did not use Non-Maximum Suppression.

Training strategy. We trained 16 epochs with image batch-size of 2. The learning rate started at 0.0025, and reduced by a factor of 10 after every 4 epochs.