– Supplementary material –
TOOD: Task-aligned One-stage Object Detection

1. Implementation details

In this section, we describe the processes of network optimization and inference in more detail.

Optimization. Our implementations are based on the MMDetection toolbox [2] and Pytorch [7]. The models with backbone ResNet-50 are trained with 4 GPUs and a mini-batch of 4 per GPU, while the others are trained with 8 GPUs and a mini-batch of 2 per GPU. We use the Stochastic Gradient Descent (SGD) optimizer with a weight decay of 0.0001 and a momentum of 0.9. Unless specific, the models are trained for 12 epochs ($1\times$ learning schedule) and the initial learning rate is set to 0.01 and then reduced by a factor of 10 at the 8-th epoch and the 11-th epoch. The input images are resized to have a shorter side of 800 while the longer side is kept less than 1333. Specifically, if an anchor is assigned to the positive samples of more than one object, we only assign this anchor to the object with the minimal area. For the experiments compared with the state-of-the-art detectors, we train the models with scale jitter and for 24 epochs ($2\times$ learning schedule) as [6].

Inference. The inference phase is the same as that of ATSS [8]. Namely, we resize the input image in the same way as the training phase (*i.e.*, the shorter side is resized to 800 while the longer side is kept less than 1333), and then forward it through the detection network to obtain the predicted bounding boxes with a predicted class. Afterward, we use a confidence threshold of 0.05 to filter out the predictions with low confidence, and then select the top 1000 scoring boxes from each feature pyramid. Finally, we adopt the Non-Maximum Suppression (NMS) with the IoU threshold of 0.6 per class to generate the final top 100 confident predictions per image.

2. Discussion

Differences between TAL and previous works. As discussed, the proposed TAL is a learning-based approach for anchor selection and weighting. Here we discuss the differences between our TAL and several recent methods in terms of anchor selection and weighting. As mentioned in the paper, the adaptive methods can be divided into two categories: (1) positive/negative anchor collection such as FreeAnchor [9], MAL [3] and PAA [4]; (2) anchor weighting such as PISA [1], NoisyAnchor [5] and GFL [6] (*e.g.*, by modifying the loss functions). These methods adaptively perform either anchor collection or anchor weighting. We propose TAL that considers *both* aspects at the same time, allowing it to measure informative or high-quality anchors more accurately. Specifically, TAL is designed to dynamically collect the positive/negative anchors from a task-alignment point of view, and further weight the positive anchors carefully, according to the degree of task-alignment at each location. Compared with the current assignment methods such as ATSS [6] and PAA [4] which first select a set of candidate anchors based on the IoU score, and then analyze the distribution characteristics of the anchors to assign samples, the design of TAL is simpler yet more efficient by directly assigning the samples based on the proposed alignment metric. Particularly, recent GFL [6] attempted to align the tasks by replacing a binary classification label with an IoU score, on the basis of ATSS. TAL is different from the GFL, by using the proposed task-alignment metric to design both sample assignment and anchor weighting, which allows it to explicitly learn to refine both classification and localization in a coordinated fashion.

References

[1] Yuhang Cao, Kai Chen, Chen Change Loy, and Dahua Lin. Prime sample attention in object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11583–11591, 2020.

[2] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[3] Wei Ke, Tianliang Zhang, Zeyi Huang, Qixiang Ye, Jianzhuang Liu, and Dong Huang. Multiple anchor learning for visual object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10206–10215, 2020.

[4] Kang Kim and Hee Seok Lee. Probabilistic anchor assignment with iou prediction for object detection. In *Proceedings of the European Conference on Computer Vision*, 2020.

[5] Hengduo Li, Zuxuan Wu, Chen Zhu, Caiming Xiong, Richard Socher, and Larry S Davis. Learning from noisy anchors for one-stage object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10588–10597, 2020.

[6] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. In *Advances in Neural Information Processing Systems*, 2020.

[7] Adam Paskze and Soumith Chintala. Tensors and dynamic neural networks in python with strong gpu acceleration. `https://github.com/pytorch`, 2017.

[8] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9759–9768, 2020.

[9] Xiaosong Zhang, Fang Wan, Chang Liu, Rongrong Ji, and Qixiang Ye. Freeanchor: Learning to match anchors for visual object detection. In *Advances in Neural Information Processing Systems*, pages 147–155, 2019.