2nd Place Solution in Google AI Open Images Object Detection Track 2019

Ruoyu Guo, Cheng Cui, Yuning Du, Xianglong Meng,
Xiaodi Wang, Jingwei Liu, Jianfeng Zhu, Yuan Feng, Shumin Han
Baidu Inc.
{guoruoyu, duyuning}@baidu.com

Abstract

We present an object detection framework based on PaddlePaddle. We put all the strategies together (multi-scale training, FPN, Cascade, Dcnv2, Non-local, libra loss) based on ResNet200-Vd backbone. and our model score on public leaderboard comes to 0.6269 with single scale test. We proposed a new voting method, called top-k voting-nms, based on the SoftNMS detection results. The voting method helps us merge all the models' results more easily and achieve 2nd place in the Google AI Open Images Object Detection Track 2019.

1. Introduction

Open Images Detection Dataset V5(OIDV5) is the largest existing object detection dataset with object location annotations. It contains over 1.7M images with 500 classes and over 12M annotated boxes, which is far more than current famous datasets, such as, COCO [1], Objects365 dataset [2].

Besides large-scale and closing to real scene, OIDV5 has other typical characteristics:

The annotations of instances in OIDV5 are not overall, and for each image, instances of classes not verified to exist in the image are not annotated. It leads to treating a bounding box as the background when an unverified instance is inside the box.

Besides, the images in OIDV5 have large-scale variation. Figure 1 shows histogram of bbox areas and their numbers. Most boxes only account for 10% of the images, however some boxes hold the whole images. So the scale of annotated boxes is very extremely imbalanced.

The most distinct characteristic of OIDV5 is category imbalance. 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.

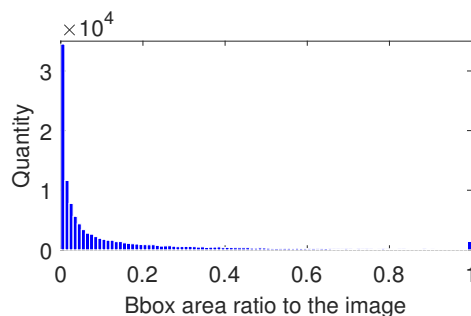Figure 2 shows the framework of our solution. It contains four key points: (1) we train an extraordinary single model with Res200-vd [5, 6], FPN [8], Dcnv2 [14], Cascade [4], Non-local [13] and libra loss [10] based on PaddlePaddle. (2) Since the 189 classes are same between OIDV5 and Objects365, we combine these two datasets to enhance the ability of the above model. (3) To further promote the performance, the detectors with the different backbones and architectures are trained. (4) We propose a modified version of NMS to ensemble models. The ensemble box is obtained by the topk bounding-boxes voting. It will make better use of the above trained models.

Figure 1. BBox areas and their numbers

2. Baseline

We mainly leverage Faster RCNN [11] framework for the object detection task. Backbone is vital to the model performance, especially for large datasets. ResNet200-Vd is utilized as backbone for our baseline model. ResNet-Vb, ResNet-Vc and ResNet-Vd structures are shown as Figure 3.

The pretrained model is based on ImageNet-1k. In the training phase, we used the mixup and label smoothing strategies to avoid overfitting and enhance the robustness of the model. In the end, the top-1 accuracy of ResNet200-vd on ImageNet-1k is 80.93%. It is worth mentioning that ResNet-Vd performs better than ResNet-Vb generally in classification and object detection task. The top-1 accuracy of ResNet50-vb and ResNet50-vd are 76.50% and 79.12%, respectively.
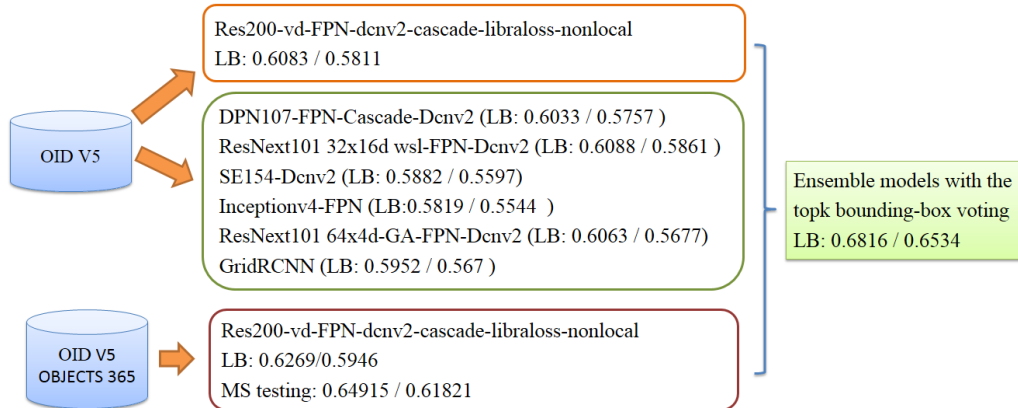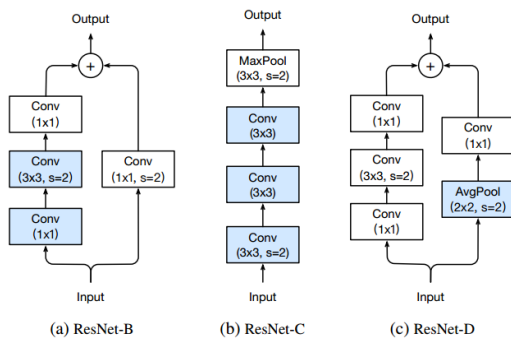
Figure 2. Framework of our solution.

Figure 3. Different ResNet structures.(From left to right, they are ResNet-Vb, ResNet-Vc and ResNet-Vd)

Feature Pyramid Networks is vital to the model performance, especially for small objects detection. Cascade RCNN can help us to obtain a detector with higher quality. Deformable ConvNets V2 has the ability to focus on pertinent image regions. Non-local Network can aggregate feature map context information. Compared to smooth-l1 loss, libra loss has the advantage of balancing the losses of inliers and outliers. We train a detector using all the above methods based on PaddlePaddle. To validate the model's effectiveness, we trained the same model on COCO dataset. The final mAP on COCO minival 2017 set reached 51.3%.

OIDV5 dataset has serious problem of class imbalance. Therefore, we adopt dynamic sampling strategy during training. For better performance on the various sizes of objects, we perform multi-scale training. To improve recall, we use SoftNMS [3] rather than NMS for the detector post processing. If not specified, all the models are evaluated

Model	Score(public/private)
ResNet200-vd-base	0.5860/0.5563
ResNet200-vd-FPN	0.5925/0.5666
ResNet200-vd-All	0.6083/0.5811

Table 1. Trained models based on ResNet200-Vd.

using single scale with SoftNMS.

For the baseline model, we utilize 8 NVIDIA Tesla V100 cards to train our models. What's more, piecewise decay strategy is adopted for the learning rate schedules. Learning rate is set as 0.01 and reduced at 10^6th and $1.4x10^6$th minibatch with the total iteration number set as $1.5x10^6$.

The final scores on public and private leaderboard are shown in table 1. To validate the backbone performance, the ResNet200-vd-base and ResNet200-vd-FPN detector based on Faster RCNN model are also trained. ResNet200-vd-FPN also performs better. Here ResNet200-vd-All denotes the detector is trained using FPN, Cascade, Dcnv2, Non-local, libra loss based on ResNet200-vd backbone.

3. Application of Objects365 Dataset

Objects365 is a brand new dataset for object detection research and it has 365 classes of objects, We find that about 189 classes between OIDV5 and Objects365 dataset are same. This helps us expand dataset more easily. We take two ways to use the Objects365 dataset.

3.1. Pretrained model

We train a ResNet200-Vd-FPN model based on Objects365 Dataset, then use the trained model as pretrained

Model	Score
Pretrained Objects365 Dataset model	0.6000/0.5708
Trained with Joint Data	0.6269/0.5946
Trained with Joint Data(MS-testing)	0.6492/0.6182
All models using ResNet200-vd(voting)	0.6582/0.6287

Table 2. Results based on Objects365 Dataset.

model to train a ResNet200-vd-All model for OIDV5. It's noted that the model is just trained 10^6 minibatches because of the time limit. As can be seen in Tabel 2, the model score on public and private leaderboard is 0.60, 0.57076, respectively, which are lower than ResNet200-vd-All model.

3.2. Train Models with Joint Data

We use category mapping for the Objects365 dataset. In detail, we expand OIDV5 using Objects365 dataset for the duplicate categories. The final model score on public and private leaderboard are 0.6269, 0.59459. We believe that the huge improvement is attributed to more data and more detailed annotation information.

Score of multi-scale test result is 0.64915 and 0.61821. The significant improvement compared to singe scale result is mainly credited with our merge strategy, which will be introduced later.

We merge all the model results based on ResNet200-vd backbone, and the final scores on public and private leaderboard are 0.65821 and 0.62874.

4. The Ensemble of Models

4.1. Top-k Voting-nms

Intuitively, the results of different models has different character, bounding box ensemble will be beneficial to the detection performance. For each model, since we filtered the bounding boxes based on soft-nms, however, it's helpless to reduce the duplicated boxes directly with same label in the same location. In this paper, Top-k voting-nms method is introduced based on voting mechanism. Based on the results of different models, we consider the average of the top-k bounding-boxes' scores and locations as the predicted score and location. Instead of keeping the highest one only in the original NMS, the ensemble box is obtained by the top-k bounding-boxes voting. By this way, the bounding boxes with same label are filtered by voting strategy. The IOU threshold is 0.5 in this step. Figure 4 shows the procedure voting strategy in this paper.

To illustrate the power of the top-k voting-nms, the multi-scale testing (ms testing) of the model trained from the OIDV5 and Objects365 dataset combine dataset (LB 0.6269/0.5946). Using the original nms, the mAP of the ms testing result is 0.5926 in private dataset which is not promoted. However, the top-k voting-nms with only refining
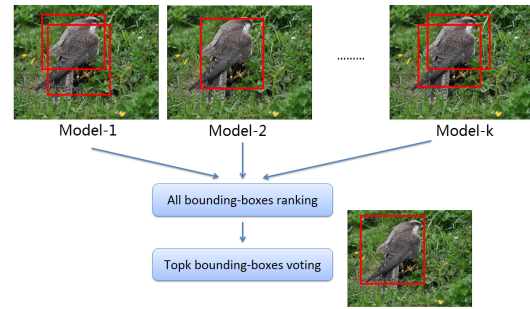

Figure 4. Topk bounding-box voting strategy

Method	Score
Soft-nms	0.6269/0.5946
Voting soft-nms	0.6271/0.5926
Top-k voting-nms(score)	0.6430/0.6131
Top-k voting-nms(score and location)	0.6492/0.6182

Table 3. Trained models for ensemble.

Model	Score
All models using ResNet200-vd(voting)	0.6582/0.6287
DPN107-FPN-Cascade-Dcnv2	0.6033/0.5757
ResNext101_32x16d_wsl-FPN-Dcnv2	0.6088/0.5861
SE154-Dcnv2 [7]	0.5882/0.5597
Inceptionv4-FPN	0.5819/0.5544
ResNext101_64x4d-GA-FPN-Dcnv2	0.6063/0.5677
GridRCNN-GN	0.5952/0.5670
Voting results	0.6816/0.6534

Table 4. Trained models for ensemble.

the score has a better result, the mAP is achieved to 0.6131. Further, when both refine score and location is adopted, the mAP is improved to 0.6492/0.6182.

4.2. The Result of Model Ensemble

Infactly, different model architectures has different perception on image. The models trained on OIDV5 dataset are ensembled to improve the performance. Specifically, the other models and scores are list as Table 4. Here, the models are just trained sixty thousand minibatches. All the models are trained with corresponding FPN pretrained models.

Besides classic methods, some novel object detection framework is adopted. In ResNext101_64x4d-GA-FPN-Dcnv2 [12], ResNext101_64x4d is used as backbone, guided anchoring [12] is adopted to produce region proposal. In GridRCNN [9], which is a novel object detection framework, to take the full advantage of the correlation of points in a grid, it adopts a grid guided localization mechanism for accurate object detection. We present detailed benchmarking results for some methods in Table 4,

The scores after merging all the above models are 0.6816 and 0.6534 on public and private leaderboard.

5. Conclusion

In this paper, we present an object detection framework based on PaddlePaddle, which performs well not only on COCO dataset but also on OIDV5. A new top-k voting method is proposed for model ensemble, which helps us win 2nd place in the Google AI Open Images Object Detection Track 2019.

References

[1] COCO - Common Objects in Context. http://cocodataset.org/.

[2] Objects365 Dataset. https://www.objects365.org/overview.html.

[3] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis. Improving object detection with one line of code. *CoRR*, abs/1704.04503, 2017.

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into High Quality Object Detection. *arXiv:1712.00726 [cs]*, Dec. 2017. arXiv: 1712.00726.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[6] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. *CoRR*, abs/1812.01187, 2018.

[7] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-Excitation Networks. *arXiv:1709.01507 [cs]*, Sept. 2017. arXiv: 1709.01507.

[8] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, 2016.

[9] Xin Lu, Buyu Li, Yuxin Yue, Quanquan Li, and Junjie Yan. Grid R-CNN. *CoRR*, abs/1811.12030, 2018.

[10] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra R-CNN: towards balanced learning for object detection. *CoRR*, abs/1904.02701, 2019.

[11] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[12] Jiaqi Wang, Kai Chen, Shuo Yang, Chen Change Loy, and Dahua Lin. Region proposal by guided anchoring. *CoRR*, abs/1901.03278, 2019.

[13] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. *CoRR*, abs/1711.07971, 2017.

[14] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. *CoRR*, abs/1811.11168, 2018.