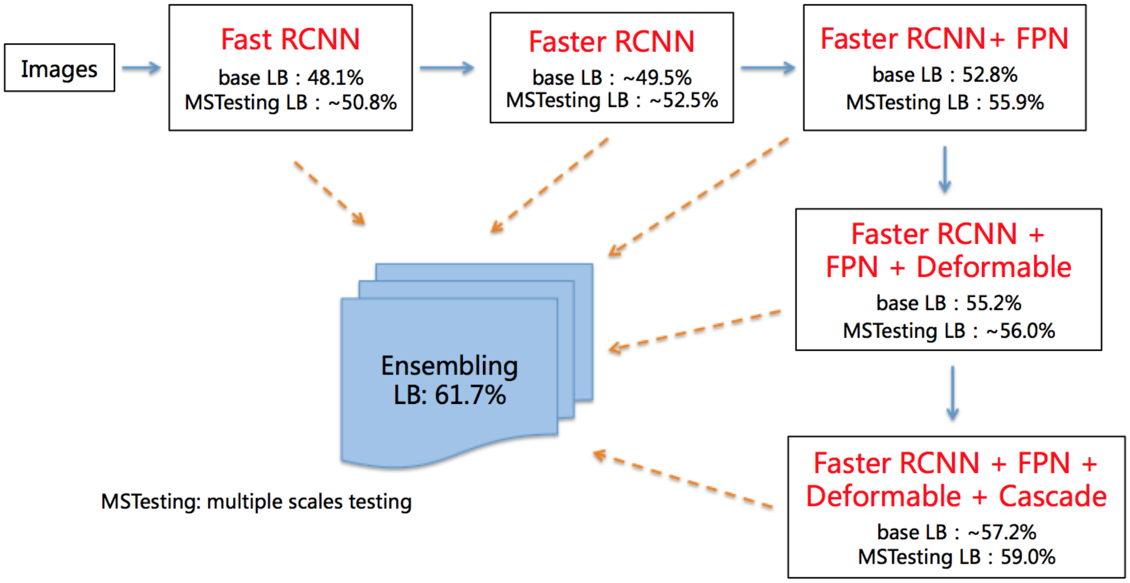
Here’s our team’s solution to the competition.

Summary

We trained detectors with different levels of complexity, to examine the performance of each detector and also to get a large model zoo useful for ensemble. Common two-stage detectors were investigated during our experiments: Fast R-CNN, Faster R-CNN, FPN, Deformable R-CNN, Cascade R-CNN. One-stage detectors such as SSD, YOLO, RetinaNet were not considered for their relative low performance.

The following figure shows our main framework: (Note: some scores are estimated from the gap between LB and local validation set)



The performance of each detector on public leaderboard evolved as follows:

1) Fast R-CNN[1] with ResNet-101, backbone was trained to get a baseline mAP 0.481. After applying test-time tricks (which will be explained later), the mAP reached 0.508. We further trained different backbone networks such as dpn98, inception-v4, seresnext101. The mAP of ensemble of different backbone models increased to 0.546. In Fast R-CNN, the proposals were generated from the following steps: first, we densely sampled candidate proposals with different scales and locations, then a network was trained to classify the candidate proposals and adjust the locations of them.
2) Faster R-CNN[2] was trained to reach 0.495, which is slightly higher than Fast R-CNN. After applying test-time tricks, the mAP increased to 0.525.
3) FPN’s[3] mAP was 0.528 and 0.559 before and after applying the test-time tricks.
4) Deformable R-CNN’s[4] mAP was 0.552 and 0.560 before and after applying the test-time tricks.
5) Cascade R-CNN’s[5] mAP was 0.572 and 0.590 before and after applying the test-time tricks.

Only Fast R-CNN was trained using multiple backbones. Other models were trained on ResNet-101 due to the computation resource and time limit.

Details

1. Dynamic Sampling for Object Detection

There are about 1.7 million images in Google Open Image V4 Dataset， the largest class has hundreds of thousands of bounding boxes, and the smallest class has only a dozen. if using all images and its bounding boxes, it will take dozens of days to converge. So we should using some important images for model training.

We use three different ways to sample data. In the first strategy, we use the full set of data sets for training, the second strategy trains subsets of data for training, and the last one performs sampling according to dynamic sampling. The general idea of the dynamic sampling strategy is that the number of bounding boxes in this category is less than a certain threshold, and all the boxes are used. When the bounding boxes of the category is greater than a certain threshold, only a fixed amount of pictures will be sampled in each epoch. The performance of the three methods.

- Full Training Data: 0.50 mAP.
- Fixed Training Data: 0.53 mAP.
- Dynamic Sampling Training Data: 0.56 mAP.

2. Feature Pyramid Networks [3]

Under the accumulation of Fast R-CNN, Faster R-CNN has already integrated feature extraction, proposal extraction, proposal bounding box regression, classification into a network. In the competition, we trained a model based on Faster R-CNN with dynamic sampling data, the metric mAP is 0.495, and the result through multi scale testing is 0.525.

However, through the analysis of training data, we find that the scale of 500 categories has large differences. Therefore, FPN is introduced into the model, that is, the multi-scale and multi-level pyramid structure is utilized to build the characteristic pyramid network. In the experiment, we take Resnet101 as the backbone network, a top-down side connection was added in the last layer of different stages. The top-down process is carried out by upsampling, while the horizontal connection is to merge the results of the upper sampling and the feature map of the same size generated from the bottom up. After fusion, 3*3 convolution checkup is used for each fusion result to eliminate aliasing effect of upper sampling. It is worth noting that FPN should be embedded in RPN network to generate different scale features and integrate them as the input of RPN network. The result of FPN is 0.528 and the result of multi scale testing is 0.559.

3. Deformable Convolution Networks [4]

We adopted deformable convolution network to enhance the transformation modeling capability of CNNs. The deformable convolution network is based on the idea of augmenting the spatial sampling locations with additional offsets modules, which is learned from the target tasks without additional supervision. We applied the deformable convolution network on Faster R-CNN architecture. The deformable convolution layers are applied after the res5a, 5b, 5c layer of ResNet-101, and the deformable position-sensitive ROI pooling is also applied. The ResNet-101 model is pre-trained on ImageNet classification dataset. The result of deformable convolution network on the public board is 0.552, and about 0.560 for multi-scale testing.

4. Cascade R-CNN [5]

We use Cascade R-CNN to train object detection models. Except training the basic models, FPN(feature pyramid networks) with 5 scales and 3 anchors are used. The second difference is that we trained a small class of 150, which is the worst categories for the full set model training. We used these 150 categories for evaluation, the MAP for the large full model is 0.477, while the model for the subset-single model training is 0.498. And the performance of the full single scale model is 0.573.

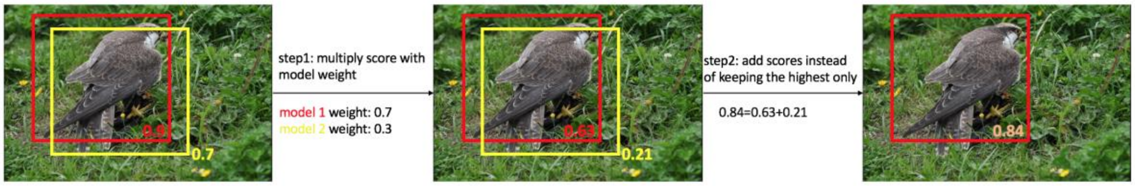
5. Test-time Tricks

In the post-processing stage, we use soft-NMS and multi-scale testing. The soft-NMS has about 0.5 to 1.3 points improvement on different models, while multi scale testing has about 0.6 to 2 points improvement on different models.

6. Model Ensemble

For each model, we predicted the bounding boxes after NMS. Boxes from different models were merged using a modified version of NMS as follows:

1) Give each model a scalar weight between 0~1. All weights sum up to 1.
2) The confidence score of boxes from each model is multiplied by its weight.
3) Merge boxes from all models and run the original NMS, except that we add scores from different models instead of keeping the highest one only. The IOU threshold is 0.5 in this step.


0.9
0.7
step1: multiply score with model weight
model 1 weight: 0.7
model 2 weight: 0.3
0.63
0.21
step2: add scores instead of keeping the highest only
0.84=0.63+0.21
0.84

Reference

[1] R. B. Girshick. Fast R-CNN. In ICCV, pages 1440–1448, 2015.

[2] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal net- works. In NIPS, pages 91–99, 2015.

[3] T.-Y. Lin, P. Dolla ´r, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In CVPR, 2017.

[4] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In ICCV, 2017.

[5] Z. Cai and N. Vasconcelos. Cascade R-CNN: Delving into High Quality Object Detection. In CVPR 2018.