

Approach for Large-Scale Hierarchical Object Detection

Yuan Gao, Xingyuan Bu, Yang Hu, Hui Shen, Ti Bai, Xubin Li and Shilei Wen
Department of Computer Vision Technology, Baidu
corresponding author: wenshilei@baidu.com

Abstract

*This report demonstrates our solution for the Open Images 2018 Challenge. Based on our detailed analysis, we find that despite the **largest existing dataset with object location annotations**, these exist extremely **data imbalance** and **label missing** in this dataset. To alleviate these problems, two simple but effective strategies are proposed, i.e., **data balance (DB)** and **hierarchical-non-maximum-suppression (HNMS)**, which could improve the absolute mean-averaged-precision (mAP) by 8.4% (DB) and 2% (HNMS). With a further ensemble strategy, the final mAP is boosted to 62.2% in the public leaderboard (ranked the 2nd place) and 58.6% in the private leaderboard (ranked the 3rd place, within 0.04 point compared to the 1st place).*

1. Introduction

To better understand the visual content, we should not only know *what* is the object, i.e., the so-called classification task, but also know *where* is the very object, i.e., the so-called location task. The object detection task is to simultaneously provide these two information for a given image.

Depending on the pipeline, most of the object detection techniques could be divided into two categories, i.e., one-stage method and two-stage method. Generally speaking, the one-stage methods focus on the speed performance while the dominant merit of the two-stage methods is the precision performance. In this challenge, we concentrate on the two-stage methods considering the outstanding precision performance.

Specifically, in the modern convolutional neural network (CNN) context, the regions with CNN features (RCNN) [2] method should be the earliest two-stage detector. Just as its name implies, the RCNN methods first output multiple region proposals using the selective-search algorithms, then regress the bounding-box (bbox) coordinates and classify into a specified class based on the extracted CNN features of the proposed region with the matured support vector ma-

chine (SVM) algorithm. To accelerate the pipeline, The SPPNet [4] is proposed by claiming that the feature maps could be shared by different proposals, and hence reducing the computation burden of the feature extraction process. Similar idea is used by the well-known fast RCNN [1] method. In this method, the features of the proposed regions are extracted by a newly-designed region-of-interest pooling (ROI-pooling) layer, and a multitask loss combined with the regression loss and the classification loss is considered for optimized training process. It should be noted that for all the above mentioned methods, the regions are proposed in an offline method such that they could not be optimized by the network. To solve this problem and therefore enable an end-to-end training style, a region proposal network (RPN) is incorporated into the overall pipeline, which shaped the well-known faster RCNN method [6]. It should be noted that the RPN is nearly cost-free considering the backbone-sharing property. Up to now, most of the improvements regarding the detection algorithms focus on the speed performance.

Another track about the improvements is the precision performance. As we know, the faster RCNN method uses the same feature maps to handle both the large and small objects, and consequently cannot adapt the object scales. To alleviate this drawback, the feature pyramid network (FPN) [5] is proposed to construct multiscale features with rich semantic information by designing a top-down architecture. On the other hand, to further use the available segmentation mask information, except for the classification and regression heads in the faster RCNN framework, an extra mask head is added in the mask-RCNN [3] method which results in the state-of-the-art algorithm performance.

The detection algorithms are pushing forward to faster and more precise by the talent researchers. However, the bbox annotations in the detection task are much more expensive compared to the label annotation in the classification task. As a result, the dataset scale for the detection task is still relatively small compared to that for the classification task, and therefore limit the performance of the detection task. To alleviate this problem, Google has open-sourced the Open Images datasets in the OpenImage chal-

lenge. Basically, 1.7 million images with bbox annotations are used in this challenge, containing 12 million instances ranging in 500 hierarchical categories. It should be noted that these exist severe data imbalances in this datasets. For example, there are 1.4 million and 14 instances corresponding to the *person* and *pressure cooker* classes, respectively. The large number of categories and the severe data imbalances induce the challenges in this dataset. Consider these two challenges, two simple tricks are adopted, i.e., data balance (DB) and hierarchical-non-maximum-suppression (HNMS), resulting absolute 8.4 points and 2 points improvement.

2. Methods and materials

2.1. Baseline

In this challenge, the faster RCNN framework is adopted as the baseline, where the backbone is the powerful SE-ResNeXt-154. To handle the objects in different scales, the FPN module is added in the stage4. Besides, the deformable ROI pooling layer is also utilized to further strengthen the performance of the baseline model. With these common tricks, our baseline model could achieve a mAP of 46.5%.

2.2. data balance

Figure 1 illustrates the data statistics for the MS-COCO datasets and the Open Images datasets, where the x-axis and the y-axis correspond to the label index and the log-transformed statistics of the datasets, respectively. Obviously, there exists severe data imbalance in the Open Images datasets compared to the MS-COCO datasets, as we mentioned in section 1. For example, the category of *person* has 1.4 million instances, which is 10^5 larger than that of the *pressure cooker* which has 14 instances. An intuitive solution to the data imbalance problem is to use the re-sampling strategy such that the images of different categories have the same probability to be sampled. Besides, by using the re-sampling strategy, the minor categories could be trained more sufficiently and hence accelerate the convergence.

2.3. Hierarchical non-maximum-suppression

In the Open Images challenges, if a label in the child node is assigned to an instance, it implies that all the labels in the parent nodes are also assigned. Consequently, during the evaluation, the model should recall all the labels in the parent nodes. However, due to the missing labels in the training datasets and the interclass competition, one could not output all the labels in the child and parent nodes for the same bounding box. To solve this problem, a hierarchical NMS strategy is proposed. In details, given all the bounding boxes of an image predicted by the model, we expand the associated single label of a single bounding box to multilabels with all the corresponded labels in the parent nodes,

where the score is same as that of the child node. Based on these expanded results, a classical NMS pipeline is applied, where the threshold of the intersection of the union (IOU) is set as 0.5. On the other hand, if the IOU of two bounding boxes belonging to the same category is larger enough, say 0.9, one should have more confidence regarding the existence of the object in this location. As a consequence, the bounding box with the highest score should be increased further based on the score of the dropped bounding box. In this challenge, a 30% score is voting to the bounding box with higher confidence.

3. Results

The datasets of the Open Images 2018 challenge contains 1.7 million images ranging in 500 categories, where 100K images are the official suggested validation dataset. In our custom settings, to accelerate the evaluation process and also enlarge the train datasets, we only use 5000 images as the mini-validation dataset, and the other as the train dataset. The initial learning rate is 0.01 and reduced to 0.001 after 40K iterations. The training process will be terminated after 50K iterations. The batch size is 48. We use the default multiscale training and testing strategies in the Detectron framework. Six Tesla-V100 GPU are utilized for training.

Figure 2 demonstrates the bar chart by adding different strategies. As can be seen, the data balance strategies could boost the performance heavily, i.e., 8.4 absolute points from the 46.5% baseline to the mAP of 54.9%. By further using the hierarchical strategy, the performance could be further improved by 2 absolute points, achieving the best single best model with a mAP of 56.9%. With a final ensemble strategy with 8 different models, we achieve the 62.2% mAP performance in the public leaderboard, ranking the 2nd place. Figure 3 illustrates a visual comparison among different single models, i.e., baseline, baseline+data balance, baseline+data balance+HNMS. For the majority category, such as the *Person* class, all the model show good results. However, for the minor category, such as the *Paddle* and the *Duck* classes, the baseline model produce inferior results with mis-located paddles and the missed ducks. The data balance strategy could greatly alleviate this problems as demonstrated in the middle of figure 3. With further hierarchical NMS, the label from the parent nodes could be correctly output as shown in the middle region of the right-most sub figure in figure 3.

4. Discussion

During the challenge, we have also tried some other tricks, some have minor improvement, some have negative effects. We would also like to present here for reference.

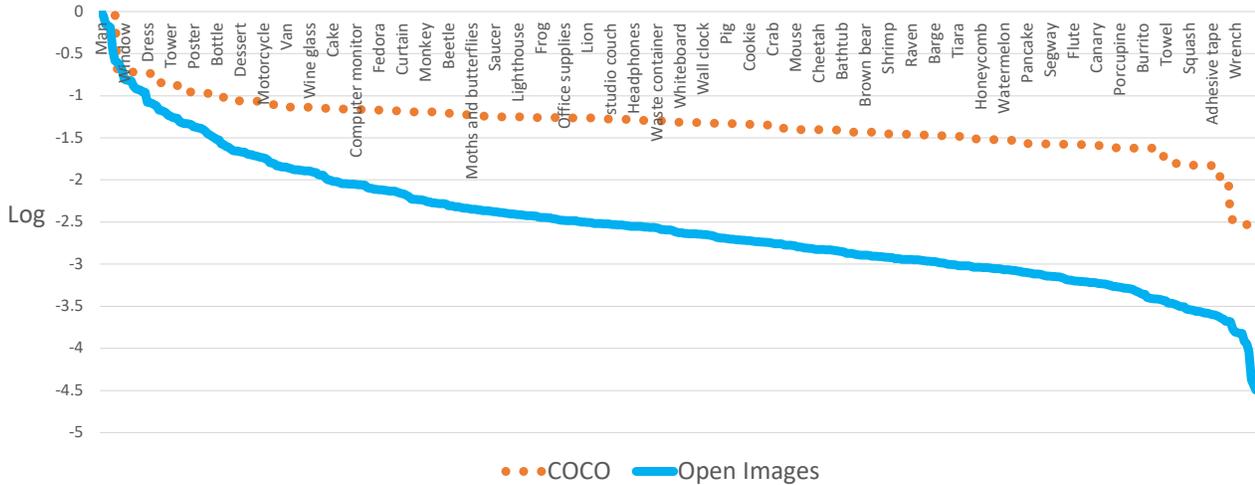


Figure 1. The data statistics of the Open Images and the MS-COCO datasets. The x-axis and the y-axis are with the label and the log-transformed instance counts, respectively. It should be noted that the label number of the Open Images and the MS-COCO datasets are different, which is 500 and 80, respectively. For better visualization, we have duplicated the statistics of the MS-COCO datasets by a mean value of 6.25 (some are duplicated by 6 times, some are 7).

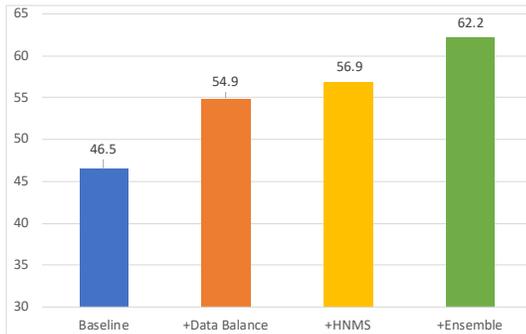


Figure 2. The trend for the improvement with different strategies. From left to right, the strategies are added step by step.

4.1. Ensemble

For the final results, we have used 8 models for ensemble: two different choices for the backbone (SE-ResNext-154 and ResNeXt-152), switch on/off for the deformable roi pooling and the data balance. These could improve the performance from our best single model with 56.9% to the final results with 62.2% in the public leaderboard.

4.2. Pretrained models

As a common step for the detection task, the backbone is usually trained from the ImageNet datasets. We also use the same strategy for the backbone training. Besides, in this challenge, based on the pretrained backbone, we have also tried to first train our model on the MS-COCO

datasets and then fine-tune on the Open Images datasets. Intuitively, this pipeline should be better than that of directly train on the Open Images datasets. However, we find no obvious performance improvement between these two strategies. A potential reason maybe that the Open Images is large enough (containing 1.7 million images) such that the relatively *small* MS-COCO dataset (containing 110K images) has minor effect.

4.3. OHEM

The online hard-example mining is a very popular strategy commonly used in the detection tasks. However, in this challenge, we find strong negative effect on the final results. Based on our analysis, the main reason should be that the sever label missing in the annotations of the training datasets. Qualitatively speaking, if one label is correctly predicted by our model, but the groundtruth doesn't contain this label, this correctly predicted label would be regarded as the hard-example and the optimizer will push it into the wrong direction, resulting negative performance. Therefore, this negative result teaches us the lesson that we should not perform OHEM tricks on the server label-missing datasets.

5. Conclusion

In this Open Image challenge, we find that there exists server data imbalance and label missing problem. We use the re-sampling strategy to tackle the data imbalance prob-

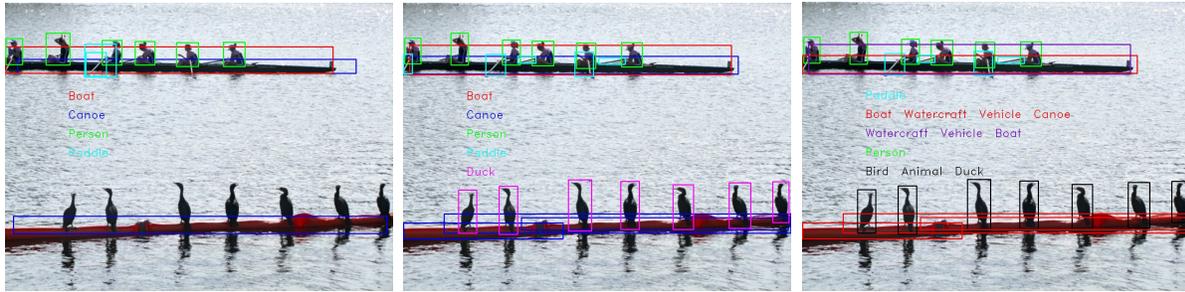


Figure 3. Visualization of our single model with different strategies. From left to right, the sub-figures correspond to the baseline model, baseline model with the data balance strategy, baseline model with the data balance and HNMS strategies. The words in the middle of the subfigures are the associated labels detected. It should be noted that the label in the parent-child node relationship shares the same color.

lem, which could improve to 54.9% compared to the baseline model with a mAP of 46.5%. To alleviate the label missing problem, a hierarchical NMS strategy is proposed, increasing the performance of the single model to 56.9% from the 54.9%. Finally, the ensemble strategy is adopted, boosting the mAP to 62.2%. Besides, we find that for the label missing datasets, the popular OHEM strategy should be avoided.

References

- [1] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1
- [2] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 1
- [3] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017. 1
- [4] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European conference on computer vision*, pages 346–361. Springer, 2014. 1
- [5] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 3, 2017. 1
- [6] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1