Deformable Part Region Learning for Object Detection

Seung-Hwan Bae

Vision and Learning Laboratory, Inha University, Korea
shbae@inha.ac.kr

Abstract

In a convolutional object detector, the detection accuracy can be degraded often due to the low feature discriminability caused by geometric variation or transformation of an object. In this paper, we propose a deformable part region learning in order to allow decomposed part regions to be deformable according to geometric transformation of an object. To this end, we introduce trainable geometric parameters for the location of each part model. Because the ground truth of the part models is not available, we design classification and mask losses for part models, and learn the geometric parameters by minimizing an integral loss including those part losses. As a result, we can train a deformable part region network without extra super-vision, and make each part model deformable according to object scale variation. Furthermore, for improving cascade object detection and instance segmentation, we present a Cascade deformable part region architecture which can refine whole and part detections iteratively in the cascade manner. Without bells and whistles, our implementation of a Cascade deformable part region detector achieves better detection and segmentation mAPs on COCO and VOC datasets, compared to the recent cascade and other state-of-the-art detectors.

Introduction

The goal of object detection or instance segmentation is to determine meaningful object locations with boxes or masks from an image. For accurate and fast detection, the progress has been made by developing powerful features (He et al. 2016) and effective detection architecture (Ren et al. 2015; Redmon et al. 2016; Liu et al. 2016) over the last decades. Among them, convolutional object detectors (Bolya et al. 2019; Fu, Shvets, and Berg 2019; Tian et al. 2019; Zhao et al. 2019) have shown significant improvements.

However, these convolutional detectors are still limited in handling the large geometric transformations and variations caused by object pose, scale, viewpoints, and part deformation. The main reason is that the most CNNs used for feature extraction have fixed structures of CNN modules (*i.e.* convolution, pooling, and RoI pooling layers) as discussed in (Dai et al. 2017). As a result, it is difficult to detect non-rigid objects and objects with different scales or poses by using the convolutional detectors. In order to improve the robustness to the geometric transformation, we propose a deformable part region network (DPR-Net) that makes decomposed part models deformable adaptively according to object scales or poses. Our DPR-Net introduces a deformable part generation module, and it contains trainable geometric parameters to transform spatial locations and scales of each part region according to object scale variation. This differs with other deformable detection methods (Girshick et al. 2015; Wan, Eigen, and Fergus 2015; Dai et al. 2017) which learn parameters to adjust spatial locations of parts. In addition, we present a weakly supervised learning to train DPR-Net because the ground truth of part models is unavailable. To this end, we evaluate classification and segmentation losses of part models by comparing scores and masks predicted from fused part features with their counterpart ground truth. By minimizing an integral loss including these part losses, the gradients of the loss can affect the transformation of part models more directly. It also makes our DPR-Net can be end-to-end trainable without extra supervision.

In order to improve the quality of detected boxes and masks progressively, we present multi-stage refinement via the deformable part model learning based on a Cascade architecture. Similar to (Cai and Vasconcelos 2018; Chen et al. 2019; Zhang et al. 2019), our Cascade deformable part region detector (Cascade D-PRD) consists of a series of consecutive detection header to predict classes or locations. However, our Cascade D-PRD can improve those qualities of whole and part models together by feeding the refined boxes at previous step to DPR-Net. Subsequently, it can improve part boxes by re-decomposing the refined boxes. Then, Cascade D-RPD can produce stronger semantic RoI features with the refined part and whole RoIs, and exploit them for training consecutive box and mask headers.

To sum up, the main contributions of this paper can be summarized as follows: (i) proposition of the DPR-Net for transforming object part regions against object geometric variation (ii) proposition of the weakly supervised object detection and segmentation to learn part models without extra supervision (iii) proposition of a new cascade scheme for refining object and part predictions progressively.

Our single D-PRD achieves the state-of-the-art results without employing other performance improvement methods on MSCOCO19. We also make extensive implementation of D-PRDs with various feature extractors and provide

thorough ablation study to prove the effectiveness of the D-PRD. Moreover, our Cascade D-PRD can boost mAP by 3% to D-PRD. Without bells and whistles, our Cascade D-PRD achieves impressive 49.2 box and 42.4 mask APs. We improve the box and mask APs by 2.1 and 1.2 points compared to the recent HTC (Chen et al. 2019) with the same backbones. Another benefit is that our DPR-Net can be applied easily for the existing anchor-based detectors with simple modification.

Related Works

The recent convolutional detectors can be categorized into two- (He et al. 2017; Bae 2019; Huang et al. 2019) and one-stage (Lin et al. 2017b; Bolya et al. 2019; Tan, Pang, and Le 2020) detectors in terms of the usage of region proposal network (RPN). In general, the former detectors shows the better accuracy but lower speed due to the extra stage by RPN. For reducing the complexity further, anchor-free detectors (Law and Deng 2018; Lu et al. 2019; Tian et al. 2019) find top peaks within a key point heatmap per class, and consider the peaks as center positions of objects. In addition, there are many efforts to improve learning of the multi-scale feature maps (Zhao et al. 2019; Ghiasi, Lin, and Le 2019; Tan, Pang, and Le 2020) by adding new pathways and modules. Multi-stage detectors based on a cascade architecture (Cai and Vasconcelos 2018; Chen et al. 2019; Vu et al. 2019) are designed for high-quality detection. These detectors enhance the quality of boxes or masks by refining the predictions progressively with a series of detection headers.

Because the recent detectors mentioned above are based on CNNs with fixed geometric structures, they are inherently limited to model geometric transformation. For the robustness to the geometric transformation, deformable part detectors (Girshick et al. 2015; Wan, Eigen, and Fergus 2015) based on CNN have been presented. In addition, a spatial transformation network (STN) (Jaderberg et al. 2015) is presented to learn affine transformation parameters within a CNN for a given image. To reduce the model capacity of the STN, the inverse STN (Lin and Lucey 2017) propagates warped parameters instead of warped images. In deformable CNN (Dai et al. 2017), spatial offsets are augmented and learned to adjust spatial locations of convolution filtering and RoI pooling. To generate different shapes of default boxes, anchor learning (Yang et al. 2018; Ke et al. 2020) is presented.

Inspired by these recent works, we present D-PRD in order to accommodate geometric transformations of decomposed part models. It can also learn regression parameters of part models from end-to-end learning. Compared to (Girshick et al. 2015; Wan, Eigen, and Fergus 2015; Dai et al. 2017), our D-PRD can adjust spatial sizes as well as spatial locations of part models. In an attempt to improve detection and segmentation accuracy more, we further design a Cascade D-PRD architecture for refining detected boxes and masks using whole and part models based on a series of D-PRD heads.

Weakly supervised object localization (Zhang et al. 2018b; Zhong et al. 2021) and detection (Zeng et al. 2019; Zhong et al. 2020) localize single or multiple objects with image category labels. However, our weak supervised learning is to detect part boxes with category labels only.

Deformable Part Region Detector

For improving the robustness of convolutional detectors over geometric transformations, we propose a deformable part region detector (D-PRD), and provide the whole architecture of the D-PRD in Fig. 1. We introduce a deformable part regression layer to transform the spatial locations and sizes of decomposed part boxes. In addition, classification and segmentation losses for part models are designed to minimize the difference between the predicted outputs (*i.e.* class labels and masks) of a combined feature from part regions and their ground truth. Since the output features of part models are affected by transformation parameters of the deformable part layer, the parameters of the decomposed part models are learned by reducing these losses.

Preliminaries of D-PRD

We use region decomposition assembly network (RDA-Net) (Bae 2019) for fusing different region features. Let $\mathbf{d} = (x, y, w, h)$ be a bounding box, where x, y, w and h are the center positions, width and height. Then, smaller decomposed regions $\{\mathbf{d}^p | p \in \{\text{left, right, bottom, upper}\}\}$ can be generated by dividing $\mathbf{d}$ into several part regions (as shown in Fig. 2). In order to generate strong semantic features, RDA-Net performs multi-stage refinement for RoI features of different regions. At stage $l(> 1)$, features of different regions are compared, and a stronger semantics feature $\mathbf{x}_l$ is kept only. In specific, after the RoIAlign (He et al. 2017), we can extract a warped feature $\mathbf{x}^{whole}$ for $\mathbf{d}$ at the corresponding scale (*or* pyramid) level of multi-scale pyramid features (*e.g.* FPN (Lin et al. 2017a), PANet (Liu et al. 2018), and BiFPN (Tan, Pang, and Le 2020)) in consideration of their box sizes. Similarly, we can extract warped part feature maps $\{\mathbf{x}_1^p | p \in \{\text{left, right, bottom, upper}\}\}$ of the same size for each part box $\{\mathbf{d}^p | p \in \{\text{left, right, bottom, upper}\}\}$.

At each l stage, the bi-directional features $\mathbf{x}_l^p$ and $\mathbf{x}_l^q$ of different regions are merged to produce the stronger feature $\mathbf{x}_l^r$ using the region assembly block (RAB) consisting of four consecutive 3x3 conv filters, ReLU functions, and one element-wise max unit over each channel as shown in Fig. 1. As a result, a combined feature $\mathbf{x}_l^r$ is the output of the RAB for the input features $\mathbf{x}_l^p$ and $\mathbf{x}_l^q$ of as follows:

$$\mathbf{x}_l^r = \text{RAB}\left(\mathbf{x}_l^p, \mathbf{x}_l^q\right) \tag{1}$$

p (*or* q) represents each part (*i.e.* left, right, bottom, upper). r means combined parts (*i.e.* left-right(l/r), bottom-upper (b/u) and comb). On the other hand, $\mathbf{x}^{whole}$ is passed through several conv layers, and is also compared with the $\mathbf{x}_3^{comb}$ to produce $\mathbf{x}_4^{comb}$. Then, the last refined feature $\mathbf{x}_4^{comb}$ is used as an input of box and mask heads. The resolutions of $\mathbf{x}^{whole}$, $\mathbf{x}_l^p$, and $\mathbf{x}_l^r$ are the same sizes at all the stages.

Deformable Part Region Network (DPR-Net)

Given a feature map of size $H \times W$, we apply k_A anchors (*or* reference boxes) per location. Each anchor $\mathbf{d} = (x, y, w, h)$ can be decomposed into k_P parts ($k_P = 4$). Therefore, there

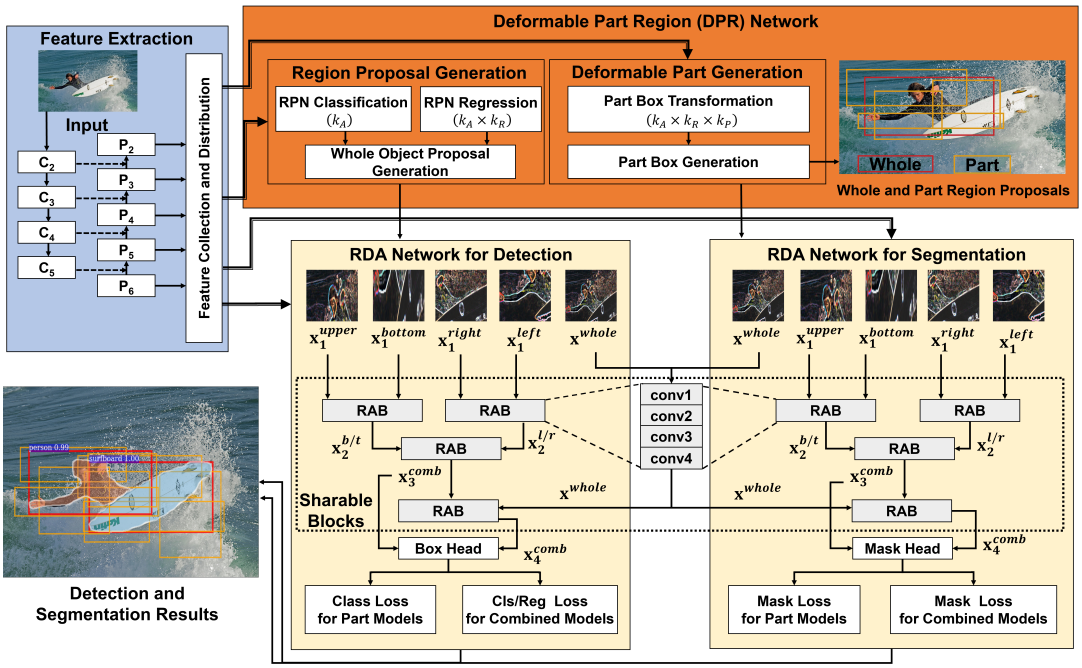

Figure 1: Proposed D-PRD: As a feature extractor, the FPN is used. In the DPR-Net, k_p part boxes for each proposal are transformed by the outputs of the part box transformation layer. In each RDA-Net, we extract the features $\mathbf{x}_3^{comb}$ refined with part model features only and $\mathbf{x}_4^{comb}$ refined with part and whole object features. For training D-PRD, we minimize the integral loss Eq. (3) including part losses. Note that all the trainable parameters of RDA networks for detection and segmentation are shared for reducing the model complexity.

exist HWk_A anchors and HWk_Ak_P part boxes in total. We assume that the k_P part boxes of the anchor are also removed when an anchor is removed by the NMS or score thresholding. This assumption avoids the additional training for predicting classification scores (*or* objectness score) per part box, and improves detection speed. In addition, we assume that each part box can be transformed independently. In this case, we need $k_R = 4$ parameters to encode coordinates of a part box. Therefore, the part box transformation layer has $k_A \times k_R \times k_P$ outputs in order to transform k_R coordinates of $k_A \times k_P$ part boxes for each location.

We use the $k_A \times k_R \times k_P$ outputs as offsets of coordinates of $k_A \times k_P$ part boxes. From the lowest to highest, they correspond to (x, y, w, h) coordinates of the left, right, upper and bottom part boxes as given in Fig. 2. For part box transformation, we apply offsets $\{\Delta x^p, \Delta y^p, \Delta w^p, \Delta h^p\}$ for coordinates of each part box using the inverse parameterization of the box regression (Girshick et al. 2014) as

$$\begin{aligned} \hat{x}^p &= x^p + \Delta x^p \cdot w^p, \quad \hat{y}^p = y^p + \Delta y^p \cdot h^p, \\ \hat{w}^p &= \exp(\Delta w^p) \cdot w^p, \quad \hat{h}^p = \exp(\Delta h^p) \cdot h^p, \end{aligned} \tag{2}$$

where $\hat{x}$ is a transformed box and p indicates left, right, upper, and bottom parts. We clip the part boxes to image boundaries to place them within the boundaries. Also, when $\hat{\mathbf{d}}^p$ has a low overlap ratio over $\mathbf{d}^p$, we use the $\mathbf{d}^p$ as a part box instead of using $\hat{\mathbf{d}}^p$. This prevents a transformed part box to be out of an object region too much. (We provide the experimental results in Fig. 4). Figure 2 shows the generation process of deformable part boxes.

Detection and Instance Segmentation Heads

From RDA-Nets, we can extract the combined strong feature responses. We feed the combined $\mathbf{x}_3^{comb}$ and $\mathbf{x}_4^{comb}$ to the box and mask heads, respectively. Here, $\mathbf{x}_3^{comb}$ is a refined feature by merging part features only, then $\mathbf{x}_4^{comb}$ is refined by fusing $\mathbf{x}_3^{comb}$ and $\mathbf{x}^{whole}$ of the whole object model. The box head includes two fully connect (FC) layers with 1024 neurons. The output of the last FC layer for $\mathbf{x}_4^{comb}$ is connected to the object classification and box regression layers with $cls + 1$ neurons and $4 \times cls$ neurons, where cls is the number of object classes and the one is added due to the background class. On the other hand, the FC output for $\mathbf{x}_3^{comb}$ is connected to the same classification layer only during training.

For instance segmentation, we also combine the outputs $\mathbf{x}_3^{comb}$ and $\mathbf{x}_4^{comb}$ of the RDA-Net with the mask head. Since a pixel-wise labeling is required, a stack of fully convolutional network (FCN) (Shelhamer, Long, and Darrell 2017) is usually used as the mask head due to its flexibility, robustness, and fast speed of training and inference. We follow the implementation of (He et al. 2017) for the mask head. It has a stack of four consecutive 3×3 convs, a 2×2 deconv layer with stride 2 for up-sampling the spatial resolution of the inputs by a factor of 2. Then, 1×1 conv is followed to produce cls masks. We use ReLU in the hidden layers. Also, cls masks from the input $\mathbf{x}_3^{comb}$ are extracted by using the same mask head. These masks for the part features are used during training only. In addition, we attach a MaskIoU head after the mask head because it can evaluate the confidence

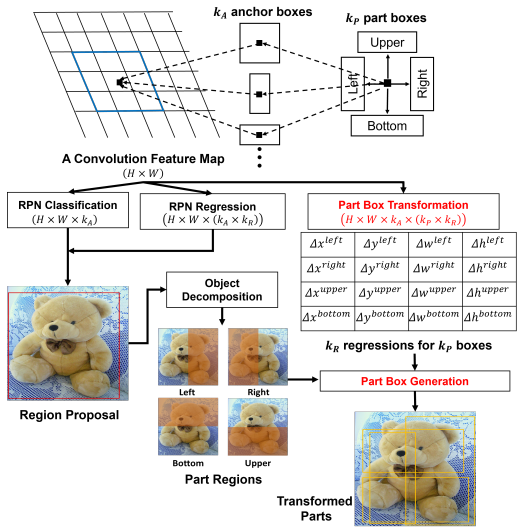

Figure 2: Deformable part region generation: when applying k_A anchors for a feature map of a size $H \times W$, HWk_A proposals are generated. Each proposal consists of k_P boxes with k_R coordinates. We learn $k_P \times k_R$ transformation parameters in the part box transformation layer, and transform each part box by applying the predicted outputs in the part box generation layer.

of each mask more accurately by calculating the pixel-level IoU between the predicted mask and the counterpart ground truth. Following the implementation of (Huang et al. 2019), the input RoI feature and output mask of the mask head are concatenated along the channel dimension, and then the 4 convolutional and 2 fully connected layers are followed to predict the the MaskIoU score per class from the concatenated feature.

Learning D-PRD with Weak Supervision

Because we should localize object part regions without the ground truth of object parts, it can be considered as a weakly supervised learning problem. To handle this problem, we define object parts with four rectangle boxes by decomposing a RoI region. We then use them as reference part boxes. In other words, spatial locations and scales of part boxes are transformed relative to their reference boxes. In return, these decomposed boxes provide good starting points when solving the complex weak-supervision problem.

To lean the transformation parameters of part boxes, we cannot apply the conventional box regression loss (Girshick et al. 2014) directly, which minimizes a mismatch between ground truth and predicted boxes, because ground truth of part regions is unavailable. However, the features extracted from the part boxes still contribute to object classification and segmentation. Therefore, we can solve the weak-supervision problem by minimizing the box classification and mask segmentation losses w.r.t. $\hat{\mathbf{d}}^p$. We first match each box $\mathbf{d}$ and the ground truth box $\mathbf{d}^*$ by evaluating IoU, and assign d to a positive label $o^* \in \{1...cls\}$ if d has an IoU more than 0.5 over any $\mathbf{d}^*$. We assign a negative label ($o^* = 0$) to $\mathbf{d}$ that has an IoU between 0.1 and 0.5. Let $\mathbf{p} = (p^0, ..., p^{cls})$ and $\mathbf{p}^{prt} = (p^{prt,0}, ..., p^{prt,cls})$ denote probability distributions over $cls + 1$ which are computed by feeding $\mathbf{x}_3^{comb}$ and $\mathbf{x}_4^{comb}$ to the box head and applying the softmax for the outputs of the head. Then, we define classification losses of the deformable part and whole models $L_{cls}^{prt}(\mathbf{p}^{prt}, \mathbf{o}^*)$ and $L_{cls}(\mathbf{p}, \mathbf{o}^*)$, and these losses evaluate difference between the prediction of class probabilities and ground truth labels using the cross entropy loss. To reduce the memory burden, the parameters of the box head used for computing $\mathbf{p}$ and $\mathbf{p}^{prt}$ are also shared.

In addition, we add mask losses $L_{mask}(\mathbf{m}, \mathbf{m}^*)$ and $L_{mask}^{prt}(\mathbf{m}^{prt}, \mathbf{m}^*)$ for the multi-task loss. These compare $\mathbf{m}$ and $\mathbf{m}^{prt}$ with the ground truth mask $\mathbf{m}^*$. Here, $\mathbf{m}^{prt}$ and $\mathbf{m}$ are the outputs of the shared mask head for the inputs $\mathbf{x}_3^{comb}$ and $\mathbf{x}_4^{comb}$, respectively. To evaluate MaskIoU losses L_{miou} for part and whole models, we compute the MaskIoU between a binary mask and its counterpart ground truth, and then consider it as the MaskIoU target $\mathbf{s}^*$. We compute the L_2 losses to regress predicted MaskIoUs from whole $\mathbf{s}$ and part $\mathbf{s}^{prt}$ models over the MaskIoU target. As a result, we present a new integral total loss by combining all the losses with whole and part models as follows:

$$
\begin{aligned}
&L\left(\mathbf{p}, \mathbf{p}^{prt}, o^*, \mathbf{t}, \mathbf{t}^*, \mathbf{m}, \mathbf{m}^{prt}, \mathbf{m}^*, \mathbf{s}, \mathbf{s}^{prt}, \mathbf{s}^*\right) \\
&= L_{cls}(\mathbf{p}, \mathbf{o}^*) + \lambda\left[o \geq 1\right] L_{reg}(\mathbf{t}, \mathbf{t}^*) + L_{mask}(\mathbf{m}, \mathbf{m}^*) \\
&+ L_{miou}(\mathbf{s}, \mathbf{s}^*) + L_{cls}^{prt}(\mathbf{p}^{prt}, \mathbf{o}^*) \\
&+ L_{mask}^{prt}(\mathbf{m}^{prt}, \mathbf{m}^*) + L_{miou}^{prt}(\mathbf{s}^{prt}, \mathbf{s}^*)
\end{aligned}
\tag{3}
$$

L_{mask} and L_{mask}^{prt} are the mask losses defined with the average binary cross entropy. The mask head produces $\mathbf{m} = \{\mathbf{m}^1, ..., \mathbf{m}^{cls}\}$ and $\mathbf{m}^{prt} = \{\mathbf{m}^{prt,1}, ..., \mathbf{m}^{prt,cls}\}$ of resolution $h_{roi}^{mask} \times w_{roi}^{mask}$ over cls classes. When evaluating L_{mask} and L_{mask}^{prt}, a mask predicted from an RoI associated with o^* is compared with ground truth mask $\mathbf{m}^*$ only. The MaskIoU head also produces $\mathbf{s} = (s^0, ..., s^{cls})$ and $\mathbf{s}^{prt} = (s^{prt,0}, ..., s^{prt,cls})$ over cls classes. $p^{prt,u}$ is a predicted class probability for class u. For L_{reg}, we evaluate box regression targets $\mathbf{t}$ and $\mathbf{t}^*$ by comparing predicted box $\mathbf{d}$ with its anchor and ground truth boxes for class o^*. During inference, the classification, mask, and MaskIoU predictions from the part models are not exploited. To refine mask scores, the predicted MaskIoU scores are multiplied with the classification scores for the same class masks. $\lambda = 1$ in our implementation.

Cascade Deformable Part Region Detector

The architecture of our Cascade D-PRD is presented in Fig. 3. For each cascade stage n, we refine the classification probability $\mathbf{p}_n$ ($\mathbf{p}_n^{prt}$), bounding box $\mathbf{d}_n$, mask $\mathbf{m}_n$ ($\mathbf{m}_n^{prt}$), MaskIoU score $\mathbf{s}_n$ ($\mathbf{s}_n^{prt}$) for the predicted detection $\mathbf{d}_n$ ($\mathbf{d}_n^p$) from the previous D-PRD. In the first and last stages, we use RDA networks for fusing different region features, and use the combined $\mathbf{x}_{n,3}^{comb}$ and $\mathbf{x}_{n,4}^{comb}$ as shown in Fig. 2. In the second stage, we use the several conv layers. The trainable parameters of the RDA and other head networks used in each stage are not shared. (We design this structure based on Table 1 of the supplementary material.) We set the IoU

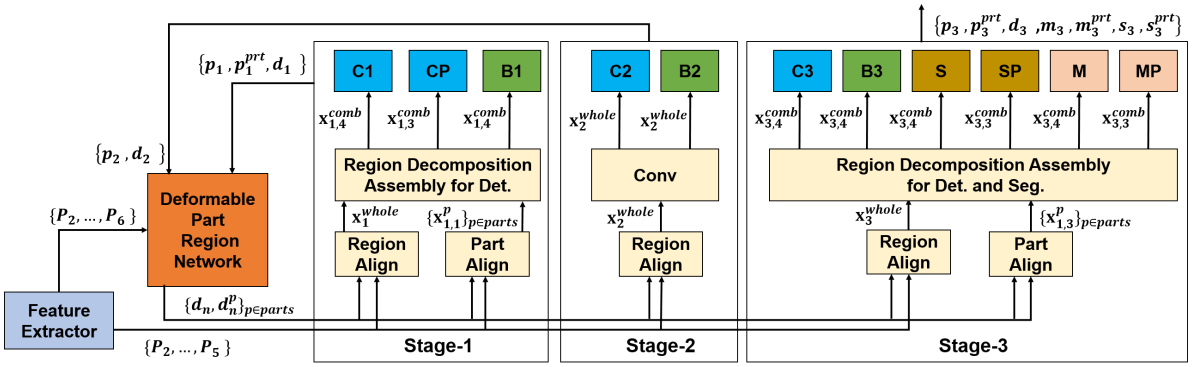

Figure 3: Proposed architecture for the Cascade deformable part region detection . Here, "C" means a classification, "B" a box, "S" a mask, and "M" a mask scoring heads. "CP", "SP", and "MP" are shared with "C", "S", and "M'. At each cascade stage n, we feed whole $\mathbf{x}_{n,4}^{comb}$ and part features $\mathbf{x}_{n,3}^{comb}$ to the original ("C", "B", "S", and "M") and part ("CP", "SP", and "MP") heads.

threshold to (0.5, 0.6, 0.7) from the first to last stage, and encourage the next box and mask heads to produce higher quality results. The outputs of the first two stages are fed to the DRP network. By using the integral loss Eq. (3), we evaluate the loss per stage, and update parameters of our Cascade detector by minimizing all the losses.

Discussion

For improving object feature discriminability, the obvious way is to learn the global context around an object and local details within the object. However, the main challenge is to find meaningful regions to learn these contexts. The meaningful region could be a larger region including other interacting objects or background, or a smaller region containing object crucial parts for the local context. To learn the global context, (Cai et al. 2016; Chen et al. 2020; Yang et al. 2018) fuse the holistic image and object RoI features. However, these methods would often miss the object part details after the global feature fusion. On the other hand, (Bae 2019; Wan, Eigen, and Fergus 2015; Zhang et al. 2018a; Zhou and Yuan 2018) can learn the local context, but learning the global context is challenging since their part models are fixed or limited deformablity within the object. Compared to those works, our DPM can learn both contexts because its high deformability of part models within or around an object. Remarkably, the deformability is tuned for each object class and size during the multi-task learning Eq. (3) without any labels of part boxes (*c.f.* (Zhang et al. 2018a; Zhou and Yuan 2018)).

Experimental Results

Our D-PRD and Cascade D-PRD are evaluated on MSCOCO17 (Lin et al. 2014) and PASCAL VOC07/12 (Everingham et al. 2015) datasets. To show the effects of proposed methods, we present the ablation study. Then, comparison results with recent detectors are provided on the benchmark datasets.

Backbone	Baseline	DPR	Part Loss	Cascade	Fixed	box AP	mask AP
R50-FPN	√					40.98	37.17
	√	√				42.20	38.83
	√	√	√			42.95	39.21
	√			√		44.33	38.46
	√	√	√	√		45.92	39.79
	√	√	√	√	√	44.61	38.67

Table 1: Ablation study on the COCO17 val set .

Implementation

We implement our D-PRD and Cascade D-PRD based on the feature pyramid network (FPN) (Lin et al. 2017a) since it has been widely used as a multi-scale feature extractor for detection and segmentation. We use ResNet50-FPN (R50-FPN), ResNet101-FPN (R101-FPN) (He et al. 2016), ResNeXt101-32x8d (X101-FPN) (Xie et al. 2017). Once collecting feature maps $\{P_2, P_3, P_4, P_5, P_6\}$ of FPN, we distribute them to the DPR and RDA networks. For the DPR network, we use the all feature levels $\{P_2, ..., P_6\}$, but $\{P_2, ..., P_5\}$ for the RDA network. As described in (Lin et al. 2017a), we set anchor sizes to $\{32^2, 64^2, 128^2, 256^2, 512^2\}$ on $\{P_2, ..., P_6\}$, respectively. Also, we apply multiple anchor ratios $\{1:2, 1:1, 2:1\}$ for each anchor. Therefore, total 15 anchors are used over the pyramid.

For RoI pooling, we assign an RoI of width w and height h (on the input image) to the corresponding pyramid level as described in (He et al. 2016). Using the RoIAlign (He et al. 2017), we then extract warped features for whole object and part RoIs at the corresponding level. As shown in Fig. 1, we set the sizes of warped features to 7×7 and 14×14 for detection and segmentation, respectively. We emphasize again that the parameters of the DPR and RDA networks are shared across all pyramid levels and all RoIs of all levels. We use the Detectron2.

Evaluation Setting and Learning Policy

We use the standard COCO metrics. For detection and instance segmentation, box AP and mask AP scores at IoU $\in [0.5 : 0.05 : 0.95]$ are considered as the most important

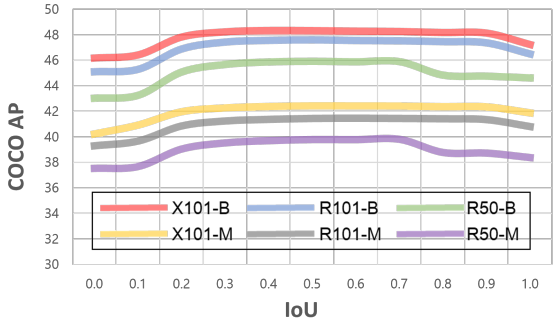

Figure 4: Comparison of the Cascade D-PRD with different backbones by changing σ. Here, σ represents a IoU score between fixed $\mathbf{d}^p$ and transformed $\hat{\mathbf{d}}^p$ part boxes. B and M means box and mask AP scores.

Detector	Parts	Fusion	box AP	Mask AP	Params (M)	Flops (B)	Memory (MiB)	Speed (fps)
R-FCN	DPM	-	34.5	-	50	-	-	5.2
(D1)	Fixed	Concat	38.7	35.2	44	116	2110	11.7
(D2)	Fixed	RDA	40.8	37.6	46	117	2182	10.0
(D3)	DPR	Concat	41.7	38.2	44	117	2170	11.3
(D4)	DPR	RDA	42.9	39.2	46	118	2242	9.5

Table 2: Comparison of (D1-D4) and DPM (Dai et al. 2017) detectors for different part models and feature fusions.

# of Parts (k_P)	RDA Stages (l)	box AP	mask AP	Speed (fps)
Mask R-CNN w/t R50-FPN (baseline)		41.00	37.20	11.29
2	2	40.78	37.58	9.02
4	3	42.95	39.21	8.60
8	4	42.96	39.21	8.18
16	5	42.82	39.10	6.65

Table 3: Effects on the number of decomposed parts.

metrics. We also use AP_{50}, AP_{75}, AP_S, AP_M, and AP_L. Refer to (Lin et al. 2014) for more details.

We use the default learning schedules 1x or 3x ($\sim$ 12 or $\sim$ 37 COCO epochs) of Detectron2 for all the evaluation below. Also, all other setting parameters for training and testing are same to those of Detectron2.

Ablation Experiments

To prove our methods, we provide some ablation studies. We train and evaluate detectors on the COCO dataset.

Effects of each method: We implement our baseline detector by attaching Mask R-CNN (He et al. 2017) to R50-FPN and R101-FPN backbones. Then, we add the proposed method step-by-step on the baseline. Table 1 shows the scores of box AP and mask AP after applying each method. For R50-FPN, our DPR-Net improves box and mask APs by 1.2 and 1.7 points. We improve box and mask APs by 0.8 and 0.4 when training D-PRD with the additional L^{prt}_{cls}, L^{prt}_{mask}, and L^{prt}_{miou} part losses of Eq. (3). Furthermore, Cascade D-PRD can improve box and mask APs by 3.0 and 0.6. Compared to each baseline detector with R50-FPN (R101-FPN), we boost box and mask APs by about 4.9 (4.6) and 2.6 (2.1) points. Using the DPR-Net and part losses only,

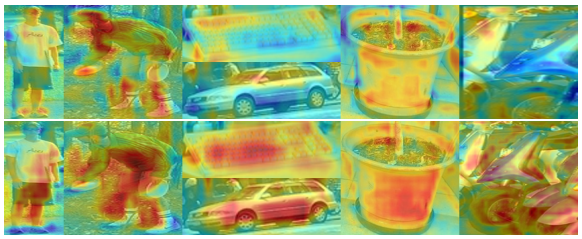

Figure 5: Gradient activations of the Cascade Mask R-CNN (Cai and Vasconcelos 2018) (top) and Cascade D-PRD (bottom) with R50-FPN are compared.

we improve box and mask APs by about 2.0/2.0 (2.1/0.8) for Res50-PFN (R101-FPN). In addition, based on R50-FPN we implement the Cascade Mask R-CNN following the implementation (Cai and Vasconcelos 2018). As shown, our Cascade D-PRD shows the better scores than the Cascade Mask R-CNN. Fixed part models (*i.e.* not deformable) also degrade mAP as shown. We prove that each method is beneficial for improving the mAP scores.

Detailed analysis of DPR and RDA networks: Table 2 shows more ablation study of the main methods. For more comparison with RDA, we implement a feature fusion method (*i.e.* Concat). To this end, we first apply a 1x1 conv to reduce the channel number of $\mathbf{x}^p_1$ by $1/k_P$, and concatenate them along the channel dimension. Then, we combine the whole $\mathbf{x}^w$ and concatenated part feature using one element-wise max unit over each channel. Compared to (D1), the proposed (D4) achieves box 4.2 and mask 4.0 AP gains. However, the cost of using our DPR and RDA networks is not much for parameters and complexity.

Amount of part deformation: When the IoU score between a transformed part $\hat{\mathbf{d}}^p$ and fixed part $\mathbf{d}^p$ is lower than σ, we exploit $\mathbf{d}^p$ instead of $\hat{\mathbf{d}}^p$ to avoid the over-deformablity of part models. To find out the best σ, we implement several Cascade D-PRD with different R50/R101/X101-FPN backbones, and evaluate box and mask APs by varying σ. When $\sigma = 1$, this indicates the decomposed part models are fixed as shown in Fig. 2. $\hat{\mathbf{d}}^p$ could be out of the $\mathbf{d}$ region when $\sigma = 0$.

Figure 4 shows the mAP comparison results of different Cascade R-DADs. For the box AP, all the detectors show the best results using $\sigma = 0.5$, but each achieves the best mask using $\sigma = [0.3, 0.4, 0.5]$, respectively. Note that the maximum differences of box and mask APs for $\sigma = [0.3, 0.7]$ is less than 0.28. Thus, these marginal differences prove that our detector is not sensitive to σ. Compared to results of using the fixed part models, we can improve box and mask AP by about 1.2 and 0.9 points by deforming part boxes in average. This also highlights that the DPR is key.

Number of part models: In Table 3, we evaluate APs and speed with different number of parts (k_P) and RDA stages (l). Decreasing k_P boosts the speed slightly, but degrades box and mask scores. We also evaluate our D-PRD with more parts ($k_P = \{8, 16\}$) and stages ($l = \{4, 5\}$). The accuracy gain is marginal, but the speed is reduced largely. Thus, we opt $k_P = 4$ and $l = 3$ for our implementation.

Method	Backbone	box AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	mask AP
One-stage detectors								
ATSS (Zhang et al. 2020b)	ResNet101-FPN	43.6	62.1	47.4	26.1	47.0	53.6	-
IQDET (Ma et al. 2021)	ResNet101-FPN	45.1	63.4	49.3	26.7	48.5	56.6	-
CentripetalNet (Dong et al. 2020) $*$	Hourglass-104	48.0	65.1	51.8	29.0	50.4	59.9	40.2
Two-stage detectors								
Mask R-CNN w/ SWN (Cai et al. 2020)	ResNeXt101	42.5	64.1	46.6	24.8	46.0	53.5	-
Grid R-CNN (Lu et al. 2019)	ResNeXt101	43.2	63.0	46.6	25.1	46.5	55.2	-
Dynamic R-CNN (Zhang et al. 2020a) $*$	ResNet101	44.7	63.6	49.1	26.0	47.4	57.2	-
D-PRD (ours)	ResNet50-FPN	43.6	63.1	48.1	26.2	46.1	54.0	38.8
D-PRD (ours)	ResNet101-FPN	45.3	64.6	49.9	27.2	48.1	56.7	40.0
Cascade detectors								
Hybrid Task Cascade (Chen et al. 2019)	ResNeXt101-FPN	47.1	63.9	44.7	22.8	43.9	54.6	41.2
HCE Cascade R-CNN (Chen et al. 2020) $\star\star$	ResNet101-FPN	46.5	65.6	50.6	27.4	49.9	59.4	-
QueryInst (Fang et al. 2021)	ResNet101-FPN	47.0	-	-	-	-	-	41.7
SCNET (Vu, Haeyong, and Yoo 2021)	ResNeXt101-FPN	48.3	-	-	-	-	-	42.7
Cascade D-PRD (ours)	ResNet50-FPN	46.5	63.9	51.1	28.8	48.5	57.3	40.1
Cascade D-PRD (ours)	ResNet101-FPN	48.3	65.7	53.0	29.6	51.0	59.8	41.7
Cascade D-PRD (ours)	ResNeXt101-FPN	49.2	66.8	53.9	30.4	51.7	60.9	42.4
Cascade D-PRD (ours) $*$	ResNeXt101-FPN	51.1	69.2	56.0	33.3	53.2	63.5	44.7

Table 4: Comparison results on the COCO19 test-dev data set. $\star$ and $*$ uses multi-scale training/testing. Our detection and segmentation results can be founded in the MSCOCO evaluation test-dev2019 bbox and test-dev2019 (segm) websites.

Method	Backbone	AP	AP_{50}	AP_{75}
Cascade RCNN	R50-FPN	51.8	78.5	57.1
Cascade RCNN	R101-FPN	54.2	79.6	59.2
Retina-SWN	R50-FPN	53.4	-	-
Retina-SWN	X101-FPN	56.8	-	-
D-PRD (ours)	R50-FPN	54.4	78.6	61.0
D-PRD (ours)	R101-FPN	56.2	80.0	63.0
Cascade D-PRD (ours)	R50-FPN	60.3	80.6	66.3
Cascade D-PRD (ours)	R101-FPN	61.1	80.8	67.2
Cascade D-PRD (ours)	X101-FPN	60.8	80.8	67.2

Table 5: Comparison of different detectors on *VOC2007 test*.

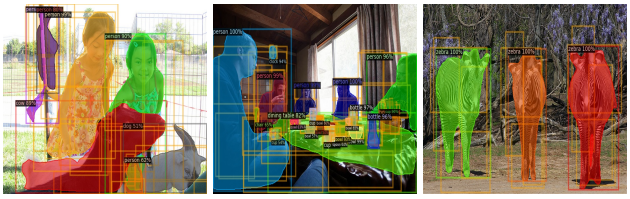

Figure 6: Detection results using our Cascade D-PRD with X101-FPN. Part boxes are colored with orange.

Benchmark Results

We evaluate our Cascade D-PRD on the COCO evaluation server and PASCAL VOC07/12 dataset, and compare our method with state-of-the-art (SOTA) detectors.

MSCOCO19: We participate in the COCO detection challenges and report the best results on evaluation server. Table 4 shows the comparison results with SOTA detectors. Without bells and whistles, we achieve the best 49.2 box AP and 42.4 mask AP with the X101-FPN backbone. In addition, our Cascade D-PRDs with R50-FPN and R101-FPN show the much better scores than other detectors with the better backbone even (*e.g.* Grid R-CNN, HCE, etc). In addition, we achieve 51.1 box and 44.7 mask APs using multi-scale testing. As shown in this challenge leaderboards, our Cascade D-PRD is ranked on the high place. We believe that more improvement can be achieved by using multi-scale training or model ensemble. We also provide the accuracy of D-PRDs. These results are better than Mask R-CNN w/ SWN (Cai et al. 2020).

PASCAL VOC: We use the VOC07/12 datasets for more comparisons. For training and evaluation, we use the VOC07/12 trainval sets and VOC07 test set, respectively. In Table 5, our Cascade D-PRD shows the best AP scores. In addition, our result overwhelms AP of the Retina-SWN (Cai et al. 2020) using the same backbone. Our D-PRDs show the better AP scores than Retina-SWN with the same backbone.

Qualitative Results: Figure 6 shows the visualization results. We depict detected whole and part bounding boxes of each object. As shown, object part regions are learned to be deformable according to an object category, scale, and pose. The deformability of the part boxes can improve the robustness against geometric variations. Furthermore, it represents crucial regions that should be extracted for learning object part details and global context around the object.

Figure 5 compares the gradient activation maps using G-CAM (Selvaraju et al. 2017). As shown, our Cascade D-PRD provides more discriminative and localized gradients within each object proposal than Cascade Mask R-CNN (Cai and Vasconcelos 2018).

Conclusion

In this work, we propose a Cascade D-PRD for high quality of detection and instance segmentation. This is achieved by the deformable part region network which can transform decomposed part regions according to the geometric transformation of an object. For learning transformation parameters without extra supervision, we design a deformable part layer and part model losses. By refining part and whole object models iteratively, we can learn the stronger semantic features. As a result, our Cascade detector overwhelms the performance of the recent detectors.

Acknowledgments

This work was supported in part by the INHA UNIVERSITY Research Grant, in part by the National Research Foundation of Korea (NRF) grants funded by the Korea government(MSIT) (No. NRF-2021R1F1A1049447 and NRF-2022R1C1C1009208) and in part by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (2020-0-01389, Artificial Intelligence Convergence Research Center(Inha University)).

References

Bae, S.-H. 2019. Object detection based on region decomposition and assembly. In *AAAI*, volume 33, 8094–8101.

Bolya, D.; Zhou, C.; Xiao, F.; and Lee, Y. J. 2019. Yolact: Real-time instance segmentation. In *ICCV*, 9157–9166.

Cai, Q.; Pan, Y.; Wang, Y.; Liu, J.; Yao, T.; and Mei, T. 2020. Learning a Unified Sample Weighting Network for Object Detection. In *CVPR*, 14161–14170.

Cai, Z.; Fan, Q.; Feris, R. S.; and Vasconcelos, N. 2016. A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection. In *ECCV*, 354–370.

Cai, Z.; and Vasconcelos, N. 2018. Cascade R-CNN: Delving Into High Quality Object Detection. In *CVPR*.

Chen, K.; Pang, J.; Wang, J.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Shi, J.; Ouyang, W.; Loy, C. C.; and Lin, D. 2019. Hybrid Task Cascade for Instance Segmentation. In *CVPR*.

Chen, Z.; Jin, X.; Zhao, B.; Wei, X.; and Guo, Y. 2020. Hierarchical Context Embedding for Region-based Object Detection. In *ECCV*.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable Convolutional Networks. In *ICCV*, 764–773.

Dong, Z.; Li, G.; Liao, Y.; Wang, F.; Ren, P.; and Qian, C. 2020. CentripetalNet: Pursuing High-Quality Keypoint Pairs for Object Detection. In *CVPR*.

Everingham, M.; Eslami, S. M. A.; Van Gool, L.; Williams, C. K. I.; Winn, J.; and Zisserman, A. 2015. The Pascal Visual Object Classes Challenge: A Retrospective. *IJCV*, 111(1): 98–136.

Fang, Y.; Yang, S.; Wang, X.; Li, Y.; Fang, C.; Shan, Y.; Feng, B.; and Liu, W. 2021. Instances As Queries. In *ICCV*, 6910–6919.

Fu, C.-Y.; Shvets, M.; and Berg, A. C. 2019. RetinaMask: Learning to predict masks improves state-of-the-art single-shot detection for free. *arXiv preprint arXiv:1901.03353*.

Ghiasi, G.; Lin, T.-Y.; and Le, Q. V. 2019. NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. In *CVPR*.

Girshick, R. B.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*, 580–587.

Girshick, R. B.; Iandola, F. N.; Darrell, T.; and Malik, J. 2015. Deformable part models are convolutional neural networks. In *CVPR*, 437–446.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. B. 2017. Mask R-CNN. In *ICCV*, 2980–2988.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*, 770–778.

Huang, Z.; Huang, L.; Gong, Y.; Huang, C.; and Wang, X. 2019. Mask Scoring R-CNN. In *CVPR*.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; and Kavukcuoglu, K. 2015. Spatial Transformer Networks. In *NIPS*, 2017–2025.

Ke, W.; Zhang, T.; Huang, Z.; Ye, Q.; Liu, J.; and Huang, D. 2020. Multiple Anchor Learning for Visual Object Detection. In *CVPR*.

Law, H.; and Deng, J. 2018. Cornernet: Detecting objects as paired keypoints. In *ECCV*, 734–750.

Lin, C.-H.; and Lucey, S. 2017. Inverse Compositional Spatial Transformer Networks. In *CVPR*.

Lin, T.; Dollár, P.; Girshick, R. B.; He, K.; Hariharan, B.; and Belongie, S. J. 2017a. Feature Pyramid Networks for Object Detection. In *CVPR*, 936–944.

Lin, T.; Maire, M.; Belongie, S. J.; Bourdev, L. D.; Girshick, R. B.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. *CoRR*, abs/1405.0312.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017b. Focal loss for dense object detection. In *ICCV*, 2980–2988.

Liu, S.; Qi, L.; Qin, H.; Shi, J.; and Jia, J. 2018. Path Aggregation Network for Instance Segmentation. In *CVPR*.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S. E.; Fu, C.; and Berg, A. C. 2016. SSD: Single Shot MultiBox Detector. In *ECCV*, 21–37.

Lu, X.; Li, B.; Yue, Y.; Li, Q.; and Yan, J. 2019. Grid r-cnn. In *CVPR*, 7363–7372.

Ma, Y.; Liu, S.; Li, Z.; and Sun, J. 2021. IQDet: Instance-wise Quality Distribution Sampling for Object Detection. In *CVPR*.

Redmon, J.; Divvala, S. K.; Girshick, R. B.; and Farhadi, A. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *CVPR*, 779–788.

Ren, S.; He, K.; Girshick, R. B.; and Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NIPS*, 91–99.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *ICCV*, 618–626.

Shelhamer, E.; Long, J.; and Darrell, T. 2017. Fully Convolutional Networks for Semantic Segmentation. *TPAMI*, 39(4): 640–651.

Tan, M.; Pang, R.; and Le, Q. V. 2020. EfficientDet: Scalable and Efficient Object Detection. In *CVPR*.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 9627–9636.

Vu, T.; Haeyong, K.; and Yoo, C. D. 2021. SCNet: Training Inference Sample Consistency for Instance Segmentation. In *AAAI*.

Vu, T.; Jang, H.; Pham, T. X.; and Yoo, C. D. 2019. Cascade RPN: Delving into High-Quality Region Proposal Network with Adaptive Convolution. In *NIPS*, 1430–1440.

Wan, L.; Eigen, D.; and Fergus, R. 2015. End-to-end integration of a Convolutional Network, Deformable Parts Model and non-maximum suppression. In *CVPR*, 851–859.

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated Residual Transformations for Deep Neural Networks. In *CVPR*, 5987–5995.

Yang, T.; Zhang, X.; Li, Z.; Zhang, W.; and Sun, J. 2018. MetaAnchor: Learning to Detect Objects with Customized Anchors. In *NIPS*, 318–328.

Zeng, Z.; Liu, B.; Fu, J.; Chao, H.; and Zhang, L. 2019. WSOD2: Learning Bottom-Up and Top-Down Objectness Distillation for Weakly-Supervised Object Detection. In *ICCV*.

Zhang, H.; Chang, H.; Ma, B.; Shan, S.; and Chen, X. 2019. Cascade RetinaNet: Maintaining Consistency for Single-Stage Object Detection. In *BMVC*, 227.

Zhang, H.; Chang, H.; Ma, B.; Wang, N.; and Chen, X. 2020a. Dynamic R-CNN: Towards High Quality Object Detection via Dynamic Training. In *ECCV*.

Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; and Li, S. Z. 2020b. Bridging the Gap Between Anchor-based and Anchor-free Detection via Adaptive Training Sample Selection. In *CVPR*.

Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; and Li, S. Z. 2018a. Occlusion-Aware R-CNN: Detecting Pedestrians in a Crowd. In *ECCV*, 657–674.

Zhang, X.; Wei, Y.; Feng, J.; Yang, Y.; and Huang, T. 2018b. Adversarial complementary learning for weakly supervised object localization. In *CVPR*.

Zhao, Q.; Sheng, T.; Wang, Y.; Tang, Z.; Chen, Y.; Cai, L.; and Ling, H. 2019. M2Det: A Single-Shot Object Detector based on Multi-Level Feature Pyramid Network. In *AAAI*.

Zhong, Y.; Wang, J.; Peng, J.; and Zhang, L. 2020. Boosting Weakly Supervised Object Detection with Progressive Knowledge Transfer. In *ECCV*.

Zhong, Y.; Wang, J.; Wang, L.; Peng, J.; Wang, Y.; and Zhang, L. 2021. DAP: Detection-Aware Pre-Training With Weak Supervision. In *CVPR*.

Zhou, C.; and Yuan, J. 2018. Bi-box Regression for Pedestrian Detection and Occlusion Estimation. In *ECCV*, 138–154.