

Dually Supervised Feature Pyramid for Object Detection and Segmentation

Fan Yang¹ Cheng Lv² Yandong Guo² Longin Jan Latecki¹ Haibin Ling³

¹Department of Computer and Information Sciences, Temple University, Philadelphia, USA

²Xmotors.ai, Mountain View, CA, USA

³Department Computer Science, Stony Brook University, Stony Brook, NY, USA.

{fyang,latecki}@temple.edu, {luc,guoyd}@xiaopeng.com, hling@cs.stonybrook.edu

Abstract

Feature pyramid architecture has been broadly adopted in object detection and segmentation to deal with multi-scale problem. However, in this paper we show that the capacity of the architecture has not been fully explored due to the inadequate utilization of the supervision information. Such insufficient utilization is caused by the supervision signal degradation in back propagation. Thus inspired, we propose a dually supervised method, named dually supervised FPN (DSFPN), to enhance the supervision signal when training the feature pyramid network (FPN). In particular, DSFPN is constructed by attaching extra prediction (i.e., detection or segmentation) heads to the bottom-up subnet of FPN. Hence, the features can be optimized by the additional heads before being forwarded to subsequent networks. Further, the auxiliary heads can serve as a regularization term to facilitate the model training. In addition, to strengthen the capability of the detection heads in DSFPN for handling two inhomogeneous tasks, i.e., classification and regression, the originally shared hidden feature space is separated by decoupling classification and regression subnets. To demonstrate the generalizability, effectiveness, and efficiency of the proposed method, DSFPN is integrated into four representative detectors (Faster RCNN, Mask RCNN, Cascade RCNN, and Cascade Mask RCNN) and assessed on the MS COCO dataset. Promising precision improvement, state-of-the-art performance, and negligible additional computational cost are demonstrated through extensive experiments. Code will be provided.

1. Introduction

As one of the most fundamental problems in computer vision, object detection has been studied for decades. Following the rise of deep learning in computer vision, object detection has been greatly advanced in recent years. Numerous deep detectors are proposed and achieve promising results on the large-scale benchmark MS COCO [17]. In

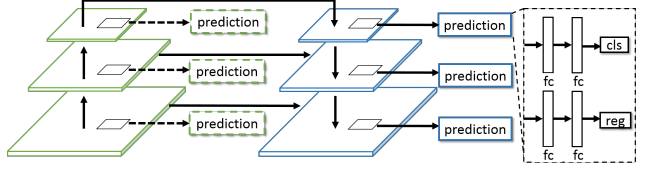


Figure 1: The diagram of the proposed dually supervised FPN [15] (DSFPN). The green and blue pyramid represent bottom-up and top-down subnets, respectively. The green dashed boxes are auxiliary prediction (i.e., detection or segmentation) heads. The black dashed box is the decoupled *cls-reg* heads in detection task. Note that the prediction heads with the same color share weights.

general, these deep detectors can be categorized into two types: one-stage model [18, 21, 22, 30, 12, 6, 29] and two-stage model [9, 8, 23, 5, 10, 2]. Compared with one-stage approaches, two-stage detection models usually consist of three components: *backbone network*, *region proposal head* and *detection head*. Given object proposals, a feature pyramid architecture (e.g., FPN [15]) is often adopted as the backbone network in the two-stage detectors. After that, a detection head is attached to the backbone in a lateral connect manner for achieving object localization and classification, as shown in the blue box in Fig. 1. As a natural extension, this framework has also been widely utilized for instance segmentation by adding segmentation heads.

Despite excellent performance obtained by this pyramid architecture, its potential has not been fully explored, owing to the insufficient utilization of supervision information. The reasons of the insufficient utilization can be summarized into two aspects: gradient degradation in feature pyramid and task inconsistency in detection head. In particular, the feature pyramid architecture is composited by two components: bottom-up subnet and top-down subnet; the prediction heads are mounted on top-down path (see Fig. 1). This may lead to the inadequate training of bottom-up subnet due to the presence of gradients degradation [11].

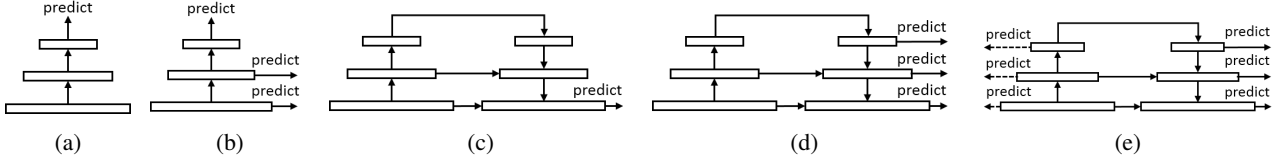


Figure 2: (a-c) Typical architectures in object classification, detection, and segmentation. (d) The proposed DSFPN.

Consequently, the top-down feature pyramid, which is composed of features from bottom-up counterpart, has shortage of representation capability. Further, the detection and segmentation performance are not saturated under feature pyramid architecture.

The task inconsistency is an inherent problem [4] in current structure of the detection head, which is constructed by two weights shared sibling subnets for classification and regression, respectively. This structure limits the ability of the head to utilize supervision information. Specifically, the sibling subnets deal with the two *inconsistent* tasks (classification and regression) with shared FC layers, resulting in sub-optimal solutions for each one [4]. Thus, in feature space the tangled supervision information in the two tasks cannot be effectively leveraged to facilitate the feature learning in the feature pyramid. As a consequence, the detection performance is seriously under-explored.

Addressing the aforementioned issues, we first propose a novel supervised learning strategy for feature pyramid network, called *dually supervised feature pyramid network* (DSFPN), which is embodied by mounting extra prediction heads (*i.e.*, detection or segmentation) on the bottom-up layers through lateral connection. This design strategy encourages the supervision signal to directly propagate to the bottom-up feature pyramid, so as to strengthen supervision information for the feature learning in it. Moreover, the prediction heads on the bottom-up part can serve as a regularization term to promote the model training. In the test phase, the auxiliary prediction heads are discarded, thus no extra computational cost is needed for model inference. In DSFPN, the *cls* and *reg* subnets (black dashed box of Fig. 1) are decoupled to enhance the capacity of detection head to handle the task inconsistency, *i.e.*, classification vs regression. Specifically, the decoupling increases the capacity of the heads such that the supervision information can be fully exploited to certain extent to strengthen the guidance of feature learning in the pyramid architecture.

The proposed *dual supervision* (DS) and *decouple* (DC) operations are general to a series of feature pyramid based detectors. The two operations are evaluated by incorporating into four representative FPN-based detectors (Faster RCNN [23], Mask RCNN [10], Cascade RCNN [2], and Cascade Mask RCNN [2]) with various backbone networks (ResNet-50/101 [11], ResNeXt-101 [27]) on the MS COCO

dataset [17].

In summary, the paper makes the following main contributions:

1. A novel DSFPN is proposed to solve the gradient degradation problem in FPN [15] for object detection and segmentation, so as to facilitate feature learning, promote model training. The task inconsistency in detection head is alleviated by decoupling the *cls* and *reg* subnets.
2. The proposed method consistently improves Faster RCNN [23], Mask RCNN [10], Cascade RCNN [2], and Cascade Mask RCNN [3] by a promising margin and achieves the state-of-the-art performance on COCO [17] with negligible extra computational cost.

The rest of the paper is organized as follows. Section 2 briefly reviews the relevant works. The proposed method is described in detail in Section 3. Section 4 presents the experimental results. Section 5 draws conclusions.

2. Related work

Object detection has been studied for decades, and numerous excellent works have been proposed. In this section, we will review the most relevant of them to highlight the differences of our work.

Two-stage methods. The two-stage methods divide the detection into two steps: object proposals and object detection. In general, candidate regions are first yielded; then a detection module classifies these regions and refines the region location. Fast [8]/Faster RCNN [23] mount a detection head on the top level of deep convolution networks (ConvNets) to take advantage of the rich feature representation for object detection (Fig. 2a). However, this manner neglects the hierarchical characteristics of the deep ConvNets. To utilize this characteristics, MS-CNN [1] proposes to predict object at multiple levels of deep ConvNets (Fig. 2b). Although this modification promotes the detectors ability to deal with multi-scales of objects, the low level features lack semantic information for object classification. FPN [15] significantly alleviates this problem by presenting a novel feature pyramid architecture, where low level features are strengthened by a top-down feature pyramid network and object prediction are conducted on each level via lateral

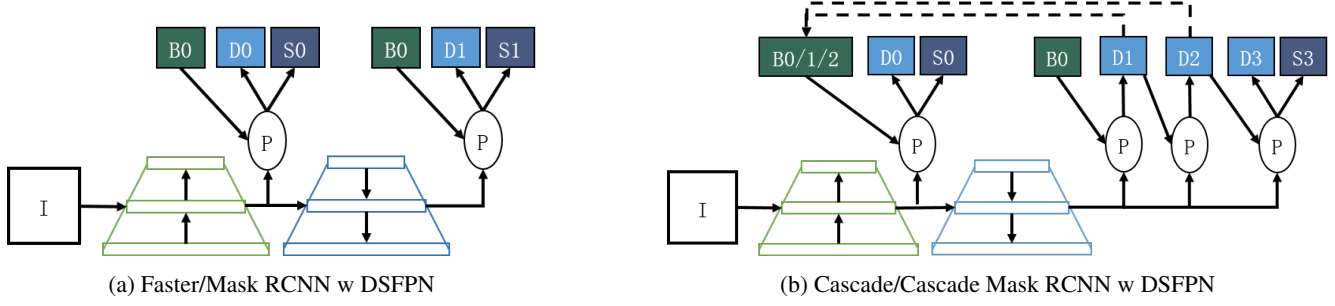


Figure 3: The illustration of Faster/Mask/Cascade/CascadeMask RCNN with dually supervised FPN [15] (DSFPN). “B”, “D”, and “S” indicate proposed *box*, *detection*, and *segmentation* heads, respectively. “P” is the short for RoI pooling operation. “I” represents the input image. Green and blue pyramid are bottom-up and top-down subnets. The heads mounted on bottom-up pyramid are auxiliary ones.

connection (Fig. 2d). Further, Mask RCNN [10] unifies object detection and segmentation into an end-to-end framework on the basis of FPN [15]. More recently, based on FPN [15], Cascade RCNN and Cascade Mask RCNN [3, 2] propose to yield high quality object detection and segmentation through multi-stage regression. Despite superior performance, the feature learning in bottom-up pyramid is not fully explored.

One-stage methods. In contrast to two-stage methods, one-stage methods directly perform object detection without proposals, e.g., SSD [18] and YOLO [21]. Same with Faster RCNN [23], YOLO [21] predicts objects on the top level of deep ConvNets. Thus, the hierarchy is ignored in detection. Similar with MS-CNN [1], SSD [18] assigns detection for different scales of objects on different levels of deep ConvNets, which separates the multi-scale objects in the feature space. DSSD [7] improves SSD [18] performance by introducing additional context via deconvolution. Further, RefineDet [30] boosts DSSD [7] using the bottom-up feature pyramid for proposal and the top-down counterpart for detection. However, the major performance gain is from the two-stage box regression but not the bottom-up feature pyramid learning. Moreover, the prediction on bottom-up pyramid is involved in both training and test phase.

Supervision in deep ConvNets. Usually, the feature learning in deep ConvNets is guided by a supervision signal from the output layer, e.g., VGG [25], ResNet [11] (Fig. 2a), and UNet [24] (Fig. 2c). This results in the gradient vanishing with the increasing of depth of layers, such that the network is hard to effectively train. While ResNet [11] alleviates the the problem via skip connection, it still remains a challenge to train deep ConvNets. DSN [13] solves this problem in general classification task by proposing a deeply-supervised training architecture, where auxiliary classifiers are introduced to hidden layers (Fig. 2b). PSPNet [31] demonstrates that adding an extra loss at the relatively high level layer of deep ConvNets can benefit the performance improve-

ment in scene parsing. The similar idea is also exploited in HED [28] for edge detection, but the prediction heads are attached on each level of deep ConvNets. Additionally, in HED [28] all the heads are needed in inference phase.

Difference from previous works. To highlight the differences of proposed dually supervised FPN (DSFPN) from previous related works, we summarize the typical network architecture for object classification, detection and segmentation in Fig. 2. For better illustration, the skip connection is omitted in the backbone network. By comparing with Fig. 2a and 2c, where the supervision signal is simply at the final output layer, DSFPN has multiple supervision signals on every layer of the network. Although in Fig. 2b the supervision signals are also on each layer, this structure is not suited for object detection or segmentation, due to the shortage of representation ability in lower layers. Note that SSD [18] adopts a similar architecture to Fig. 2b but uses relatively higher layers. In contrast to Fig. 2d, DSFPN (Fig. 2e) adds auxiliary supervision signals on inner layers, such that the network can be trained more effectively.

3. Method

In this section we elaborate the proposed *dually supervised FPN* (DSFPN) in details and present its formulation.

3.1. Overview

Generally speaking, DSFPN is proposed to deal with the gradient degradation in feature pyramid network for object detection or segmentation at region level during training phase. To this end, when proposals are given, auxiliary prediction heads (*i.e.*, object detection or segmentation heads) are attached to the bottom-up subnet (Fig. 3), such that supervision signal can immediately direct the feature learning in it. In the test phase, these auxiliary heads are discarded so that no extra computation is needed in model inference. Formally, given a set of N proposal boxes $B = \{b_i | i = 1, 2, \dots, N\}$, they are projected to a

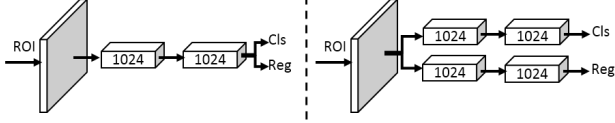


Figure 4: The structure of detection head in FPN [15] (left) and the proposed DSFPN (right).

level of feature layer according to the box size. Here we use the k -th level as an example, the boxes on the k -th level are B_k . The feature maps of the k -th level are denoted as f^k . Thus, the detection and segmentation heads on top-down pyramid can be formulated as $D_{td}^k(f_{td}^k, B_k)$ and $S_{td}^k(f_{td}^k, B_k)$, respectively. Accordingly, the prediction heads on bottom-up pyramid are $D_{bu}^k(f_{bu}^k, B_k)$ and $S_{bu}^k(f_{bu}^k, B_k)$. In the training phase, the predictions over the both pyramid subnets are $D = \{D_{td}^k, D_{bu}^k | k = 1, 2, \dots, M\}$ and $S = \{S_{td}^k, S_{bu}^k | k = 1, 2, \dots, M\}$, where M is the number of levels of pyramid. Here, ‘td’ and ‘bu’ indicate top-down and bottom-up, respectively. In the inference phase, the predictions are $D = \{D_{td}^k | k = 1, 2, \dots, M\}$ and $S = \{S_{td}^k | k = 1, 2, \dots, M\}$, since D_{bu}^k and S_{bu}^k are removed. The given proposal boxes are not necessary from proposal stage (e.g., RPN [23]), they can also come from latter stages. The effect of choosing the boxes from different stages will be discussed in Experiment. Since the object or segmentation heads over different levels share same weights, the k in each head notation is dropped for simplicity in subsequent sections.

3.2. Dual supervision

The proposed DSFPN is general to two- and multi-stage detectors, we describe the concrete architecture one by one. **Two-stage detector.** In two-stage detectors, we use two representative methods, Faster RCNN [23] and Mask RCNN [10], as examples to illustrate the process of incorporating dual supervision (DS). As shown in Fig. 3a, the proposal boxes, B_0 , are first given and projected to each level of pyramid. In Faster/Mask RCNN [23, 10], B_0 is yielded by RPN [23], which is not displayed in Fig. 3a. The detection and segmentation over both top-down and bottom-up pyramids are D_1, S_1, D_0 , and S_0 , which correspond to D_{td}, S_{td}, D_{bu} , and S_{bu} , respectively. The final loss of the detector can be written as

$$\mathcal{L}_{final} = \alpha_1 \mathcal{L}(D_{gt}, D_0) + \alpha_2 \mathcal{L}(D_{gt}, D_1) + \alpha_3 \mathcal{L}(S_{gt}, S_0) + \alpha_4 \mathcal{L}(S_{gt}, S_1) \quad (1)$$

where the D_{gt} and S_{gt} represent ground truth of detection and segmentation; $\mathcal{L}(D, \cdot)$ and $\mathcal{L}(S, \cdot)$ are the detection and segmentation loss functions; $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are the weights for each loss. The specific formulas of the two functions are

referred to [10]. The $\mathcal{L}(S, \cdot)$ can be omitted when segmentation ground truth is not available.

Multi-stage detector. Recently proposed Cascade and Cascade Mask RCNN [2, 3] are used to demonstrate how to integrate DS into multi-stage detectors (Fig. 3b). The proposed boxes can be produced from RPN [23], detection head at first stage (D_1) or second stage (D_2), which are denoted as B_0, B_1 , and B_2 , respectively. On top-down pyramid, the proposed boxes can only be B_0 . However, on bottom-up pyramid, the boxes can be one of the B_0, B_1 , and B_2 . Similar to two-stage detectors, the predictions over bottom-up pyramid are D_0 and S_0 . However, the predictions over top-down pyramid are separated into three sets: D_1, D_2 , and D_3 . As stated in Cascade Mask RCNN [3], the segmentation head is simply attached at stage three (denoted as S_3) for the trade-off between accuracy and computation complexity. Thus, the final loss of the detectors is written as

$$\mathcal{L}_{final} = \alpha_1 \mathcal{L}(D_{gt}, D_0) + \alpha_2 \mathcal{L}(S_{gt}, S_0) + \alpha_3 \mathcal{L}(S_3, S_3) + \sum_{i=1}^T \alpha_{si} \mathcal{L}(D_i, D_i) \quad (2)$$

where i is the index of the prediction stage on the top-down pyramid; T is the total number of stages; S_3 and S_3 are the segmentation predictions at stage three and its corresponding ground truth; D_i and $D_{i_{gt}}$ are detections and ground truth at stage i ; $\mathcal{L}(D, \cdot)$ and $\mathcal{L}(S, \cdot)$ are the detection and segmentation loss functions; $\alpha_1, \alpha_2, \alpha_3, \alpha_{si}$ are weights of each prediction head. Segmentation related terms only exist in Cascade Mask RCNN [3].

3.3. Decoupled head

For detection task, the architecture of the *cls-reg* decoupled head is shown in Fig. 4 right. Comparing with the original detection head structure (Fig. 4 left), the decoupled head separates the classification and regression tasks in hidden feature space, which is achieved by taking apart the shared two hidden layers with 1,024 nodes. Formally, the detection head is written as

$$\begin{aligned} D(B, f) &= C(B, f) + R(B, f) \\ C(B, f) &= W_c^2 \cdot W_c^1 \cdot f^B \\ R(B, f) &= W_r^2 \cdot W_r^1 \cdot f^B \end{aligned} \quad (3)$$

where C and R symbolize classification and regression, respectively; W_c and W_r represent the weights of hidden layers of classification and regression, respectively; ‘1’ and ‘2’ denote indices of hidden layers; f^B is the feature in box regions. This simple operation increases the capacity of detection head to address two inconsistent tasks. Moreover, the decoupling disentangles the propagation of supervision signal until the RoI pooling.

Table 1: The **ablation study** in representative **two-stage detectors**, *i.e.*, Faster RCNN and Mask RCNN. $1\times$ and $2\times$ indicate 12 epochs and 24 epochs of training schedule, respectively. \star indicates multi-scale testing.

Method	backbone	DS	DC	AP_{bbox}	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP_{mask}	AP_{50}	AP_{75}
Faster RCNN $_{1\times}$	ResNet50	\times	\times	36.7	58.4	39.6	21.1	39.9	48.1	-	-	-
		\checkmark	\times	37.7	59.0	40.6	20.6	40.6	40.6	-	-	-
		\times	\checkmark	37.5	58.8	40.8	21.1	40.7	49.2	-	-	-
		\checkmark	\checkmark	38.0	59.3	41.3	21.3	40.9	49.7	-	-	-
Faster RCNN $_{2\times}$	ResNet50	\times	\times	37.9	59.3	41.1	21.5	41.1	49.9	-	-	-
		\checkmark	\checkmark	39.2	60.1	42.7	22.2	42.3	52.1	-	-	-
Faster RCNN $_{1\times}$	ResNet101	\times	\times	39.4	61.2	43.4	22.6	42.9	51.4	-	-	-
		\checkmark	\checkmark	40.0	61.2	43.4	22.6	43.3	53.3	-	-	-
Faster RCNN $_{2\times}$	ResNet101	\times	\times	39.8	61.3	43.3	22.9	43.3	52.6	-	-	-
		\checkmark	\checkmark	41.1	62.2	44.3	23.2	44.4	54.8	-	-	-
Faster RCNN $_{1\times}$	ResNeXt101	\times	\times	41.5	63.8	44.9	24.8	45.4	53.5	-	-	-
		\checkmark	\checkmark	42.5	63.8	46.3	25.2	46.4	55.7	-	-	-
Faster RCNN $_{2\times}$	ResNeXt101	\times	\times	40.8	62.5	44.4	23.4	44.3	53.8	-	-	-
		\checkmark	\checkmark	42.6	63.5	46.4	24.7	46.5	56.0	-	-	-
Faster RCNN $_{2\times}\star$	ResNeXt101	\times	\times	44.0	65.0	47.9	28.7	46.9	57.4	-	-	-
		\checkmark	\checkmark	45.4	65.4	49.2	29.4	48.5	58.9	-	-	-
Mask RCNN $_{1\times}$	ResNet50	\times	\times	37.7	59.2	40.9	21.4	40.8	49.8	33.9	55.8	35.8
		\checkmark	\checkmark	38.8	59.6	42.3	21.5	41.6	51.7	34.6	56.0	37.0
Mask RCNN $_{2\times}$	ResNet50	\times	\times	38.6	59.8	42.1	22.2	41.5	50.8	34.5	56.5	36.3
		\checkmark	\checkmark	40.0	60.9	43.1	22.4	42.6	54.1	35.6	57.6	37.7
Mask RCNN $_{1\times}$	ResNet101	\times	\times	40.0	61.8	43.7	22.5	43.4	52.7	35.9	58.3	38.0
		\checkmark	\checkmark	40.7	61.5	44.6	22.5	43.8	54.5	36.3	58.3	38.5
Mask RCNN $_{2\times}$	ResNet101	\times	\times	40.9	61.9	44.8	23.5	44.2	53.9	36.4	58.3	38.7
		\checkmark	\checkmark	42.1	62.6	45.8	23.9	45.3	56.1	37.1	59.2	39.5
Mask RCNN $_{1\times}$	ResNeXt101	\times	\times	42.4	64.3	46.5	25.5	46.7	54.6	37.5	60.6	39.9
		\checkmark	\checkmark	42.8	63.5	46.7	25.0	46.5	56.3	37.6	60.1	39.9
Mask RCNN $_{2\times}$	ResNeXt101	\times	\times	42.3	63.6	45.9	25.4	45.9	55.7	37.2	59.9	39.4
		\checkmark	\checkmark	43.3	63.9	47.1	25.9	46.8	57.0	37.9	60.4	40.4

Table 2: The **ablation study** in representative **multi-stage detectors**, *i.e.*, Cascade RCNN and Cascade Mask RCNN. $1\times$ and $2\times$ indicate 12 epochs and 24 epochs of training schedule, respectively.

Method	backbone	DS	DC	AP_{bbox}	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP_{mask}	AP_{50}	AP_{75}
Cascade RCNN $_{1\times}$	ResNet50	\times	\times	40.9	59.0	44.6	22.5	43.6	55.3	-	-	-
		\checkmark	\checkmark	41.3	59.6	44.9	22.6	44.1	55.8	-	-	-
Cascade RCNN $_{2\times}$	ResNet50	\times	\times	41.3	59.7	44.9	23.8	44.0	55.3	-	-	-
		\checkmark	\checkmark	42.1	60.5	45.6	23.5	44.8	56.9	-	-	-
Cascade RCNN $_{1\times}$	ResNet101	\times	\times	42.8	61.4	46.8	24.1	45.8	57.4	-	-	-
		\checkmark	\checkmark	43.1	61.7	47.0	24.4	46.5	57.8	-	-	-
Cascade RCNN $_{2\times}$	ResNet101	\times	\times	43.0	61.4	46.9	23.9	46.1	58.0	-	-	-
		\checkmark	\checkmark	43.7	62.3	47.5	24.7	47.0	58.8	-	-	-
CascadeMask RCNN $_{1\times}$	ResNet50	\times	\times	41.3	59.6	44.9	23.1	44.2	55.4	35.4	56.2	37.8
		\checkmark	\checkmark	42.0	60.1	45.8	23.4	45.1	56.6	35.9	56.7	38.4
CascadeMask RCNN $_{1\times}$	ResNet101	\times	\times	43.3	61.7	47.2	24.2	46.3	58.2	37.1	58.6	39.8
		\checkmark	\checkmark	43.6	61.9	47.4	24.6	46.8	58.5	37.3	58.5	40.0
CascadeMask RCNN $_{2\times}$	ResNet101	\times	\times	43.5	62.1	47.2	24.3	46.9	58.2	37.2	58.7	39.5
		\checkmark	\checkmark	44.5	63.0	48.4	24.9	47.8	60.0	38.1	59.6	41.0

4. Experiments

4.1. Implementation details

We perform experiments on the MS COCO dataset [17] with 80 categories. The union of 80k train images and 35k images of val images (train2017) is used as train dataset. The 5k of val (val2017) is utilized to report results of ablation study. The final results are reported on

the test-dev dataset for fair comparison with state-of-the-art detectors. We implement our dually supervised FPN (DSFPN) based on the detetron¹ library and Caffe2² framework. By convention, we set the short side of an input image to 800 pixels and the other side is less than 1,333 pixels. The number of anchors and proposals are

¹<https://github.com/facebookresearch/Detector>

²<https://caffe2.ai/>

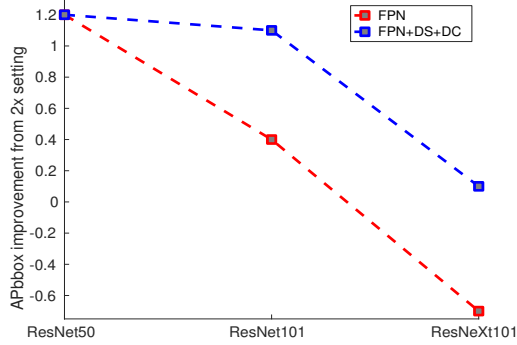


Figure 5: The effect of DS+DC on performance change from $2\times$ setting. The detector is Faster RCNN.

set as default. The RoIAlign in [10] is adopted in all experiments. The backbone network of all detectors includes FPN [15]. For Faster/Mask/Cascade RCNN based on ResNet50/ResNet101 [11], synchronized SGD is used to train the model on 8 P100 GPUs, each mini-batch involves 2 images per GPU. The other detectors are trained with 1 image per GPU, due to the limited GPU memory. No data augmentation is used except for horizontal flipping. Both $1\times$ (*i.e.*, 12 epochs) and $2\times$ (*i.e.*, 24 epochs) training schedules are utilized in experiments. The weights of each item of final loss are set to 1. In inference, all experiments are conducted on a single image scale ($800 \times 1,333$ pixels) without bells and whistles, unless otherwise noted.

4.2. Ablation study

We validate the effects of dual supervision (DS) and decouple (DC) operation on Faster/Mask RCNN [23, 10] and Cascade/CascadeMask RCNN [2, 3]. The generalizability of the proposed DSFPN is evaluated by comparing experimental results on the val2017 and test-dev datasets. The inference time is measured on a 1080Ti GPU with input size $800 \times 1,333$ pixels.

Effects of DS and DC on Faster/Mask RCNN. From Table 1, we note that DS and DC improve the baseline, Faster RCNN $_{1\times}$ with ResNet50 [11], by 1 and 0.8 points in AP_{bbox} , respectively. Combining them together, the improvement in AP_{bbox} reaches 1.3 points. By conducting experiments over various baseline, backbone, and training schedule, we find that the increment is consistent in terms of AP_{bbox} over different IoUs and scales. An interesting observation is that in baselines, the deeper the backbone, the less the effect of doubling training schedule ($2\times$) (Fig. 5). For example, when backbone is ResNet50 and detector is Faster RCNN, the $2\times$ setting increases the AP_{bbox} by 1.2 points; when backbone is ResNet101, the incremental margin declines to 0.4 points; when backbone is ResNeXt101, the incremental magnitude even becomes a negative value

Table 3: The effect of boxes in the bottom-up pyramid.

	backbone	stage	AP	AP ₅₀	AP ₇₅
Cascade RCNN $1\times$	ResNet50	0	41.3	59.6	44.9
Cascade RCNN $1\times$	ResNet50	1	41.0	59.2	44.3
Cascade RCNN $1\times$	ResNet50	2	40.9	59.0	44.1

(-0.7 points). This is because the deeper the network, the harder the training. In contrast, when the baselines are equipped with DS and DC, the $2\times$ setting consistently improves AP_{bbox} over all backbones. This can be attributed to the promotion of model training via enhanced supervision signal. In addition to detection, Mask RCNN with DS and DC also outperforms all the ones without the two operations in terms of AP_{mask} , and the effect on $2\times$ is apparent as well.

Effects of DS and DC on Cascade/CascadeMask RCNN.

As shown in Table 2, promising improvement of AP_{bbox} is gained in all baseline detectors by adding DS and DC. Similar to the two-stage detectors, when DS and DC are involved in model, detectors can benefit more from the double training schedule ($2\times$), especially for deeper backbones. Specifically, when backbone network is ResNet101 and detector is Cascade RCNN, $2\times$ helps the model with DS and DC to obtain a higher improvement margin than that without DS and DC (43.7 vs 43.1 and 43.0 vs 42.8). By comparing the performance in terms of AP_{50} and AP_{75} in detection task, we see that adding DS and DC can achieve larger increment in AP_{50} than that in AP_{75} . This can be attributed to the multi-stage regression yielding high-quality box, such that little room remains to explore in this direction. In the segmentation task, adding DP and DC can still lead to performance improvement, even if the backbone is ResNet101. **Boxes generated from stage 0, 1, and 2.** The experimental results with boxes from different stages are listed in Table 3. We note that (1) using boxes from stage 0 (*i.e.*, RPN [23]) achieves best performance; and (2) the higher the stage, the lower the performance. It suggests that when the boxes are coarse (*i.e.*, boxes from stage 0), the auxiliary detection heads can better assist the model training. The reason may be that the coarse boxes contain a diversity of region samples, so that the auxiliary heads can facilitate the model training by taking advantage of the diversity.

Generalizability of DSFPN. To demonstrate the generalizability of the proposed DSFPN, we conduct extensive experiments on val2017 and test-dev datasets. The experimental results are listed in Table 4. By comparing the results in val2017 and test-dev, we note that in two-stage methods, the performance increment gained by adding DSFPN is present in all backbones; moreover, the improvement is consistent across the val2017 and test-dev. This demonstrates that the proposed DSFPN promotes detection performance not by over-fitting to a validation data, it can gener-

Table 4: Generalizability of the proposed DSFPN.

method	backbone	DSFPN	test speed	val2017						test-dev					
				AP_{bbox}	AP_{50}	AP_{75}	AP_{mask}	AP_{50}	AP_{75}	AP_{bbox}	AP_{50}	AP_{75}	AP_{mask}	AP_{50}	AP_{75}
FasterRCNN _{2x}	ResNet50	\times	0.074s	37.9	59.3	41.1	-	-	-	37.9	59.6	41.0	-	-	-
		\checkmark	0.075s	39.2	60.1	42.7	-	-	-	39.3	60.5	42.8	-	-	-
FasterRCNN _{2x}	ResNet101	\times	0.097s	39.8	61.3	43.3	-	-	-	40.0	61.7	43.5	-	-	-
		\checkmark	0.098s	41.1	62.2	44.3	-	-	-	41.3	62.5	44.5	-	-	-
FasterRCNN _{2x}	ResNeXt101	\times	0.202s	40.8	62.5	44.4	-	-	-	41.1	63.1	44.6	-	-	-
		\checkmark	0.203s	42.6	63.5	46.4	-	-	-	43.0	64.4	46.7	-	-	-
MaskRCNN _{2x}	ResNet50	\times	0.082s	38.6	59.8	42.1	34.5	56.5	36.3	39.0	60.5	42.6	34.9	57.2	36.9
		\checkmark	0.084s	40.0	60.9	43.1	35.6	57.6	37.7	40.2	61.1	43.7	35.7	57.8	37.8
MaskRCNN _{2x}	ResNet101	\times	0.104s	40.9	61.9	44.8	36.4	58.5	38.7	41.0	62.4	44.8	36.4	59.1	38.6
		\checkmark	0.105s	42.1	62.6	45.8	37.1	59.2	39.5	42.3	63.0	45.9	37.3	59.8	39.6
MaskRCNN _{2x}	ResNeXt101	\times	0.210s	42.3	63.6	45.9	37.2	59.9	39.4	42.6	64.1	46.5	37.6	60.8	39.9
		\checkmark	0.213s	43.3	63.9	47.1	37.9	60.4	40.4	43.8	64.7	47.9	38.4	61.5	40.9
CascadeRCNN _{2x}	ResNet50	\times	0.101s	41.3	59.7	44.9	-	-	-	41.5	59.9	45.2	-	-	-
		\checkmark	0.115s	42.1	60.8	45.6	-	-	-	42.1	60.8	45.7	-	-	-
CascadeRCNN _{2x}	ResNet101	\times	0.123s	43.0	61.4	46.9	-	-	-	43.0	61.3	47.0	-	-	-
		\checkmark	0.137s	43.7	62.3	47.5	-	-	-	43.7	62.4	47.4	-	-	-
CascadeMaskRCNN _{2x}	ResNet101	\times	0.131s	43.5	62.1	47.2	37.2	58.7	39.5	43.7	62.3	47.6	37.4	59.4	40.0
		\checkmark	0.147s	44.5	63.0	48.4	38.1	59.6	41.0	44.5	63.0	48.3	38.0	60.0	40.8

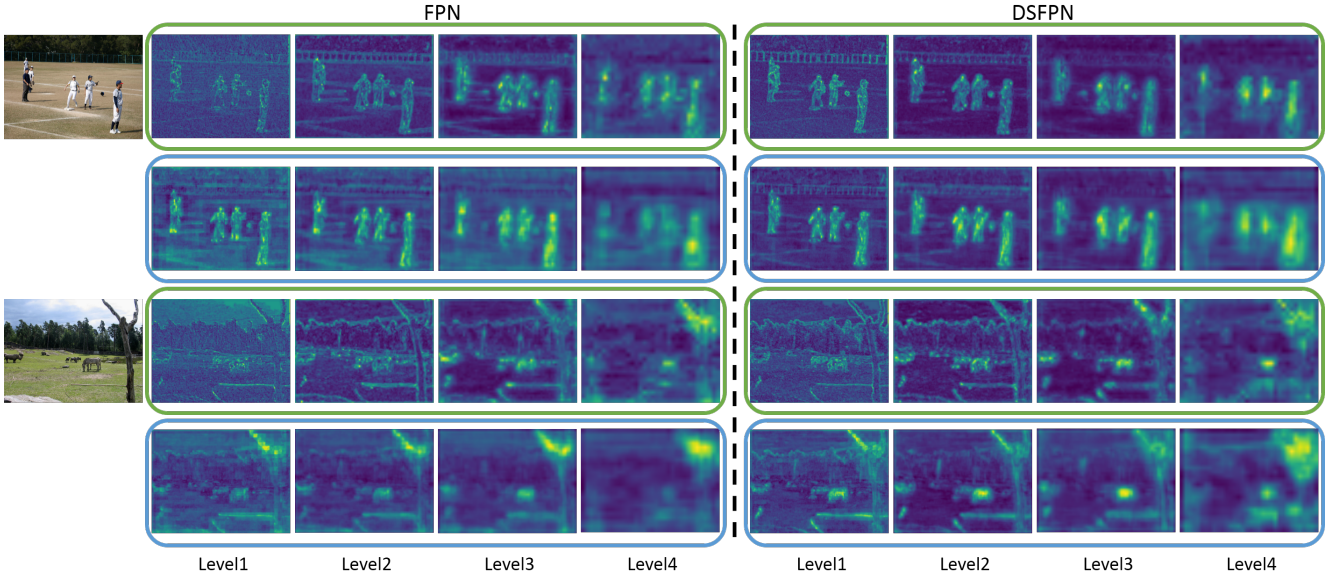


Figure 6: The feature map of each layers in FPN [15] (left) and DSFPN (right). The green and blue boxes indicate that the feature maps enclosed are from bottom-up and top-down pyramids, respectively.

alize to test dataset. In multi-stage methods, the consistent improvement also exists, but with subtle less margin. Apart from detection task, in the instance segmentation, both two and multi-stage methods with DSFPN surpass that without DSFPN. The aforementioned observations illustrate that the proposed DSFPN is general and robust to datasets and tasks.

Inference time. From Table 4, we see that in two-stage methods, with or without DSFPN, the inference speed is almost the same; in multi-stage methods, the speed has a trivial difference. The subtle change in speed demonstrates that our DSFPN strengthens detectors precision with little extra computational cost.

Visualization of feature maps. To verify that DSFPN does facilitate the feature learning in bottom-up pyramid and en-

hances the feature representation capability of top-down pyramid, we visualize the feature maps from each level of the pyramids in Fig. 6. The displayed feature maps are the summation of the raw feature maps over channel at each pyramid level. We can note that the feature maps from each level of bottom-up pyramid of DSFPN (green boxes in right of Fig. 6) are cleaner and with less noise than that from FPN [15] (green boxes left of Fig. 6). Accordingly, the feature maps from top-down pyramid (blue boxes in right of Fig. 6) also have stronger activation, comparing with the counterpart (blue boxes in left of Fig. 6). Especially, at lower levels, the feature maps from DSFPN are significantly superior to that from FPN [15]. These observations experimentally demonstrate that the proposed DSFPN guides fea-

Table 5: Comparison with state-of-the-art methods on the COCO [17] test-dev dataset.

method	backbone	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
YOLOv2 [22]	DarkNet-19	21.6	44.0	19.2	5.0	22.4	35.5
SSD-513 [18]	ResNet101	31.2	50.4	33.3	10.2	34.5	49.8
DSSD-513 [7]	ResNet101	33.2	53.3	35.2	13.0	35.4	51.1
RefineDet512 [30]	ResNet101	36.4	57.5	39.5	16.6	39.9	51.4
RetinaNet800 [16]	ResNet101	39.1	59.1	42.3	21.8	42.7	50.2
CornerNet [12]	Hourglass-104	40.5	56.5	43.1	19.4	42.7	53.9
FASF800 [32]	ResNeXt101	42.9	63.8	46.3	26.6	46.2	52.7
Faster RCNN+++ [11]	ResNet101	34.9	55.7	37.4	15.6	38.7	50.9
Faster RCNN w FPN [15]	ResNet101	36.2	59.1	39.0	18.2	39.0	48.2
Mask RCNN [10]	ResNeXt101	39.8	59.8	-	22.1	43.2	51.2
Grid RCNN [19] w FPN	ResNet101	41.5	60.9	44.5	23.3	44.9	53.1
Grid RCNN [19]w FPN	ResNeXt101	43.2	63.0	46.6	25.1	46.5	55.2
Libra RCNN [20] w FPN	ResNet101	41.1	62.1	44.7	23.4	43.7	52.5
Libra RCNN [20] w FPN	ResNeXt101	43.0	64.0	47.0	25.3	45.6	54.6
Cascade RCNN [2] w FPN	ResNet101	42.8	62.1	46.3	23.7	45.5	55.2
TridentNet [14]	ResNet101	42.7	63.6	46.5	23.9	46.6	56.6
Cascade RCNN w Rank-NMS [26]	ResNet101	43.2	61.8	47.0	24.6	46.2	55.4
Faster RCNN w DSFPN (ours)	ResNeXt101	43.0	64.4	46.7	25.3	46.1	54.7
Mask RCNN w DSFPN (ours)	ResNet101	42.3	63.0	45.9	23.4	44.9	54.6
Mask RCNN w DSFPN (ours)	ResNeXt101	43.8	64.7	47.9	25.5	46.8	55.7
Cascade RCNN w DSFPN (ours)	ResNet101	43.7	62.4	47.4	24.3	45.9	56.8
Cascade Mask RCNN w DSFPN (ours)	ResNet101	44.5	63.0	48.3	24.3	47.2	58.5

ture learning more effectively than FPN [15] does.

4.3. Comparison with state-of-the-arts

To demonstrate the effectiveness of the proposed DSFPN, it is compared with representative and recently proposed methods on the COCO test-dev dataset. The results are shown in Table 5. In the ResNet101 setting, the proposed methods outperform all other one-, two-, or multi-stage methods. In multi-stage methods, the proposed method surpasses other methods by a large margin. These experimental results clearly show the advantage of the proposed method.

4.4. Learning curves of FPN and DSFPN

To study the effectiveness of the proposed dually supervised (DS) training strategy in facilitating model learning, we leverage Mask RCNN with ResNet50 as detector and compute the AP_{bbox} over training data and validation data (val2017) with interval 5,000 iterations in $1\times$ training setting. Due to the large volume of training data, we compute the metric over randomly sampled $10k$ images from entire training images. From Fig. 7, we note that the detector with DSFPN converges faster than that with FPN in training data (red solid line vs blue solid line). This can be attributed to DS training manner. Specifically, DS introduces auxiliary prediction heads to assist the final prediction heads via directly supervising bottom-up subnet. Therefore, the detector with DSFPN can converge more efficiently. Moreover, the detector with DSFPN can converge to a higher AP_{bbox} point. This demonstrates that DS not only promotes detector training efficiently but also effectively.

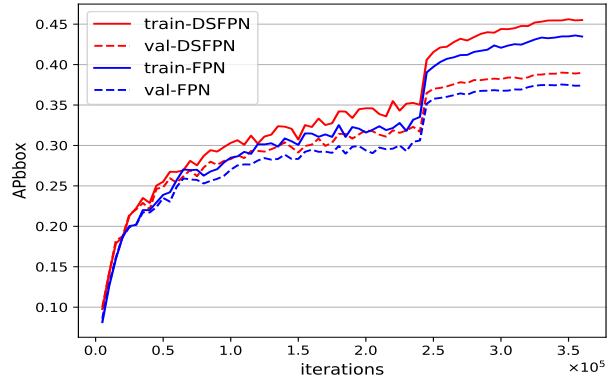


Figure 7: Learning curves of detectors with FPN (blue) and DSFPN (red). The detector is Mask RCNN with ResNet50, trained in $1\times$ setting.

5. Conclusion

In this work, we propose a novel dually supervised training strategy, named DSFPN, to improve FPN [15] by dealing with the gradient exploding or vanishing problem. DSFPN is generally effective for object detection and segmentation tasks, and it needs little extra computation during detection/segmentation inference. In addition, a simple but effective *cls-reg* decoupled head is proposed for object detection in DSFPN to alleviate task inconsistency. To demonstrate the superiority, generalizability, and efficiency of the proposed DSFPN, we integrate DSFPN into Faster/Mask/Cascade/CascadeMask RCNN, and promising results are observed in the extensive validation experiments on the COCO dataset.

References

- [1] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *European conference on computer vision*, pages 354–370. Springer, 2016. 2, 3
- [2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 1, 2, 3, 4, 6, 8
- [3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *arXiv preprint arXiv:1906.09756*, 2019. 2, 3, 4, 6
- [4] Bowen Cheng, Yunchao Wei, Honghui Shi, Rogerio Feris, Jinjun Xiong, and Thomas Huang. Revisiting rcnn: On awakening the classification power of faster rcnn. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 453–468, 2018. 2
- [5] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016. 1
- [6] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Object detection with keypoint triplets. *arXiv preprint arXiv:1904.08189*, 2019. 1
- [7] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Amrith Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017. 3, 8
- [8] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1, 2
- [9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra 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
- [10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1, 2, 3, 4, 6, 8
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 2, 3, 6, 8
- [12] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018. 1, 8
- [13] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Artificial intelligence and statistics*, pages 562–570, 2015. 3
- [14] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. *arXiv preprint arXiv:1901.01892*, 2019. 8
- [15] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 1, 2, 3, 4, 6, 7, 8
- [16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 8
- [17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 2, 5, 8
- [18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1, 3, 8
- [19] Xin Lu, Buyu Li, Yuxin Yue, Quanquan Li, and Junjie Yan. Grid r-cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7363–7372, 2019. 8
- [20] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 821–830, 2019. 8
- [21] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1, 3
- [22] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 1, 8
- [23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian 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, 2, 3, 4, 6
- [24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3
- [25] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3
- [26] Zhiyu Tan, Xuecheng Nie, Qi Qian, Nan Li, and Hao Li. Learning to rank proposals for object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 8
- [27] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 2
- [28] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015. 3

- [29] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. *arXiv preprint arXiv:1904.11490*, 2019. 1
- [30] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Single-shot refinement neural network for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4203–4212, 2018. 1, 3, 8
- [31] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 3
- [32] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. *arXiv preprint arXiv:1903.00621*, 2019. 8