Deformable Part Region Learning and Feature Aggregation Tree Representation for Object Detection

Seung-Hwan Bae, *Member, IEEE*

***Abstract*—Region-based object detection infers object regions for one or more categories in an image. Due to the recent advances in deep learning and region proposal methods, object detectors based on convolutional neural networks (CNNs) have been flourishing and provided promising detection results. However, the accuracy of the convolutional object detectors can be degraded often due to the low feature discriminability caused by geometric variation or transformation of an object. In this article, we propose a deformable part region (DPR) learning in order to allow decomposed part regions to be deformable according to the geometric transformation of an object. Because the ground truth of the part models is not available in many cases, we design part model losses for the detection and segmentation, and learn the geometric parameters by minimizing an integral loss including those part losses. As a result, we can train our DPR network without extra supervision, and make multi-part models deformable according to object geometric variation. Moreover, we propose a novel feature aggregation tree (FAT) so as to learn more discriminative region of interest (RoI) features via bottom-up tree construction. The FAT can learn the stronger semantic features by aggregating part RoI features along the bottom-up pathways of the tree. We also present a spatial and channel attention mechanism for the aggregation between different node features. Based on the proposed DPR and FAT networks, we design a new cascade architecture that can refine detection tasks iteratively. Without bells and whistles, we achieve impressive detection and segmentation results on MSCOCO and PASCAL VOC datasets. Our Cascade D-PRD achieves the 57.9 box AP with the Swin-L backbone. We also provide an extensive ablation study to prove the effectiveness and usefulness of the proposed methods for large-scale object detection.**

***Index Terms*—Convolutional object detector, deformable part model, feature aggregation tree, cascade detection, large scale object detection, instance segmentation.**

Manuscript received 6 October 2022; revised 21 March 2023; accepted 14 April 2023. Date of publication 20 April 2023; date of current version 4 August 2023. This work was supported in part by the National Research Foundation of Korea (NRF) under Grant NRF-2022R1C1C1009208, in part by Institute of Information & Communications Technology Planning & Evaluation (IITP) under Grant 2022-0-00448 (Deep Total Recall, 10%), in part by IITP under Grant RS-2022-00155915 (AI Convergence Innovation Human Resources Development: Inha University), and in part by Basic Science Research Program through the NRF under Grant 2022R1A6A1A03051705. Recommended for acceptance by V. Morariu.

The author is with the Vision and Learning Laboratory, Department of Computer Engineering, Inha University, Incheon 22212, South Korea (e-mail: shbae@inha.ac.kr).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2023.3268864, provided by the author.

Digital Object Identifier 10.1109/TPAMI.2023.3268864

I. Introduction

OBJECT detection is to find all the instances of one or more classes of objects given an image. A classical object detector is usually trained with multi-scale image features and classifiers, and finds object instances based on the sliding window search at multi-scale images. Although efficient feature extraction and classification methods [1], [2], [3] were developed, the complexity of the detectors based on the sliding window paradigm usually relies on the resolutions of resized images and the number of pyramid levels.

For efficient object search in an image, some region proposal algorithms based on superpixels [4], [5], [6] and sliding windows [7], [8] have been proposed to hypothesize object regions. In addition, several works [9], [10], [11] show the feature extraction and object classification can be performed by deep convolutional features from end-to-end learning. Therefore, great progress in object detection has been also made by combining the region proposal algorithms and CNN features. The most notable work is the R-CNN [12] framework. They first generate object region proposals using the selective search [4], extract CNN features [13] of the regions, and classify them with class-specific SVMs. Then, Fast RCNN [14] improves training and inference speed using feature sharing and RoI pooling.

The recent convolutional detectors [15], [16], [17] integrate the external region proposal modules into a CNN for boosting the training and detection speed further. As a result, the detection accuracy can be also enhanced by joint learning of region proposal and classification modules. The one-stage detectors [17], [18], [19] slide anchors (*or* predefined default boxes) on feature maps for classifying and regressing object features. Since the most features extracted by sliding anchors are background, costive-sensitive learning [20], [21] is developed to handle the class imbalance. On the other hand, two-stage detectors [22], [23] first find possible object regions using a region proposal network (RPN) [15], then classify the pooled RoI features in the next stage. For more accurate detection, the multi-stage detection methods [24], [25] refine RoI iteratively using a series of consecutive detection headers. For faster detection, anchor-free detectors [21], [26], [27] estimate key points corresponding to object corners and centers directly from image features.

For scale-invariant detection, multi-scale feature representation which generates a feature pyramid at different scales is used for object detection. In order to learn stronger semantic feature

0162-8828 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Inha University. Downloaded on August 10,2023 at 06:07:49 UTC from IEEE Xplore. Restrictions apply.

maps, features propagated from a top-down pathway [28], [29] are fused with bottom-up features by using the lateral connection. Recently, the feature pyramid representation can be enhanced by combining features from multiple pathways with the multi-scale feature fusion [30], [31] and the cross-scale connection [32], [33]. For improving lower layer feature representation, path aggregation FPN (PAFPN) [30] adds the extra bottom-up pathway following the top-down pathway. NAS-FPN [32] searches for a suitable architecture for feature pyramid representation by using the Neural Architecture Search algorithm [34]. AugFPN [35] presents residual feature augmentation to increase the semantic information of higher pyramid levels. EfficientDet [33] leverages top-down and bottom-up feature fusion repeatedly with bi-directional features.

However, these convolutional detectors are still limited to 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 [22]. 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. We carefully design trainable geometric parameters of part models to transform them across spatial location and scale domains. This is the advance compared to existing deformable part models [22], [36], [37] that can adjust the spatial locations of parts. Because the ground truth of the part models is unavailable in general, we present a weak-supervised learning by designing a multi-task loss of part models. 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.

For learning more discriminative features from deformable part regions, we propose a novel feature aggregation tree (FAT) network. We design it based on a binary tree model with multiple region features as leaf nodes and learn the relationship between feature nodes hierarchically based on the bottom-up feature aggregation. In FAT, we present attention learning and mechanism (*or* attention layer) to measure the correlation between different node features. We compute spatial and channel attention maps for roof nodes of two sub-trees, and exploit the attention maps for fusing the most refined features of the trees. As a result, we can generate strong semantic features at the top levels of the FAT, and feed these features to detection heads.

In order to refine detection quality progressively, we present a cascade detector by incorporating the proposed DPR and FAT networks. Compared to the recent cascade detectors [24], [25], [38], we replace region proposal and pooling networks with DPR and FAT networks as shown in Fig. 4. It makes our detector improve box and mask qualities for whole and part models more and more at the next cascade stage. As a result, our cascade scheme achieves more detection gains than recent cascade ones [24], [25], [39], [40].

To sum up, the main contributions of this article can be summarized as follows: (i) proposition of the deformable part region network and part model losses for transforming decomposed object parts and learning the transformation parameters without extra supervision (ii) proposition of the feature aggregation tree network for aggregating multiple region features along the bottom-up pathways (iii) proposition of the attention mechanism for generating a strong and discriminative semantic feature by fusing the most refined features of sub-trees. (iv) proposition of Cascade D-PRD for refining boxes and masks of whole and part models progressively.

Our single deformable part region detector (D-PRD) achieves the state-of-the-art results without employing other performance improvement methods on MSCOCO19. We also make the extensive implementation of D-PRDs with various feature extractors and provide a thorough ablation study to prove the effectiveness of the D-PRD. Without bells and whistles, our Cascade D-PRD achieves impressive 49.9 box and 43.2 mask APs. We have improved the box and mask APs on average by 4.8 (4.0) and 4.2 (2.9) points compared to the recent Cascade R-CNN [24] (HTC [25]) with the same backbones. With the Swin-L backbone and self-learning [41], we can significantly increase our detection score to 57.9. Another benefit is that our DPR and FAT networks can be applied easily to the existing anchor-based detectors with a simple modification.

II. Related Works

In this section, we present the previous study which is related to our work.

A. Object Detection

Due to the progress of deep learning, convolutional object detectors have flourished for object detection and instance segmentation. There are two main streams in convolutional detection. In two-state detectors [14], [15], [23], [42], region of interests (RoIs) are generated from a RPN [15], and object classes and regions are then predicted by feeding the RoI features to the followed R-CNN or its variants. Due to the extra RPN, these detectors show better accuracy, but lower speed. One-stage detectors [16], [17], [18], [33], [43] apply anchors for feature maps, and predict object classes and regions from the extracted features within anchors.

For reducing the complexity further, anchor-free detectors [21], [26], [27], [44] find top peaks within a key point heatmap per class, and consider the peaks as center positions of objects. Then, object regions are determined by regressing them with predicted offsets of object location and size. In addition, there are many efforts [28], [30], [31], [32], [33], [45], [46] to improve multi-scale feature maps based on feature fusion from different scales and pathways.

On the other hand, multi-stage detectors [24], [25], [38] based on a cascade architecture have been developed for improving detection accuracy. These detectors enhance the quality of boxes or masks by refining the predictions progressively with a series of detection headers. As one of the pioneer works, Cascade R-CNN [24] presents a cascade scheme consisting of several

Authorized licensed use limited to: Inha University. Downloaded on August 10,2023 at 06:07:49 UTC from IEEE Xplore. Restrictions apply.

R-CNN heads, and provides a higher quality of detections to the next headers. Following the idea, Cas-RectinaNet [38] has been developed. Hybrid Task Cascade [25] presents an extension of Cascade R-CNN for improving instance segmentation. Cascade RPN [47] improves the region proposal quality by using RPN iteratively. CascadePSP [48] presents a refinement model for high-resolution image segmentation. In [49], [50], cascade architectures for human-object interaction recognition and monocular 3D human pose estimation have been presented.

B. Feature Representation for Detection

Many works [28], [30], [31], [32], [33], [45], [46], [51] tried to improve image feature maps which are used as inputs of the task headers (*e.g.* classification, regression, etc.). Among them, feature pyramid network (FPN) [28] shows that fusing feature maps of bottom-up and top-down pathways is beneficial to learn strong semantic features. After this work, PANet [30] and M2Det [31] improve the FPN further by adding a new pathway and U-shape module for multi-scale feature fusion. [32], [33], [52] present a cross-scale connection for aggregating feature maps at different resolutions. NAS-FPN [32] uses the neural architecture search for finding a feature network architecture automatically. EfficientDet [33] combines top-down and bottom-up features repeatedly with bi-directional features. AFI-GAN [51] presents a GAN-based interpolation for improving features through a top-down pathway.

Recently, the attention mechanism has shown an effective way to weigh and visualize the weighted regions of an image or a feature map. Inspired by that, many attention methods have been also presented in order to improve CNN feature representation further. [53] refines CNN features using spatial and channel attentions between differently pooled features. CenterMask [54] and YOLOv4 [55] exploit spatial attention for improving object detection. The inverted attention method [56] is developed to discover fine-grain discriminative features. [57], [58] use the attention mechanism for improving correlated features between support and query images in few-shot object detection. [59] presents the pyramid-constrained self-attention network for salient object detection. DETR [60] applies the transformer-based encoder and decoder for reasoning the global relations between an image and the objects. [61] addresses the slow convergence of training the DETR. UP-DETR [62] provides unsupervised pre-training with random query patch detection. The attention-guided distillation [63] resolves the imbalance learning problem when applying the knowledge distillation for object detection.

C. Invariant Detection Under Geometric Transformation

Because many recent detectors [14], [15], [17], [23], [33], [64] are based on CNNs with fixed geometric structures, they are inherently limited to model geometric transformations. To improve the robustness over the geometric transformation, [36] presents a DeepPyramid DPM by combining a ConvNet and a deformable part model (DPM) [65]. For joint training of the ConvNet and DPM, [37] designs a loss function by integrating nonmaximum suppression (NMS) inferences. In addition, a spatial transformation network (STN) [66] 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 [67] propagates warped parameters instead of warped images. In deformable CNN [22], spatial offsets are augmented and learned to adjust spatial locations of convolution filtering and RoI pooling. [68], [69] present anchor learning to generate different shapes of anchors. 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 [22], [36], [37], 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 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.

III. Deformable Part Region Detector

For improving the robustness of convolutional detectors over geometric transformations, we propose a deformable part region detector, and provide the overall architecture of the D-PRD in Fig. 1. We introduce a deformable part region network to transform the spatial locations and sizes of part boxes. To learn more discriminative features for object detection, we present a region aggregation tree that combines multi-region features hierarchically. 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 an aggregated feature from part regions and their ground truth. Since the aggregated features between part models are affected by transformed part regions, the deformable part network can be learned by reducing these losses.

A. Deformable Part Region Network

Let $\mathbf{d} = (x, y, w, h)$ be a bounding box, where x, y, w and h are the center positions, width and height[1]. Then, smaller decomposed part regions $\{\mathbf{d}^p = (x^p, y^p, w^p, h^p)\}_{p=1}^{k_P}$ can be generated by dividing $\mathbf{d}$ into several k_P rectangular regions as shown in Fig. 2. For instance, for $k_P = 4$ the smaller decomposed part regions from $\{\mathbf{d}^p\}_{p=1}^{4}$ can be left, right, upper, and bottom part boxes as shown in Fig. 2.

Given a feature map of size $H \times W$, we apply k_A anchors (*or* reference boxes) per location. We consider each anchor box as a possible objectness region. Then, each anchor box $\mathbf{d} = (x, y, w, h)$ can be decomposed into k_P part regions. Therefore, there exist HWk_A possible object regions and HWk_Ak_P part regions in total. We assume that the assigned k_P part boxes to an anchor box are also removed when the anchor is removed by the NMS or score thresholding. This assumption avoids the unnecessary 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 transform

[1]We generate each object bounding box by using the region proposal network (RPN) as shown in Fig. 3. However, other bounding box generation methods can be combined with DPR.

Authorized licensed use limited to: Inha University. Downloaded on August 10,2023 at 06:07:49 UTC from IEEE Xplore. Restrictions apply.

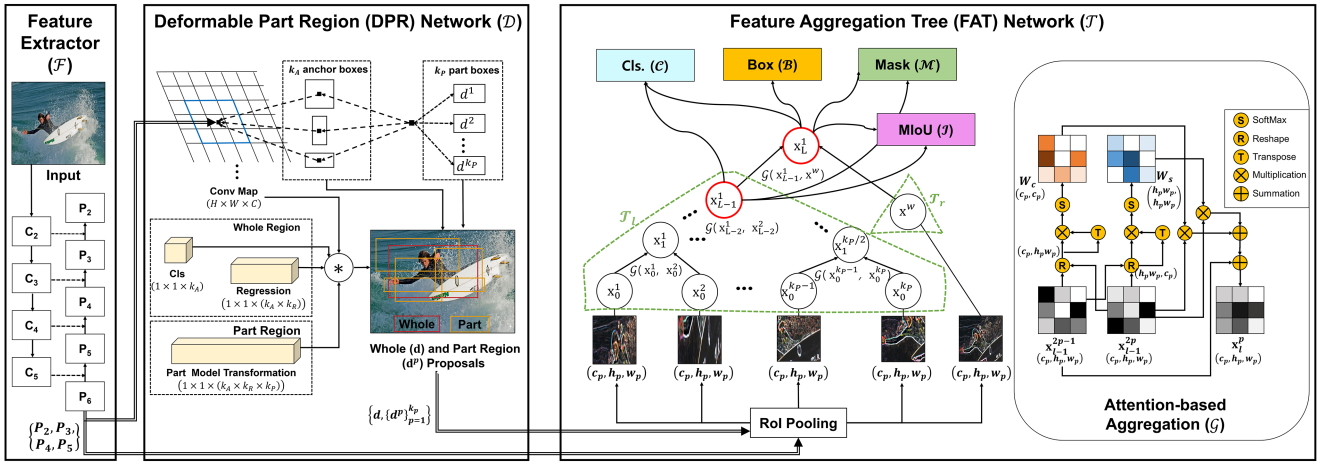


Fig. 1. Proposed deformable part region detector (D-PRD): we use FPN as a feature extractor. In the DPR-Net, k_P part boxes for each proposal are transformed by the the part model transformation layer. Here, $*$ means convolution, and Cls, regression, and part model transformation layers are designed by 1×1 conv. In FAT-Net, we first learn strong semantic features $\mathbf{x}^1_{L-1}$ and $\mathbf{x}^1_L$ by aggregating different region features with the bottom-up aggregation, and feed both features to classification ($\mathcal{C}$), box ($\mathcal{B}$), mask ($\mathcal{M}$), MaskIoU ($\mathcal{I}$) heads.

Fig. 2. Decomposed part regions $\mathbf{d}^p$ with different k_P.

four coordinates (x^p, y^p, w^p, h^p) of a part box. Therefore, the part model transformation layer has $k_A \times k_R \times k_P$ channels in order to transform k_R coordinates of $k_A \times k_P$ part boxes centered at each pixel of the Conv Map of the size $H \times W$.

As shown in Fig. 3, we first predict the offsets $\{\Delta x^p, \Delta y^p, \Delta w^p, \Delta h^p\}$ to transform each part box by convolving the feature Conv Map with the part model transformation layer. The part box transformation is then achieved by applying offsets $\{\Delta x^p, \Delta y^p, \Delta w^p, \Delta h^p\}$ for coordinates (x^p, y^p, w^p, h^p) of each part box using the inverse parameterization of the box regression [12] as follows:

$$\hat{x}^p = x^p + \Delta x^p \cdot w^p, \quad \hat{y}^p = y^p + \Delta y^p \cdot h^p,$$
$$\hat{w}^p = \exp(\Delta w^p) \cdot w^p, \quad \hat{h}^p = \exp(\Delta h^p) \cdot h^p, \qquad (1)$$

where $\hat{\mathbf{d}}^p = (\hat{x}^p, \hat{y}^p, \hat{w}^p, \hat{h}^p)$ is a transformed box for p-th part. Therefore, the locations and sizes of the part boxes assigned to all remaining anchor boxes can be transformed after this transformation. We clip the part boxes to image boundaries to place them within the boundaries. Also, when the transformed $\hat{\mathbf{d}}^p$ has a low overlap ratio over its original $\mathbf{d}^p$, we replace $\hat{\mathbf{d}}^p$ with $\mathbf{d}^p$. This prevents a transformed part box to be out of an object region too much. (We provide the experimental results in Fig. 5).

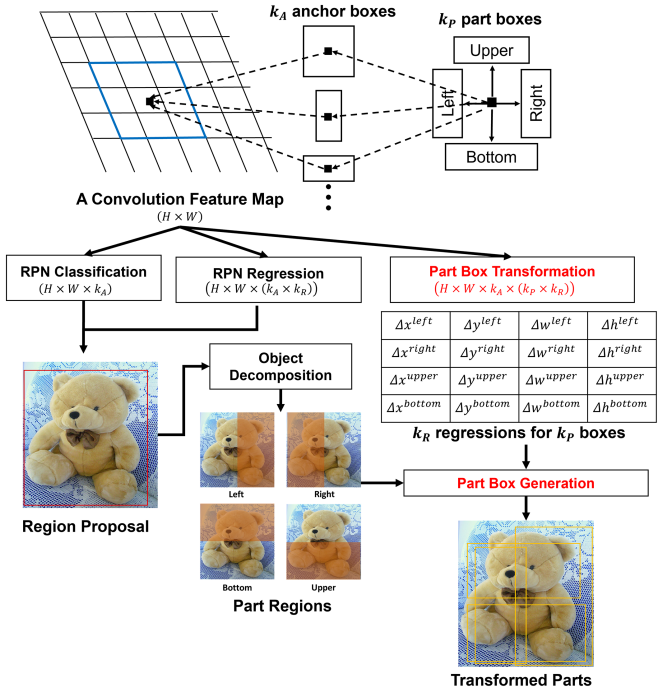


Fig. 3. Deformable part region generation: when applying k_A anchors for a feature map of a size $H \times W$, HWk_A proposals are generated. Each proposal consists of k_P boxes with k_R coordinates. We learn $k_P \times k_R$ transformation parameters in the part box transformation layer, and transform each part box by applying the predicted outputs in the part box generation layer.

B. Feature Aggregation Tree

Given object $\mathbf{d}$ and transformed part $\mathbf{d}^p$ region proposals from the DPR-Net, a region aggregation tree network $\mathcal{T}$ learns stronger semantic features by aggregating the region features hierarchically in a bottom-up fashion. As an input of $\mathcal{T}$, we can extract a warped feature $\mathbf{x}^w$ for $\mathbf{d}$ at the corresponding scale (*or* pyramid) level of multi-scale pyramid features (*e.g.* FPN [28], PANet [30], and BiFPN [33]) in consideration of

Authorized licensed use limited to: Inha University. Downloaded on August 10,2023 at 06:07:49 UTC from IEEE Xplore. Restrictions apply.

their box sizes. Similarly, we can extract warped part feature maps $\{\mathbf{x}_0^p\}_{p=1}^{k_P}$ of the same size for each part box$\{\mathbf{d}^p\}_{p=1}^{k_P}$. For the RoI feature extraction for whole object and its parts, we use the RoIAlign [23]. We consider $\mathcal{T}$ as a binary tree of the level (*or* depth) L[2] consisting of a left $\mathcal{T}_l$ and right $\mathcal{T}_r$ subtrees. $\mathcal{T}_l$ is assumed to be a perfect binary tree of the level $L-1$ where nodes are full per level. On the other hand, $\mathcal{T}_r$ has one node $\mathbf{x}^w$ only.

Given $k_P = 2^{L-1}$ part RoI features (since $\mathcal{T}_l$ is a perfect binary tree), we first construct $\mathcal{T}_l$ by fusing leaf nodes from left to right at $l=0$ to generate their parent nodes at next level. To this end, we define a fusion block $\mathcal{G}$. The fusion block $\mathcal{G}$ uses an attention mechanism by correlating different region features as described in Section III-C. Then, a parent node $\mathbf{x}_l^p$ with stronger semantic features can be generated by merging features of its two children as:

$$\mathbf{x}_l^p = \mathcal{G}\left(\mathbf{x}_{l-1}^{2p-1}, \mathbf{x}_{l-1}^{2p}\right), \quad l \geq 1 \ \text{ and } \ 1 \leq p \leq 2^{L-l} \tag{2}$$

The block $\mathcal{G}$ is shared across levels in $\mathcal{T}$. From the the bottom-up aggregation for $L-1$ phases, we can construct $\mathcal{T}_l$ rooted at $\mathbf{x}_{L-1}^1$. Sequentially, we can build $\mathcal{T}$ with a root $\mathbf{x}_L^1$ by merging $\mathcal{T}_l$ and $\mathcal{T}_r$. This can also be achieved by merging $\mathbf{x}_{L-1}^1$ and $\mathbf{x}^w$. Note that the depth L of $\mathcal{T}$ is $\log k_P + 1$ since the depth of $\mathcal{T}_l$ is $\log k_P$. Here, the one is added due to the last fusion between $\mathcal{T}_l$ and $\mathcal{T}_r$. We then feed these learned strongest semantic features for object detection.

There are obvious benefits for connecting between $\mathcal{T}_l$ and $\mathcal{T}_r$. For scale-invariant detection, exploiting a feature at a certain level only is not sufficient as mentioned in FPN [28] and BiFPN [33]. $\mathcal{T}_l$ contains stronger saliency, whereas $\mathcal{T}_r$ preserves more locality. In this sense, we can learn more discriminative features by fusing $\mathcal{T}_l$ and $\mathcal{T}_r$ with different aggregation levels. To reduce the semantic gap between them, we however use $\mathcal{G}$. In addition, the skip connection along $\mathcal{T}_r$ pathway would mitigate the gradient vanishing.

C. Attention Learning for FAT

In this section, we elaborate on our attention-based feature aggregation $\mathcal{G}$ in order to learn the correlation between features of different child nodes and generate their parent node feature. We assume that the dimensions of the feature tensors $\mathbf{x}_{l-1}^{2p-1}$ and $\mathbf{x}_{l-1}^{2p}$ at $l-1$ level are the same each other as $(c_p \times h_p \times w_p)$, where c_p, h_p, w_p are the number of channels, height, and width of the RoI feature map after the pooling.

We can reshape the dimension $\mathbf{x}_{l-1}^{2p-1}$ and $\mathbf{x}_{l-1}^{2p}$ from $\mathbb{R}^{c_p \times h_p \times w_p}$ to $\mathbb{R}^{c_p \times h_p w_p}$. Then, we evaluate the normalized channel correlation between child features through their multiplication:

$$W_c = S\left(\mathbf{x}_{l-1}^{2p-1} \otimes \left(\mathbf{x}_{l-1}^{2p}\right)^T\right) \in [0,1]^{C \times C}, \tag{3}$$

where $\otimes$ denotes matrix multiplication. We apply the softmax function $S(\cdot)$ along the last dimension.

In order to learn the spatial attention, we also reshape the $\mathbf{x}_{l-1}^{2p-1}$ and and $\mathbf{x}_{l-1}^{2p}$ to make the dimension as $\mathbb{R}^{h_p w_p \times c_p}$. Similarity, we compute the normalized spatial correlation as

$$W_s = S\left(\mathbf{x}_{l-1}^{2p-1} \otimes \left(\mathbf{x}_{l-1}^{2p}\right)^T\right) \in [0,1]^{h_p w_p \times h_p w_p} \tag{4}$$

Next, the channel and spatial attention summaries for $\mathbf{x}_{l-1}^{2p}$ are computed as $\mathbf{x}_{l-1}^{2p} \otimes W_c$ and $\mathbf{x}_{l-1}^{2p} \otimes W_s$, and then we compute the parent node feature $\mathbf{x}_l^p$ with the attention features as

$$\begin{aligned}\mathbf{x}_l^p &= \mathcal{G}\left(\mathbf{x}_{l-1}^{2p-1}, \mathbf{x}_{l-1}^{2p}\right) \\ &= \mathbf{x}_{l-1}^{2p-1} + \mathbf{x}_{l-1}^{2p} \otimes W_s + \mathbf{x}_{l-1}^{2p} \otimes W_c\end{aligned} \tag{5}$$

Thus $\mathbf{x}_l^p$ encodes the spatial and channel relation of its children with enhanced representation. Also, it is possible to use the attention summary of its sibling $\mathbf{x}_{l-1}^{2p-1} \otimes W_s$ and $\mathbf{x}_{l-1}^{2p-1} \otimes W_c$.

In addition, for training FAT we maximize the mutual information between the regions feature of children nodes. We can represent the mutual information between $\mathbf{x}_{l-1}^{2p-1}$ and $\mathbf{x}_{l-1}^{2p}$ as

$$\begin{aligned}&\mathcal{I}\left(\mathbf{x}_{l-1}^{2p-1}, \mathbf{x}_{l-1}^{2p}\right) \\ &= \int\int p\left(\mathbf{x}_{l-1}^{2p-1}, \mathbf{x}_{l-1}^{2p}\right) \log \frac{p\left(\mathbf{x}_{l-1}^{2p-1}, \mathbf{x}_{l-1}^{2p}\right)}{p\left(\mathbf{x}_{l-1}^{2p-1}\right) p\left(\mathbf{x}_{l-1}^{2p}\right)} d\mathbf{x}_{l-1}^{2p-1} d\mathbf{x}_{l-1}^{2p} \\ &= D_{KL}\left(\mathbb{P}_{(\mathbf{x}_{l-1}^{2p-1}, \mathbf{x}_{l-1}^{2p})} || \mathbb{P}_{\mathbf{x}_{l-1}^{2p-1}} \otimes \mathbb{P}_{\mathbf{x}_{l-1}^{2p}}\right)\end{aligned} \tag{6}$$

The mutual information can capture nonlinear statistical dependencies and is zero if and only if both features are mutually independent. The mutual information is equivalent to the Kullback-Leibler (KL-) divergence between the joint $\mathbb{P}_{(\mathbf{x}_{l-1}^{2p-1}, \mathbf{x}_{l-1}^{2p})}$ and the product of the marginals $\mathbb{P}_{\mathbf{x}_{l-1}^{2p-1}} \otimes \mathbb{P}_{\mathbf{x}_{l-1}^{2p}}$. Because optimizing (6) is challenging, we exploit the dual representation of the KL-divergence [70] as follows:

$$D_{KL}(\mathbb{J} || \mathbb{M}) = \sup_{T: \mathbb{R}^d \to \mathbb{R}} \mathbb{E}_J[T] - \log\left(\mathbb{E}_M[e^T]\right) \tag{7}$$

For simplicity, we denote $\mathbb{J}$ and $\mathbb{M}$ as the joint $\mathbb{P}_{(\mathbf{x}_{l-1}^{2p-1}, \mathbf{x}_{l-1}^{2p})}$ and marginals $\mathbb{P}_{\mathbf{x}_{l-1}^{2p-1}} \otimes \mathbb{P}_{\mathbf{x}_{l-1}^{2p}}$. The supremum can be taken over any class of functions T such that the two expectations are finite. In general, $\mathcal{T}_\omega(\mathbf{x}_{l-1}^{2p-1}, \mathbf{x}_{l-1}^{2p}) \in \mathbb{R}$ can be modeled by a neural network with parameter ω. The expectation is evaluated using empirical samples from $\mathbb{J}$ and $\mathbb{M}$ or by shuffling the samples from the joint distribution along the batch axis. At the level of the FAT, we maximize the mutual information with respect to the parameters of the attention aggregation $\mathcal{G}$ and $\mathcal{T}$ as:

$$\begin{aligned}&\mathcal{I}\left(\mathbf{x}_{l-1}^{2p-1}, \mathbf{x}_{l-1}^{2p}\right) \geq \\ &\mathbb{E}_J\left[T_\omega(\mathbf{x}_{l-1}^{2p-1}, \mathbf{x}_{l-1}^{2p})\right] - \log\left(\mathbb{E}_M\left[e^{T_\omega(\mathbf{x}_{l-1}^{2p-1}, \mathbf{x}_{l-1}^{2p})}\right]\right)\end{aligned} \tag{8}$$

We maximize the $\mathcal{I}$ for all feature nodes when $l \geq 1$. Therefore, the total mutual information of FAT can be represented as:

$$\underset{\omega, W_s, W_c}{\operatorname{argmax}} \mathcal{I}_{FAT}(\mathcal{T}) = \sum_{l=1}^{L} \sum_{p=1}^{2^{L-l}} \mathcal{I}\left(\mathbf{x}_{l-1}^{2p-1}, \mathbf{x}_{l-1}^{2p}\right) \tag{9}$$

Since we aim at maximizing the mutual information instead of estimating the exact value, we can reformulate (8) into formulations of non-KL divergences. We can exploit the

[2]We denote $l=0$ as a leaf level due to the bottom-up aggregation.

Authorized licensed use limited to: Inha University. Downloaded on August 10,2023 at 06:07:49 UTC from IEEE Xplore. Restrictions apply.

f-divergences [71] which can measure the differences of two probability distributions. In addition, Noise-Contrastive estimations [72] (*or* InfoNCE) can be used to maximize the lower bound on mutual information. In Table V, we compare the detection performance by applying the different divergences or InfoNCE.

D. Detection Heads

We feed the aggregated discriminative features $\mathbf{x}^1_{L-1}$ and $\mathbf{x}^1_L$ from the FAT network to each head as shown in Fig. 1. To reduce the memory burden, all the detection heads are shared when predicting outputs for the given inputs $\mathbf{x}^1_{L-1}$ and $\mathbf{x}^1_L$. The outputs extracted from the part features are used for training only. Here, $\mathbf{x}^1_{L-1}$ is a refined feature by aggregating part features only during $L-1$ phases, but $\mathbf{x}^1_L$ is refined by fusing $\mathbf{x}^1_{L-1}$ and $\mathbf{x}^w$ of the whole object model.

The box head includes two fully connected (FC) layers with 1024 neurons. The last FC layer is connected to the 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. We use $\mathbf{x}^1_{L-1}$ for classification only.

For instance segmentation, we also propagate the outputs $\mathbf{x}^1_{L-1}$ and $\mathbf{x}^1_L$ of the FAT to the mask head. Since pixel-wise labeling is required, a stack of fully convolutional networks (FCNs) [73] is usually used as the mask head due to its flexibility, robustness, and fast speed of training and inference. We follow the implementation of [23] 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 $\mathbf{x}^1_{L-1}$ are extracted by using the same mask head.

In addition, we attach a MaskIoU head after the mask head because it can evaluate the confidence of each mask more accurately by calculating the pixel-level IoU between the predicted mask and the counterpart ground truth. Following the implementation of [42], 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 MaskIoU score per class from the concatenated feature.

E. Training

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 learn the transformation parameters of part boxes, we cannot apply the conventional box regression loss [12] directly, which minimizes a mismatch between the ground truth and predicted boxes, because the 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 $\mathbf{d}$ to a positive label $o^* \in \{1,\ldots,cls\}$ if $\mathbf{d}$ 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 $\mathbf{p}=(p^0,\ldots,p^{cls})$ and $\mathbf{p}^{prt}=(p^{prt,0},\ldots,p^{prt,cls})$ denote probability distributions over $cls+1$ which are computed by feeding $\mathbf{x}^1_{L-1}$ and $\mathbf{x}^1_L$ from the FAT network $\mathcal{T}$ 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 $L^{prt}_{cls}(\mathbf{p}^{prt},o^*)$ and $L_{cls}(\mathbf{p},o^*)$, and these losses evaluate the difference between the prediction of class probabilities and ground truth labels using the cross entropy loss.

In addition, we add mask losses $L_{mask}(\mathbf{m},\mathbf{m}^*)$ and $L^{prt}_{mask}(\mathbf{m}^{prt},\mathbf{m}^*)$ 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 $\mathbf{x}^1_{L-1}$ and $\mathbf{x}^1_L$, 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:

$$\begin{aligned} &L_{DPR}\left(\mathbf{p},\mathbf{p}^{prt},o^*,\mathbf{t},\mathbf{t}^*,\mathbf{m},\mathbf{m}^{prt},\mathbf{m}^*,\mathbf{s},\mathbf{s}^{prt},\mathbf{s}^*\right)\\ &=L_{cls}(\mathbf{p},o^*)+\lambda\left[o\geq 1\right]L_{reg}(\mathbf{t},\mathbf{t}^*)+L_{mask}(\mathbf{m},\mathbf{m}^*)\\ &+L_{miou}(\mathbf{s},\mathbf{s}^*)+L^{prt}_{cls}(\mathbf{p}^{prt},o^*)\\ &+L^{prt}_{mask}(\mathbf{m}^{prt},\mathbf{m}^*)+L^{prt}_{miou}(\mathbf{s}^{prt},\mathbf{s}^*) \end{aligned} \tag{10}$$

L_{mask} and L^{prt}_{mask} are the mask losses defined with the average binary cross entropy. The mask head produces $\mathbf{m}=\{\mathbf{m}^1,\ldots,\mathbf{m}^{cls}\}$ and $\mathbf{m}^{prt}=\{\mathbf{m}^{prt,1},\ldots,\mathbf{m}^{prt,cls}\}$ of resolution $h^{mask}_{roi}\times w^{mask}_{roi}$ over cls classes. When evaluating L_{mask} and L^{prt}_{mask}, a mask predicted from an RoI associated with o^* is compared with ground truth mask $\mathbf{m}^*$ only. The MaskIoU head also produces $\mathbf{s}=(s^0,\ldots,s^{cls})$ and $\mathbf{s}^{prt}=(s^{prt,0},\ldots,s^{prt,cls})$ 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^*. To refine mask scores, the predicted MaskIoU scores are multiplied by the classification scores for the same class masks. $\lambda=1$ in our implementation. Finally, we can train all the parameters of the DPR ($\mathcal{D}$) and FAT ($\mathcal{F}$) networks by maximizing (9) and minimizing (10).

F. Discussion of DPR and FAT Networks

In order to improve 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

Authorized licensed use limited to: Inha University. Downloaded on August 10,2023 at 06:07:49 UTC from IEEE Xplore. Restrictions apply.

for the local context. To learn the global context, [46], [68], [74] 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, [37], [75], [76] 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 of 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 (10) without any labels of part boxes (*c.f.* [75], [76]).

Moreover, there are also several benefits of the proposed feature aggregation tree method. We can consider the FAT as a network of extracting the strongest semantic or saliency features across spatial and channel dimensions between feature maps of object parts. We can improve the feature invariance against geometric object pose and part configuration variations because we learn the stronger feature responses and feed them to the following object detection heads. Moreover, the FAT is the generalized method of RDA-Net [77] which merges four part features. However, the arbitrary k_P parts (in the form of power of 2) can be merged by using FAT. Also, the weighted feature aggregation using our attention mechanism leads to more informative and softened features rather than learning maxout features [77]. For the attention learning, other methods learn the feature correlation between whole feature map [53], [57] or object RoI [54], [55] levels. However, our FAT can learn the more detailed self-correlation of an object because the smaller part features are correlated each other and then the last refined part feature ($\mathbf{x}^1_{L-1}$) is correlated with the whole object feature ($\mathbf{x}^w$).

IV. Cascade Deformable Part Region Detector

In general, cascade object detection refines detections for N stages by feeding the results at previous stage to the input at next stage. For improving this multi-stage detection, we exploit the proposed DPR-Net ($\mathcal{D}$) and FAT-Net ($\mathcal{T}$) described in Section II-I-A and B. We also attach several detection heads to $\mathcal{T}$ described in Section III-D. Now, we formulate our cascade object detection scheme per $n(\geq 1)$ stage.[3] When the stage $n = 1$, we refine all the object $\mathbf{d}_0$ and part $\{\mathbf{d}^p_0\}^{k_P}_{p=1}$ boxes from the DPR-Net ($\mathcal{D}$). Otherwise, the refined bounding boxes $\mathbf{d}_{n-1}$ and $\{\mathbf{d}^p_{n-1}\}^{k_P}_{p=1}$ from the previous stage are handled. At each stage n, we first feed $\mathbf{d}_{n-1}$ and $\{\mathbf{d}^p_{n-1}\}^{k_P}_{p=1}$ to the tree $\mathcal{T}$ in order to extract aggregation features for the whole object $\mathbf{x}^b_{n,L}$ and the combined part $\mathbf{x}^b_{n,L-1}$ boxes:

$$\left(\mathbf{x}^b_{n,L-1}, \mathbf{x}^b_{n,L}\right) = \mathcal{T}\left(\mathbf{d}_{n-1}, \{\mathbf{d}^p_{n-1}\}^{k_P}_{p=1}\right) \tag{11}$$

Subsequently, we feed the aggregation features $\mathbf{x}^b_{n,L-1}$ and $\mathbf{x}^b_{n,L}$ to the box regression $\mathcal{B}$ and classification $\mathcal{C}$ layers:

$$\begin{aligned} \mathbf{p}_n &= \mathcal{C}_n(\mathbf{x}^b_{n,L}), \quad \mathbf{p}^{prt}_n = \mathcal{C}_n(\mathbf{x}^b_{n,L-1}) \\ \mathbf{d}_n &= \mathcal{B}_n(\mathbf{x}^b_{n,L}), \quad \{\mathbf{d}^p_n\}^{k_P}_{p=1} = \mathcal{D}\left(\mathbf{d}_n\right) \end{aligned} \tag{12}$$

[3]For simplicity, we omit a node index 1 for the root features of $\mathcal{T}$.

From the classification layers, the classification probabilities $\mathbf{p}_n$ and $\mathbf{p}^{prt}_n$ for object and part models are predicted. In the DPR-Net $\mathcal{D}$, we simply generate part boxes by dividing the input refined boxes $\mathbf{d}_n$ into k_P parts without the additional box transformation. We then provide $\mathbf{d}_n$ and $\{\mathbf{d}^p_n\}^{k_P}_{p=1}$ to the same FAT $\mathcal{T}$ in order to extract their refined mask features $\mathbf{x}^m_{n,L-1}$ and $\mathbf{x}^m_{n,L}$:

$$\left(\mathbf{x}^m_{n,L-1}, \mathbf{x}^m_{n,L}\right) = \mathcal{T}\left(\mathbf{d}_n, \{\mathbf{d}^p_n\}^{k_P}_{p=1}\right) \tag{13}$$

Finally, masks $\mathbf{m}_n$ ($\mathbf{m}^{prt}_n$) and MaskIoU scores $\mathbf{s}_n$ ($\mathbf{s}^{prt}_n$) for objects (and their parts) are predicted with the mask ($\mathcal{M}$) and MaskIoU ($\mathcal{U}$) heads:

$$\begin{aligned} \mathbf{m}_n &= \mathcal{M}_n\left(\mathbf{x}^m_{n,L}\right), \quad \mathbf{m}^{prt}_n = \mathcal{M}_n\left(\mathbf{x}^m_{n,L-1}\right) \\ \mathbf{s}_n &= \mathcal{I}_n(\mathbf{x}^m_{n,L}), \quad \mathbf{s}^{prt}_n = \mathcal{I}_n(\mathbf{x}^m_{n,L-1}) \end{aligned} \tag{14}$$

In our implementation, we use the same $\mathcal{T}$ when extracting box $\mathbf{x}^b_{n,L}$ and mask $\mathbf{x}^m_{n,L}$ aggregation features. Also, the $\mathcal{C}_n$, $\mathcal{M}_n$, and $\mathcal{I}_n$ are shared at the same stage when inferring object and part model results. However, these heads are not shared across the cascade stage.

Fig. 4 compares our architecture with Cascade R-CNN [24] and Hybrid Task Cascade (HTC) [25]. From the viewpoint of system architecture design, the main differences come from integrating our DPR-Net and FAT-Net into the cascade scheme. Note that the DPR-Net and FAT include the RPN and pooling layer, respectively, as shown in Fig. 1. Therefore, our DPR-Net can replace the existing RPN, and generate deformable part boxes as well as whole object boxes.

For the aspect of feature learning, the pooled object RoI features are convolved and then fed to the detection heads in Cascade R-CNN and Hybrid Task Cascade (HTC). On the other hand, in our FAT, the part RoI features as well as object RoI features are extracted and their correlations are learned from the attention mechanism. By providing the attention features to the detection heads at the stage, our Cascade D-PRD achieves the better box AP scores than [24], [25] as shown in Tables VII and X. The overall procedure of the mask predictions at each stage follows the Cascade R-CNN. Therefore, the improved segmentation methods (*e.g.* interleaved execution and mask information flow) of HTC are not applied in our detector. Nevertheless, our Cascade D-PRD also outperforms HTC for mask AP. This shows the effects of the mask attention features $\mathbf{m}_n$ and $\mathbf{m}^{prt}_n$ aggregated by FAT.

For training our cascade D-PRD, we extend the multi-task loss (10) to N cascade stages as

$$L = \sum_{n=1}^{N} \lambda_n L^n_{DPR} \tag{15}$$

We set $\lambda_n = 1$, and $N = 3$ by default as same to [24], [25]. We set the IoU threshold to (0.5, 0.6, 0.7) for box regression at $n = (1, 2, 3)$, and encourage the next box and mask heads to produce higher quality results.

Authorized licensed use limited to: Inha University. Downloaded on August 10,2023 at 06:07:49 UTC from IEEE Xplore. Restrictions apply.

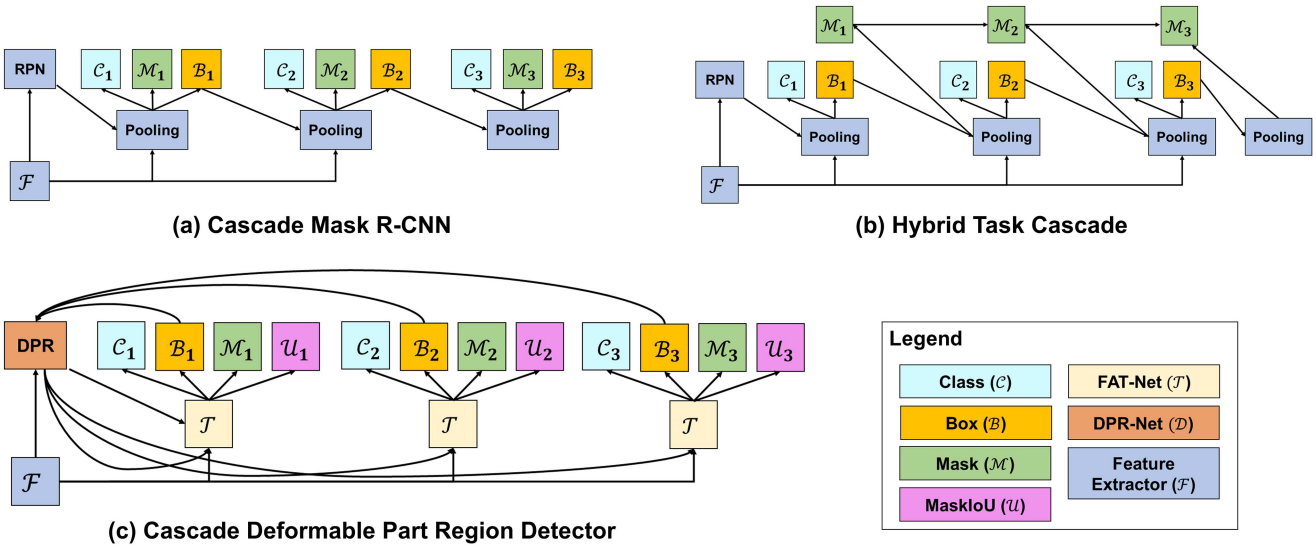


Fig. 4. Comparison of different cascade architectures. In (c), features from transformed part regions of DPR are aggregated per stage n by T_n, and are used for several detection tasks.

We can also extend the total mutual information gain to N stages as

$$I = \sum_{n=1}^{N} \lambda_n \mathcal{I}_{FAT}^{n}(\mathcal{T}_n) = \sum_{n=1}^{N} \sum_{l=1}^{L} \sum_{p=1}^{2^{L-l}} \mathcal{I}\left(\mathbf{x}_{n,l-1}^{2p-1}, \mathbf{x}_{n,l-1}^{2p}\right) \quad (16)$$

For training our cascade D-PRD, the min-max optimization is performed for (15) and (16).

The part detection results (*i.e.* $\mathbf{p}_n^{prt}$, $\mathbf{m}_n^{prt}$, and $\mathbf{s}_n^{prt}$) from the several different heads are used only for evaluating loss (10) and (15). It means that the process of predicting the part detection results is not required in the inference phase. Nevertheless, part detection results $\{\mathbf{d}_n^p\}_{p=1}^{k_P}$ can still contribute to the overall detection since they are used to generate the aggregated attention feature $\mathbf{x}_{n,L}^b$ and $\mathbf{x}_{n,L}^m$ which are inputs of detection and segmentation heads.

V. Experimental Results

Our D-PRD and Cascade D-PRD are evaluated on MSCOCO17 [78] and PASCAL VOC07/12 [79] 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.

A. Implementation

We implement our D-PRD and Cascade D-PRD based on the feature pyramid network (FPN) [28] 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) [11], ResNeXt101-32x8d (X101-FPN) [80]. Once collecting feature maps $\{P_2, P_3, P_4, P_5, P_6\}$ of FPN, we distribute them to the DPR and FAT networks. For the DPR network, we use all the feature levels $\{P_2, \ldots, P_6\}$, but $\{P_2, \ldots, P_5\}$ for the FAT network. As described in [28], we set anchor sizes to $\{32^2, 64^2, 128^2, 256^2, 512^2\}$ on $\{P_2, \ldots, P_6\}$, 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 [11]. Using the RoIAlign [23], 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 FAT networks are shared across all pyramid levels and all RoIs of all levels. We implement all the detectors using the Detectron2 [81]. We train our detectors using multi-scale training with the short edge in the range [640, 800] and the long edge up to 1333 which is the default setting for the FPN-based detector implementation provided by Detectron2.

We set the size of the image batch to 16 for training. Since we use 8 GPUs, we assign 2 images to each GPU. For training Cascade D-PRD, we increase the positive IoU threshold by [0.5, 0.6, 0.7] at each cascade stage. We use the group normalization for normalizing convolution layers. We set the batch size of RoI proposals per image to 512 followed by the default setting of Dectectron2. We set the number of RPN proposals to 1000. We set the channel dimension of the box and mask, and MaskIoU heads to 256. We use 1-MLP and 2-MLPs for the box and MaskIoU heads. We set the pixel standard deviation for R50/101 and X101 backbones to [1.0, 1.0, 1.0] and [57.375, 57.120, 58.395], respectively.

B. Evaluation setting

1) Evaluation Measure: We use the standard COCO-style metrics. For the detected boxes and instance masks, we evaluate IoU scores between predicted results and ground truth results. Then, we evaluate average precision at IoU $\in [0.5 : 0.05 : 0.95]$ (box/mask AP), at IoU 0.5 (AP_{50}), at IoU 0.75 (AP_{75}), and

Authorized licensed use limited to: Inha University. Downloaded on August 10,2023 at 06:07:49 UTC from IEEE Xplore. Restrictions apply.

TABLE I
ABLATION STUDY: EFFECTS OF THE PROPOSED DPR, FAT, AND CASCADE METHODS ON THE COCO2017 VAL SET. BASED ON OUR RE-IMPLEMENTED MASK SCORING R-CNN [42], WE GRADUALLY ADD THE DPR AND FAT NETWORKS, AND CASCADE SCHEME (**CASCADE**). WE USE THE 3× COCO TRAINING SCHEDULE. HERE, **FIXED** MEANS THAT DECOMPOSED PART METHODS ARE NOT DEFORMABLE

Backbone	Baseline	DPR	FAT	Cascade	Fixed	box AP	mask AP
R50-FPN	√					40.88	38.33
	√	√				42.20	38.83
	√	√	√			44.80	41.26
	√			√		44.33	38.46
	√	√	√	√		48.24	42.39
	√	√	√	√	√	46.22	40.56
R101-FPN	√					43.01	39.34
	√	√				44.41	39.73
	√	√	√			47.29	41.61
	√	√	√	√		49.47	42.45
	√	√	√	√	√	47.19	41.31

average precision on small (AP_S), medium (AP_M), large objects (AP_L). When evaluating the metric scores, we use the publicly available code [78] or evaluation servers for those competitions. All the metrics indicate that higher scores are better performance. For each task, box AP and mask AP scores are considered as the most important metrics.

2) Learning Strategy: We use the default learning schedules 1× or 3× (∼12 or ∼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.

C. Ablation Experiments

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

1) Effects of Each Method: We implement our baseline detector (Mask Scoring R-CNN [42]) to R50-FPN and R101-FPN backbones as described in Section III-D. Then, we add the proposed method step-by-step to the baseline. Table I 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 1.3 and 0.5 points over the baseline detector. We improve box and mask APs by 2.6 and 2.4 by using FAT networks. Furthermore, the Cascade D-PRD can improve box and mask APs by 3.4 and 1.1 for box and maks APs. For R101-FPN, we can improve box/mask APs 1.4/0.4 points by using DPR-Net, and 2.9/1.9 points by adding FAT-Net. In addition, our cascade scheme boosts the APs considerably about 2.2/0.8 points.

Remarkably, compared to each baseline detector with R50-FPN(R101-FPN), we boost box and mask APs by about 7.4(6.5) and 4.1(3.1) points by using all the proposed methods (*i.e.* DPR, FAT, and Cascade methods). When using DPR and FAT networks only, we improve box and mask APs by about 3.9/2.9 (4.3/2.3) for R50-PFN (R101-FPN). However, the fixed part models (*i.e.* not deformable) degrades mAP as shown.[4] We prove that each method indeed contributes significantly to improve mAPs for detection and instance segmentation.

2) Detailed Analysis of DPR and FAT: Table II shows more ablation study of the main methods. For more comparison with our FAT, we implement a feature fusion method (*i.e.* Concat). We first apply a 1×1 conv to reduce the channel number of $\mathbf{x}_0^p$ 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. Here, (D1) does use the Concat method which is the simple feature fusion above mentioned, and its parts are fixed. (D2) uses our FAT but its parts are not deformable. (D3) uses DPR and Concat methods. (D4) is the extension of (D3). It has the additional four consecutive convolution layers with the kernel size of 3 after box and mask RoI pooling. Then, convolved RoI features are merged by the Concat method. (D5) uses the DPR and FAT methods. We also present the performance of the DPM method [22]. Compared to (D1), (D2) and (D3) improve box /mask APs by 2.1/2.3 and 3.0/3.0. These results show the effects of our deformable part model and feature aggregation method. Note that the proposed (D4) achieves box 4.3 and mask 4.5 AP gains compared to (D1). Compared to (D4) adding more convolution layers to (D3), (D5) still shows the better results. In particular, adding more Conv layers can improve AP scores, but increases detector parameters and Flops. As a result, the speed is greatly reduced. However, the cost of using our DPR and FAT networks is not much in term of parameters and complexity as shown.

TABLE II
FOR INFERENCE, COMPARISONS OF DETECTORS WITH R50-FPN TRAINED DURING THE 1× EPOCH ON COCO. WE HAVE IMPLEMENTED DIFFERENT VARIANTS OF D-PRD (D1-D5) WITH DIFFERENT PART MODELS AND FEATURE FUSION METHODS

Detector	Parts	Fusion	box AP (%)	Mask AP (%)	Params (M)	Flops (B)	Memory (MiB)	Speed (fps)
[22]	DPM	-	34.5	-	50	-	-	5.2
(D1)	Fixed	Concat	38.7	35.2	44	116	2110	11.7
(D2)	Fixed	FAT	40.8	37.6	46	117	2182	10.0
(D3)	DPR	Concat	41.7	38.2	44	117	2170	11.3
(D4)	DPR	Concat-C4	42.5	38.7	50	118	2190	7.3
(D5)	DPR	FAT	43.0	39.7	46	118	2242	9.5

3) Amount of Part Deformation: As discussed in Section II-I-A, 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. It is possible that $\hat{\mathbf{d}}^p$ is non-overlapped with $\mathbf{d}$ when $\sigma = 0$. Thus, the lower σ allows the part models to be deformable more.

Fig. 5 shows the mAP comparison results of different Cascade R-DADs with different backbones. 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 deformable part region method is key.

4) Number of Part Models: In Table III, we evaluate APs and speed by applying the different number of parts and tree levels for

[4]More experimental results of part deformablity can be found in Section V-C3.

Authorized licensed use limited to: Inha University. Downloaded on August 10,2023 at 06:07:49 UTC from IEEE Xplore. Restrictions apply.

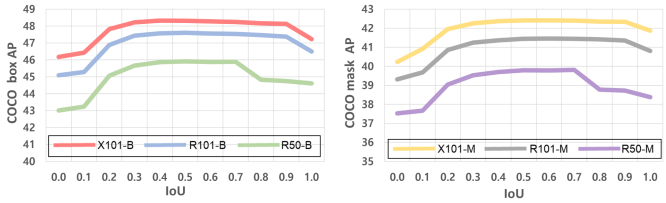


Fig. 5. Comparison of the Cascade D-PRD with different backbones by changing σ. Here, σ represents an IoU score between fixed $\mathbf{d}^p$ and transformed $\hat{\mathbf{d}}^p$ part boxes within the same RoI box. Here, B and M means box and mask AP scores.

TABLE III
EVALUATION WITH THE DIFFERENT NUMBER OF PARTS IN D-PRD

Part Number (k_P)	FAT Level (L)	box AP (%)	mask AP (%)	Speed (fps)
2	2	39.78	37.58	10.3
4	3	43.03	39.69	9.50
8	4	43.10	39.26	8.16
16	5	43.35	39.57	6.65

TABLE IV
COMPARISONS OF CASCADE D-PRD STRUCTURES

Detector	Stage1	Stage2	Stage3	box AP	mask AP	fps
Cascade	Conv	Conv	FAT	47.52	42.20	4.50
D-PRD	FAT	Conv	FAT	48.24	42.39	3.86
wt R50-FPN	FAT	FAT	FAT	49.06	42.57	3.04

FAT. Decreasing k_P boosts the speed, but degrades AP 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.

5) Structure of Cascade D-PRD: To determine the best structure of the Cascade D-PRD, we change the network head of each stage with FAT or Conv networks. We implement the Conv network by feeding RoI features to four 3×3 conv layers with ReLU (without using the DPR-Net). We then train them with the $1\times$ schedule. We compare detection results in Table IV. Using FAT networks at all the stages can improve AP scores, but increases the run-time in return. In consideration of the trade-off, we consider FAT-Conv-FAT as a default scheme of our cascade D-PRD.

6) Mutual Information: For investigating the effect of the mutual information functions used for training the FAT (9), we implement the different mutual information functions based on the f-divergences [71], [82] and InfoNCE [72]. In Table V, we first compare our detectors with fixed part models to show the effects of each function more clearly. We then report the detection scores using deformable part models in the last row. In addition, we provide the detection results of the baseline detector which is described in Section V-C1. We compare the detectors trained for $3\times$ COCO training schedules.

All the detectors trained with the mutual information show better box and mask scores over the baseline. In particular, the JSD-RKL and GAN functions provide the more gains than others. However, the performance differences for different functions are marginal. Since the JSD-RKL provides the best results in terms of both AP metrics, we use this function for training our

TABLE V
COMPARISON WITH DIFFERENT MUTUAL INFORMATION FUNCTIONS. WE EVALUATE THE BOX AND MASK APS FOR OUR DETECTORS WHEN APPLYING EACH MUTUAL INFORMATION FUNCTION ON THE COCO2017 VAL SET

DPR	FAT	MI Function	Output Activation	box AP (%)	mask AP (%)
Baseline (Mask R-CNN)				40.98	37.17
	√	JSD-KL	$\mathbf{x}$	41.76	37.40
	√	JSD-RKL	$-\exp(\mathbf{x})$	41.88	37.46
	√	InfoNCE	$\log(\text{softmax}(\mathbf{x}))$	41.76	37.44
	√	GAN	$-\log(1+\exp(-\mathbf{x}))$	41.80	37.36
	√	Squared Hellinger	$1-\exp(-\mathbf{x})$	41.88	37.45
	√	Pearson χ^2	$\mathbf{x}$	41.93	37.33
√	√	JSD-RKL	$-\exp(\mathbf{x})$	44.80	41.26

TABLE VI
COMPARISON WITH OTHER PART MODEL-BASED DETECTORS. HERE, † INDICATES OUR RE-IMPLEMENTATION RESULTS USING DETECTRON2

Test set	Detector	Backbone	box AP	box AP_{50}	Mask AP
COCO test-dev	Deformable R-FCN [22]	R101	34.50	-	-
	Deformable R-FCN [22]	Inception-ResNet	36.10	-	-
	RDA-Net [77]	R50-FPN	37.10	-	-
	R-DAD †	R50-FPN	40.78	-	37.58
	Cascade R-DAD †	R50-FPN	44.61	-	38.67
	D-PRD (ours)	R50-FPN	45.50	-	40.80
	Cascade D-PRD (ours)	R50-FPN	48.70	-	42.10
VOC test	DP-DPM [37]	VGG	-	45.20	-
	CONV-DPM [38]	ConvNet	-	46.90	-
	D-PRD (ours)	R50-FPN	54.39	78.58	-
	Cascade D-PRD (ours)	R50-FPN	60.31	80.55	-

detectors unless otherwise mentioned in other sections. We also confirm that leveraging the DPR-Net also boosts the detection scores further. Specifically, it improves box and mask APs by 2.92 and 3.81 points when comparing detectors with/without the DPR-Net by using the same JSD-RKL loss.

D. Comparison With Part Model Detectors

Table VI shows the comparison of different part model-based detectors on COCO test-dev and VOC test sets. Compared to Deformable R-FCN [22] and R-DAD [77], our D-PRD can improve AP scores significantly. In Table II, we also provide # params and speed of [22]. Our D-PRD overwhelms [12] for accuracy and speed. For a more fair comparison, we have re-implemented R-DAD and its cascade version. When comparing our detectors with the re-implemented ones, we can improve box and mask APs by 4.72 and 3.22 points for the non-cascade detection and 4.09 and 3.43 for the cascade detection. These results also demonstrate the effects of our deformable part model and feature aggregation methods.

Moreover, we provide comparisons with other part-based detectors on the PASCAL VOC test set. As shown, our detectors are the much better detection scores than DP-DPM [36] and CONV-DPM [37] which are designed based on the pioneer work [65] of the deformable part detection.

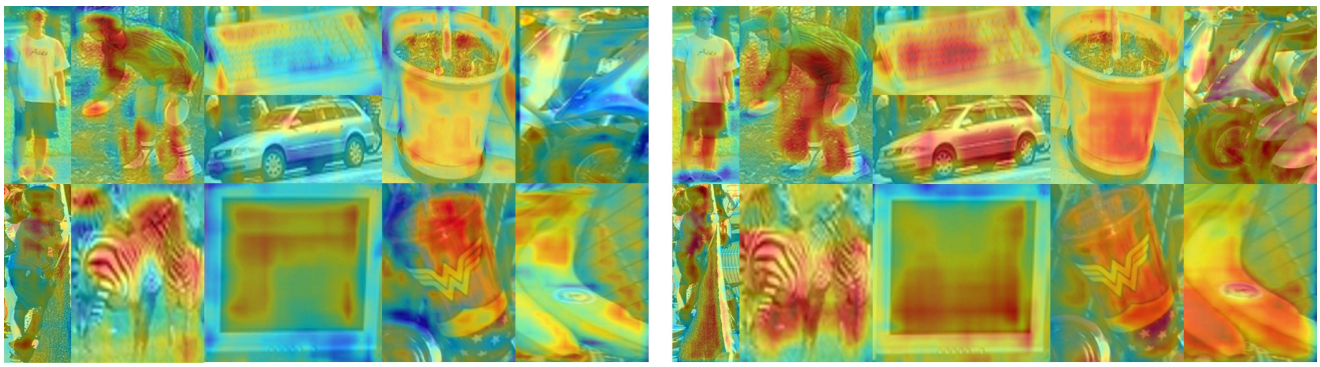
E. Comparison With Cascade Detectors

Our Cascade D-PRDs with different backbones are compared with the Cascade Mask R-CNN and Hybrid Task Cascade (HTC) for detection and segmentation on the COCO2019 test-dev set.

Authorized licensed use limited to: Inha University. Downloaded on August 10,2023 at 06:07:49 UTC from IEEE Xplore. Restrictions apply.

TABLE VII
COMPARISON OF DIFFERENT CASCADE DETECTORS ON THE *COCO2019 TEST-DEV* DATASETS

Method	Backbone	box AP	mask AP	$\mathbf{AP_{50}}$	$\mathbf{AP_{75}}$	$\mathbf{AP_S}$	$\mathbf{AP_M}$	$\mathbf{AP_L}$	speed (fps)
Cascade Mask R-CNN [24]	R50-FPN	42.7	36.9	58.6	39.7	19.6	39.3	48.8	3.0
	R101-FPN	44.4	38.4	60.2	41.4	20.2	41.0	50.6	2.9
	X101-FPN	46.6	40.1	62.7	43.4	22.0	42.8	52.9	2.5
Hybrid Task Cascade (HTC) [25]	R50-FPN	43.6	38.4	60.0	41.5	20.4	40.7	51.2	2.5
	R101-FPN	45.3	39.7	61.8	43.1	21.0	42.2	53.5	2.4
	X101-FPN	47.1	41.2	63.9	44.7	22.8	43.9	54.6	2.1
Cascade D-PRD (ours)	R50-FPN	48.7	42.1	64.0	46.0	25.2	44.3	53.7	3.8
	R101-FPN	49.5	42.8	64.8	46.8	25.2	44.8	55.0	3.6
	X101-FPN	49.9	43.2	65.5	47.3	25.8	45.4	55.2	3.3



(a) Cascade Mask R-CNN [24] (b) Cascade D-PRD (proposed)

Fig. 6. Gradient activations of the Cascade Mask R-CNN [24] (left) and Cascade D-PRD (right) with R50-FPN are compared.

The Cascade Mask R-CNN [24] also means that the mask head is added on the Cascade R-CNN. In addition, HTC [25] further enhances this Cascade architecture for improving segmentation results.

In Table VII, we compare the accuracy and speed of different Cascade detectors. For mask AP, we report AP_{50}, AP_{75}, AP_S, AP_M, and AP_L. For the same backbone, our detectors show the better box and mask AP scores. Compared to Cascade Mask R-CNN, our detector improves 4.8 and 4.2 points in average for box and mask APs. In addition, our detector achieves 4.0 box and 2.9 mask AP gains in average over HTC. For the speed, our detectors are faster about 21.5% and 34.6% over the cascade Mask R-CNN and HTC. Remarkably, the average precision differences between ours and other cascade detectors become larger for more difficult metrics (*i.e.* $\mathbf{AP_{75}}$ and $\mathbf{AP_S}$). In specific, the score differences between Cascade D-PRD and Cascade Mask R-CNN (HTC) for $\mathbf{AP_{50}}$ and $\mathbf{AP_L}$ metrics are about 4.3 (2.9 for HTC) and 3.9 (1.5 for HTC)[5]. On the other hands, the score differences for $\mathbf{AP_{75}}$ and $\mathbf{AP_S}$ are 5.2 (3.6 for HTC) and 4.8 (4.0 for HTC). The main structural differences between the cascade detectors are to leverage deformable part models for object detection as shown in Fig. 4. Therefore, we verify that our deformable part model and feature aggregation method are also beneficial for improving cascade detection. In particular, our deformable part-based cascade detection is more attractive for small object detection and higher-quality detection.

Fig. 6 compares the gradient activation maps. We use the Gradient-weighted class activation mapping (G-CAM) [83] in order to highlight meaningful regions for detection tasks. We demonstrate the G-CAM maps for RoI features extracted from the Cascade detectors. As shown, our Cascade D-PRD provides more discriminative and localized gradients within each object proposal than Cascade Mask R-CNN [24].

[5]The scores are averaged AP scores for three different backbones (R50-FPN, R101-FPN, and X101-FPN).

F. Detailed Evaluation of Feature Aggregation Tree

As shown in Fig. 1, at the leaf level $l = 0$ an aggregation pair of the part RoI features (*i.e.* $x_0^{k_p-1}$ and $x_0^{k_p}$) is determined by their assigned node indices. When decomposing an object box with $k_P = 4$ part boxes, a node index in each sub-region is determined by the following priority: leaf, right, top, and bottom as depicted in Fig. 2. Since these indices affect the aggregation pair and order, we investigate how much the fixed indexing rule affects overall detection performance. To this end, we make a comparison by assigning an index of the leaf nodes in a random manner so as to allow FAT to be more flexible in determining the indices of the leaf nodes. In a similar manner, we determine the aggregation order of part features based on the assigned indices in this case. We then compare the detection scores between the fixed ordering and random ordering in Table VIII. Both ordering strategies show the almost similar AP scores. It implies that our FAT is not sensitive to the order of input features. This is because the aggregation order can be different according to the leaf node index, but all the part features are correlated with each other through several feature aggregations eventually.

In our D-PRD, we replace our deformable part region network with the deformable roi pooling method [22] since both methods can deform pooling regions adaptively. Feature aggregation trees are used for the fair comparison. As shown in Table VIII, our

Authorized licensed use limited to: Inha University. Downloaded on August 10,2023 at 06:07:49 UTC from IEEE Xplore. Restrictions apply.

TABLE VIII
COMPARISON OF THE FEATURE AGGREGATION TREE BY APPLYING DIFFERENT METHODS ON THE COCO2017 VAL SET. THE R50-FPN IS USED AS A BACKBONE

Method	box AP (%)	mask AP (%)
with attention in FAT		
FAT w/t pre-defined indexing	43.00	39.70
FAT w/t random indexing	42.93	39.62
FAT w/t deformable pooling [22]	42.03	37.58
D-PRD w/t deformable conv [22]	44.13	40.91
without attention in FAT		
FAT w/t element-wise addition	39.11	36.39
FAT w/t element-wise max	40.22	37.11

DPR-Net shows the better scores than the deformable pooling. In addition, we evaluate D-PRD with deformable convolution [22]. In specific, we apply the deformable convolution on the stage 3-5 of R50-FPN. Since the deformable convolution enhances the robustness of the detection backbone over the geometric variation, we can improve box and mask AP scores by 1.1 and 1.2 points.

In addition, we compare our detector with/without attention learning to show its effects. To this end, we replace the attention function $\mathcal{G}(\cdot)$ with element-wise sum or element-wise max functions. To keep the channel dimensionality, we apply both non-attention functions across channels. As shown in Table VIII, large accuracy degradation occurs when applying the non-attention functions. Therefore, our spatial and channel attention function is key for learning more powerful aggregation features.

G. Evaluation With One-Stage Detector

To show the compatibility of our methods with other detection frameworks, we combine our cascade D-PRD with the RetinaNet [43]. As a baseline, we use the implemented RetinaNet of Detectron2 since it is the improved version of the original one. We maintain the tuned parameters and use the feature maps $\{P_3, P_4, P_5, P_6\}$ except for the P_2 (*c.f.* used in most two-stage detectors). Then, by feeding the multi-scale features to the RetinaNet, we can predict the box proposals and their confidence scores over cls object classes. Here, the total number of proposals is $\sum_{l=3}^{6} H_l \times W_l \times A_l$, where H_l and W_l are the height and width of the feature map P_l at level l. A_l is the number of the used anchors. However, we keep 512 proposals only by using the score thresholding and non-maximum suppression. We then feed the remaining ones to our Cascade D-PRD shown in Fig. 4. Since the RetinaNet uses classification and box headers only, we do not use Mask and MaskIoU headers per cascade stage for a fair comparison. Except for this, the other default setting used for the Cascade D-PRD is unchanged. We train our Cascade D-PRD w/t Ret with the 3× schedule on the MSCOCO dataset.

In Table IX, we have compared our Cascade D-PRD w/t Ret with the RetinaNet. We can greatly boost the box AP by 5.3 compared to the RetinaNet w/t R50-FPN. In addition, our cascade detector even shows the better scores than RetinaNet w/t R101-FPN. Therefore, more improvement can be easily achieved by using the larger backbone or the higher-resolution P_2 feature map. This experiment also indicates that our Cascade D-PRD can be compatible with other frameworks and provide the obvious accuracy gains.

TABLE IX
COMPARISON BETWEEN THE RETINANET [20] AND OUR CASCADE D-PRD W/T RET DETECTORS ON THE COCO2017 VAL DATASETS

Method	Backbone	box AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
RetinaNet	R50-FPN	38.7	58.0	41.5	23.3	42.3	50.3
RetinaNet	R101-FPN	40.4	60.3	43.2	24.0	44.3	52.2
Cascade D-PRD w/t Ret	R50-FPN	44.0	61.1	47.8	25.0	47.1	60.0

H. 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.

1) MS-COCO: We participate in the COCO detection challenges and report the best results on evaluation server. We train our several versions of D-PRD and Cascade D-PRD with different backbones (*i.e.* ResNet50-FPN, ResNet101-FPN, and X101-FPN). For Cascade D-PRDs, we compare the feature aggregation methods by using region decomposition assembly network (RDA-Net) and feature aggregation tree (FAT) methods. We use the RDA-Net to aggregate two different RoI features of part models instead of using FAT (2). More specially, for $\mathbf{x}_{l-1}^{2p-1}$ and $\mathbf{x}_{l-1}^{2p}$, we can merge the features of children nodes by using region assembly block (RAB) at each level l: $\mathbf{x}_l^p = \text{RAB}(\mathbf{x}_{l-1}^{2p-1}, \mathbf{x}_{l-1}^{2p})$. Here, the RAB consists of four consecutive 3×3 conv filters, ReLU functions, and one element-wise max unit over each channel. The main effect of the RDA method can extract maximum responses across spatial locations between feature maps of object parts. Thus, we can improve the spatial invariance more to feature position without a deep hierarchy than using max pooling which supports a small region (*e.g.* 2×2). More details can be found in [77], [93].

Table X shows the comparison results with SOTA detectors. Without bells and whistles, we achieve the best 49.9 box AP and 43.2 mask AP with the X101-FPN backbone. In addition, our Cascade D-PRD with R50-FPN shows the much better scores than other detectors with the better backbone even (*e.g.* HTC, HCE, D2Det, QueryInst, etc). In addition, we achieve 51.9 box and 45.5 mask APs using multi-scale testing. As shown in these 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 with R50-FPN and R101-FPN. Our detectors show the best results among detectors without using the performance improvement methods.

In addition, we extensively compare our D-PRD and Cascade D-PRD with FAT and RDA-Nets [93]. When comparing the scores for all the backbones (ResNet50-FPN/ResNet101-FPN/ResNeXt101-FPN) of using FAT and RDA-Nets, detectors using FAT shows the better box and mask APs by 1.9 (1.7 for multi-scale testing) and 2.3 (2.1 for multi-scale testing) on average. In particular, the FAT contributes to improving box and mask scores for $\mathbf{AP_S}$ by 3.0 (3.6) and 2.7 (3.4) points. We also

Authorized licensed use limited to: Inha University. Downloaded on August 10,2023 at 06:07:49 UTC from IEEE Xplore. Restrictions apply.

TABLE X
COMPARISON RESULTS ON THE COCO19 TEST-DEV DATA SET. ∗ AND ⋆ ARE MULTI-SCALE TESTING AND DATA AUGMENTATION RESULTS. † AND ○ SHOW THE RE-IMPLEMENTED RESULTS BY OURS AND OUR DETECTOR WITHOUT MASK HEADERS. OUR DETECTION AND SEGMENTATION RESULTS CAN BE FOUNDED IN THE MSCOCO EVALUATION TEST-DEV2019 (BBOX) AND IN THE MSCOCO EVALUATION TEST-DEV2019 (SEGM), RESPECTIVELY

Method	Backbone	box AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	mask AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
One-stage detectors													
HoughNet [84]	ExtremeNet	43.1	62.2	46.8	24.6	47.0	54.4	-	-	-	-	-	-
ATSS [44]	ResNet101-FPN	43.6	62.1	47.4	26.1	47.0	53.6	-	-	-	-	-	-
IQDET [85]	ResNet101-FPN	45.1	63.4	49.3	26.7	48.5	56.6	-	-	-	-	-	-
MAL [86]	ResNeXt101	45.9	65.4	49.7	27.8	49.1	57.8	-	-	-	-	-	-
CentripetalNet [87]	Hourglass-104	46.1	63.1	49.7	25.3	48.7	59.2	38.8	60.4	41.7	19.8	41.3	51.3
HoughNet [84] ∗	ExtremeNet	46.4	65.1	50.7	29.1	48.5	58.1	-	-	-	-	-	-
MAL [86] ∗	ResNeXt101	47.0	66.1	51.2	30.2	50.1	58.9	-	-	-	-	-	-
SAPD [88]	X-101-64x4d-DCN	47.4	67.4	51.1	28.1	50.3	61.5	-	-	-	-	-	-
CentripetalNet [87] ∗	Hourglass-104	48.0	65.1	51.8	29.0	50.4	59.9	40.2	62.3	43.1	22.5	42.6	52.1
Two-stage detectors													
Deformable Mask-RCNN (val17) [22]	ResNet50-FPN	42.7	-	-	-	-	-	37.5	-	-	-	-	-
Mask R-CNN w/ SWN [89]	ResNeXt101	42.5	64.1	46.6	24.8	46.0	53.5	-	-	-	-	-	-
Mask RCNN †	ResNet101-FPN	42.9	63.3	46.8	26.4	46.6	56.1	38.6	60.8	42.2	22.2	41.0	49.8
Grid R-CNN [26]	ResNeXt101	43.2	63.0	46.6	25.1	46.5	55.2	-	-	-	-	-	-
Mask Lab [90]	ResNet101(JFT)	43.0	63.9	47.1	24.8	46.7	55.2	38.1	61.1	40.4	19.6	41.6	51.4
Dynamic R-CNN [91] ∗	ResNet101	44.7	63.6	49.1	26.0	47.4	57.2	-	-	-	-	-	-
DETR [60]	ResNet101	44.9	64.7	47.7	23.7	49.5	62.3	-	-	-	-	-	-
PANet [30]	ResNeXt101	47.4	67.2	51.8	30.1	51.7	60.0	42.0	65.1	45.7	22.4	44.7	58.1
D2Det [92]	ResNet101-FPN	45.4	64.0	49.5	25.8	48.7	58.1	-	-	-	-	-	-
D2Det [92] ∗	ResNet101-deform v2	50.1	69.4	54.9	32.7	52.7	62.1	-	-	-	-	-	-
D-PRD w/t RDA-Net [93]	ResNet50-FPN	43.6	63.1	48.1	26.2	46.1	54.0	38.8	60.8	42.2	22.2	41.0	49.8
D-PRD w/t RDA-Net [93]	ResNet101-FPN	45.3	64.6	49.9	27.2	48.1	56.7	40.0	62.2	43.3	23.0	42.3	51.9
D-PRD w/t FAT (ours)	ResNet50-FPN	45.5	65.3	50.4	27.9	47.7	56.6	40.8	63.0	44.5	24.4	42.8	52.0
D-PRD w/t FAT (ours) ∗	ResNet50-FPN	46.6	66.8	51.7	31.1	48.2	56.8	42.4	64.9	46.3	27.4	43.8	53.0
D-PRD w/t FAT (ours)	ResNet101-FPN	47.6	66.4	52.3	29.0	50.1	59.6	41.9	64.0	45.7	24.5	44.0	54.1
D-PRD w/t FAT (ours) ∗	ResNet101-FPN	49.5	68.9	54.4	32.5	52.0	61.3	44.0	66.7	48.1	27.7	46.0	56.1
Cascade detectors													
Cascade RPN [47]	ResNet50-FPN	40.6	58.9	44.5	22.0	42.8	52.6	-	-	-	-	-	-
Cascade R-CNN [24]	ResNet101-FPN	42.8	62.1	46.3	23.7	45.5	55.2	-	-	-	-	-	-
Cascade R-CNN + Dconv †	ResNet50-FPN	45.3	64.6	49.9	27.2	48.1	56.7	40.0	62.2	43.3	23.0	42.3	51.9
Cascade R-CNN + Dconv †	ResNet101-FPN	46.5	63.9	51.1	28.8	48.5	57.3	40.1	61.8	43.7	23.7	41.9	51.0
Cascade R-CNN [24] ∗	ResNeXt-152	50.9	69.0	55.8	33.4	53.5	63.3	-	-	-	-	-	-
Hybrid Task Cascade (HTC) [25]	ResNeXt101-FPN	47.1	63.9	44.7	22.8	43.9	54.6	41.2	63.9	44.7	22.8	43.9	54.6
HCE Cascade R-CNN [74]	ResNet101-FPN	44.1	63.2	47.9	25.2	46.9	57.0	-	-	-	-	-	-
HCE Cascade R-CNN [74] ∗	ResNet101-FPN	46.5	65.6	50.6	27.4	49.9	59.4	-	-	-	-	-	-
QueryInst [39]	ResNet101-FPN	47.0	-	-	-	-	-	41.7	64.4	45.3	24.2	43.9	53.9
SCNET [40]	ResNeXt101-FPN	48.3	-	-	-	-	-	42.7	65.7	46.4	24.1	45.7	56.3
GCNet [94] ∗	X-101-64x4d-DCN	52.3	70.9	56.9	-	-	-	-	-	-	-	-	-
Copy-Paste [95] ⋆	Cascade Eff-B7 NAS-FPN	57.3	-	-	-	-	-	49.1	-	-	-	-	-
Cascade D-PRD w/t RDA-Net [93]	ResNet50-FPN	46.5	63.9	51.1	28.8	48.5	57.3	40.1	61.8	43.7	23.7	41.9	51.0
Cascade D-PRDw/t RDA-Net [93]	ResNet101-FPN	48.3	65.7	53.0	29.6	51.0	59.8	41.7	63.6	45.5	24.4	43.9	53.2
Cascade D-PRD w/t RDA-Net [93]	ResNeXt101-FPN	49.2	66.8	53.9	30.4	51.7	60.9	42.4	64.7	46.3	25.2	44.7	54.2
Cascade D-PRD w/t RDA-Net [93] ∗	ResNet50-FPN	48.5	66.4	53.2	31.6	50.2	60.1	42.5	64.4	46.5	26.3	44.1	54.3
Cascade D-PRD w/t RDA-Net [93] ∗	ResNet101-FPN	50.2	68.0	55.0	33.0	52.2	61.9	43.8	66.0	48.1	27.4	45.5	55.8
Cascade D-PRD w/t RDA-Net [93] ∗	ResNeXt101-FPN	51.1	69.2	56.0	33.3	53.2	63.5	44.7	67.1	49.1	27.8	46.6	57.2
Cascade D-PRD w/t FAT (ours)	ResNet50-FPN	48.7	66.2	53.4	30.3	51.2	60.4	42.1	64.0	46.0	25.2	44.3	53.7
Cascade D-PRD w/t FAT (ours)	ResNet101-FPN	49.5	67.0	54.2	30.3	51.9	62.0	42.8	64.8	46.8	25.2	44.8	55.0
Cascade D-PRD w/t FAT (ours)	ResNeXt101-FPN	49.9	67.6	54.8	31.0	52.4	62.1	43.2	65.5	47.3	25.8	45.4	55.2
Cascade D-PRD w/t FAT (ours) ∗	ResNet50-FPN	50.2	68.0	55.1	33.3	52.2	61.5	44.1	66.1	48.4	28.1	45.8	55.5
Cascade D-PRD w/t FAT (ours) ∗	ResNet101-FPN	51.1	68.9	56.2	34.4	52.8	62.6	44.7	67.0	49.2	28.9	46.2	56.3
Cascade D-PRD w/t FAT (ours) ∗	ResNeXt101-FPN	51.9	69.9	57.1	34.7	54.0	63.2	45.5	68.0	50.1	29.3	47.3	57.1
Cascade D-PRD w/t FAT (ours) ∗ ○	ResNeXt101-FPN	51.8	69.8	56.9	33.8	54.3	63.9	-	-	-	-	-	-
Cascade D-PRD w/t FAT (ours) ⋆ ○	ResNeXt101-FPN	53.8	71.3	59.1	35.1	57.0	66.7	-	-	-	-	-	-
Cascade D-PRD w/t FAT (ours) ∗ ⋆ ○	ResNeXt101-FPN	55.2	73.1	60.7	38.3	58.0	67.0	-	-	-	-	-	-
Cascade D-PRD w/t FAT (ours) ⋆ ○	Swin-L	55.2	72.6	60.7	36.3	58.5	69.4	-	-	-	-	-	-
Cascade D-PRD w/t FAT (ours) ∗ ⋆ ○	Swin-L	57.9	75.6	63.6	40.9	61.1	71.2	-	-	-	-	-	-

show that FAT is the more effective method for small object detection and segmentation.

For boosting our Cascade D-PRD with ResNeXt101-FPN, we exploit the self-learning [41] on the COCO unlabeled image set. Note that this self-learning is also applied in other recent detectors [95], [96], [97] for the system-level comparison. In specific, we use the box pseudo-labels provided by CenterNet2 [96], and train our Cascade D-PRD on COCO17 train (118 k) and unlabed (123 k) datasets. In addition, we increase the ranges of multi-scale training for the short edge to [480, 1400] and the maximum long edge to 1600. As a result, we achieve 53.8 and 55.2 box AP scores without/with multi-scale testing.

For the more system-level detection comparison, we apply the stronger backbone Swin-L [98] for the our cascade D-PRD. We exploit the ImageNet-22 K pre-trained Swin-L model. We feed multi-scale features of the Swin-L to the subsequent networks (*i.e.* DPR/FAT and several detection headers shown in Fig. 4(c)). We use the default multi-scale training (the shorter side is [640, 800] and longer side is at most 1333), and AdamW optimizer with the initial learning rate of 0.00005, weight decay of 0.05, and batch size of 8. We train our cascade D-PRD with Swin-L by using the 3× schedule on COCO labeled and unlabeled sets. As a result, we can further improve box AP to 55.2. By

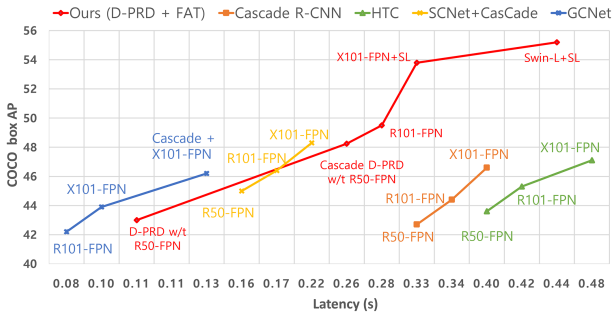


Fig. 7. Comparisons with the recent cascade-based detectors in terms of speed and accuracy. All scores are evaluated for single-model single-scale. Our cascade D-PRD achieves the 55.2 AP while showing the better speed than other detectors (*i.e.* Cascade R-CNN and HTC). Here, SL means the self-learning on the COCO unlabeled dataset.

applying multi-scale testing, the final detection score reaches 57.9. We expect that more enhancement is possible via the more extensive self-training [95] on the unlabeled Object 365 dataset, stronger multi-scale training [98], and longer training schedule and large-scale jitter [81], [95], etc.

Note that our detectors can achieve the almost same detection AP scores (w/t mask of 51.9 and w/o mask of 51.8) when

Authorized licensed use limited to: Inha University. Downloaded on August 10,2023 at 06:07:49 UTC from IEEE Xplore. Restrictions apply.

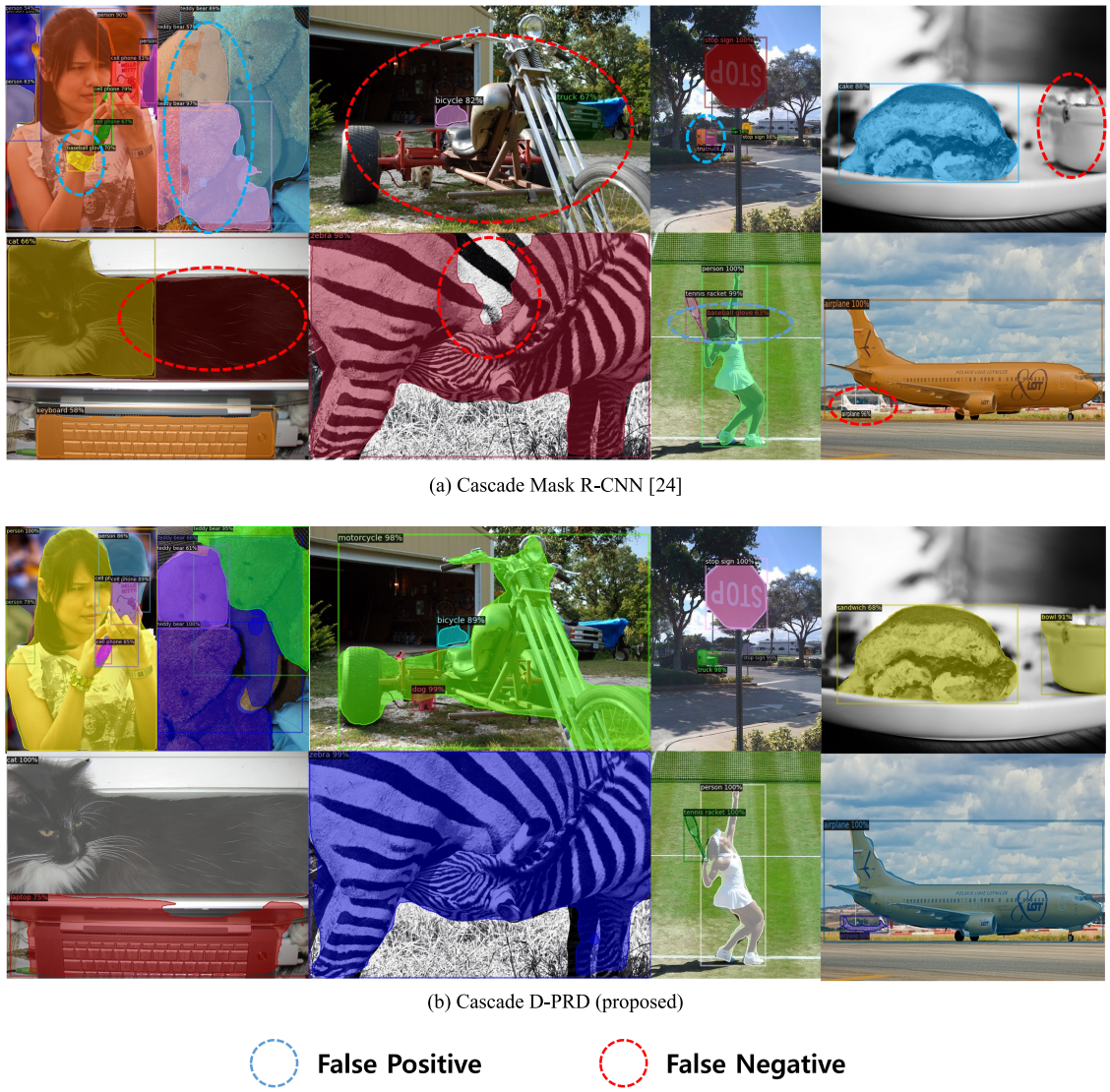


Fig. 8. Detection and instance segmentation results of the Cascade Mask R-CNN (top) and Cascade D-PRD with X101-FPN (bottom) are compared.

comparing the scores between Cascade D-PRD with X101-FPN. Moreover, our cascade detectors using the self-learning achieve the high box AP scores without self-learning on mask instances. These results indicate that our weak-supervision for learning deformable part models is not affected by the usage of the mask information.

Fig. 7 compares the accuracy and latency for the several cascade-based detectors: cascade R-CNN [24], HTC [25], SCNet [40], GCNet [94]. We also report the performance variation for the different backbones. We find out that there is a large accuracy improvement but the degradation of latency when applying the cascade scheme. In addition, the self-learning on the additional dataset leads to significant accuracy improvement. Our Cascade D-PRD consistently outperforms the recent cascade detectors [24], [25] with the lower latency. Compared to the SCNet [40] and GCNet [94], they show the better latency, but our detector achieves the much higher accuracy. In particular, our cascade D-PRD with the Swin-L backbone is much superior to other cascade detectors in terms of accuracy.

TABLE XI
COMPARISON OF DIFFERENT DETECTORS ON *VOC2007 TEST*. CASCADE R-CNN RESULTS CAN BE FOUNDED IN THIS CODE WEBSITE. THE FCOS, ATSS, PAA, AND IQET DETECTION RESULTS COME FROM [85]. SR-RCNN AND RETINA-SWN RESULTS ARE IN [99] AND [89]

Method	Backbone	AP	AP_{50}	AP_{75}
SR-RCNN [99]	ResNet-50	-	79.1	-
SR-RCNN [99]	ResNet-101	-	80.6	-
Cascade R-FCN [24]	R50-FPN	51.8	78.5	57.1
Cascade R-FNN [24]	R101-FPN	54.2	79.6	59.2
Retina-SWN [89]	R50-FPN	53.4	-	-
Retina-SWN [89]	X101-FPN	56.8	-	-
FCOS [27]	R50-FPN	56.2	79.9	61.9
ATSS [44]	R50-FPN	56.7	79.9	62.3
PAA [100]	R50-FPN	58.3	80.7	64.4
IQDET [85]	R50-FPN	59.0	81.4	65.0
Cascade D-PRD (ours)	R50-FPN	64.9	82.9	72.2
Cascade D-PRD (ours)	R101-FPN	65.0	83.0	71.9
Cascade D-PRD (ours)	X101-FPN	65.5	83.8	72.5

2) Pascal VOC: We evaluate our Cascade D-PRDs on the VOC07/12 datasets in order to show the generalization ability of our detectors. We maintain the overall structures of our detectors

Authorized licensed use limited to: Inha University. Downloaded on August 10,2023 at 06:07:49 UTC from IEEE Xplore. Restrictions apply.

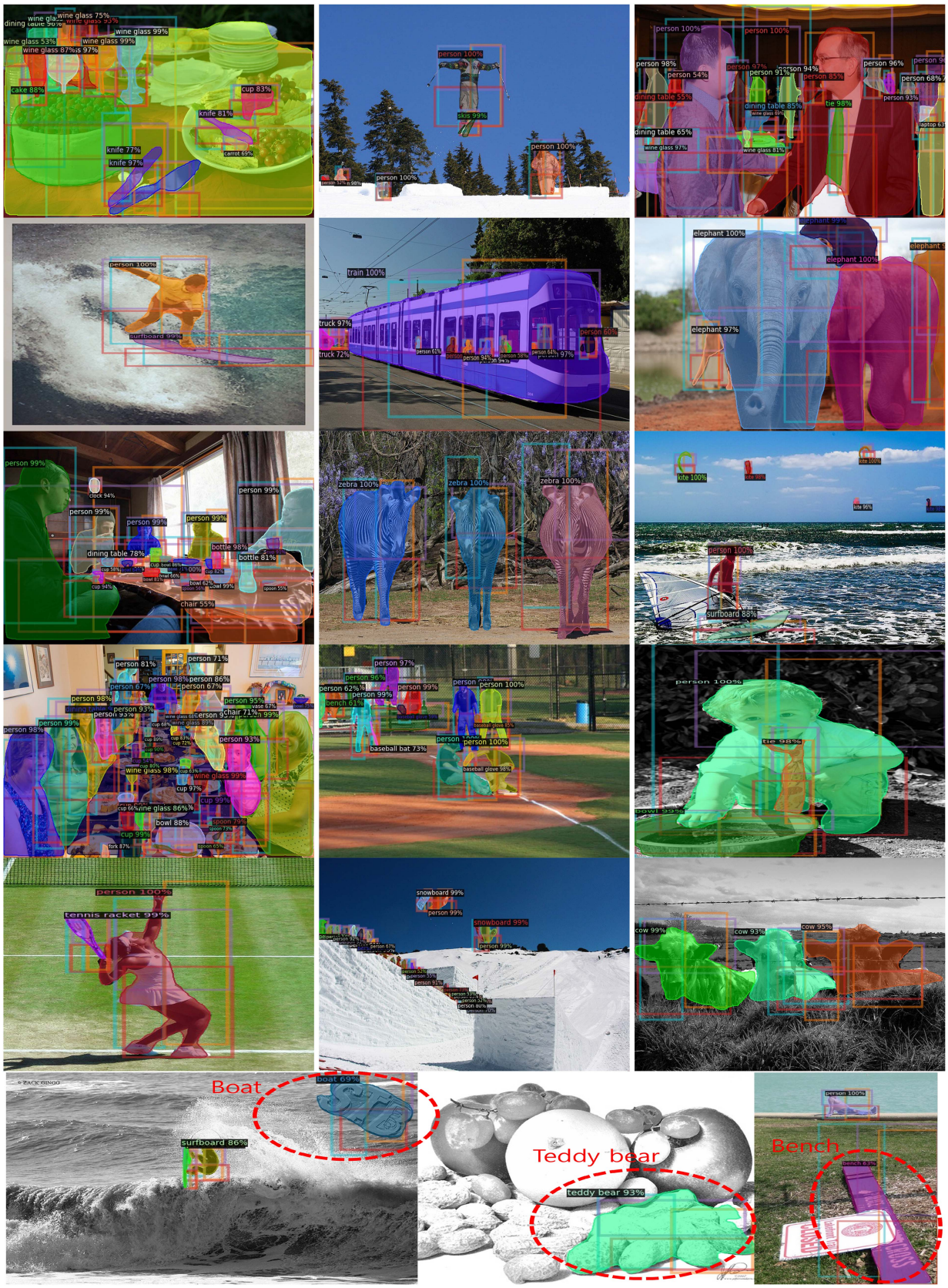


Fig. 9. Instance segmentation results on the COCO dataset using our Cascade D-PRD with X101-FPN. We mark a bounding box and mask with the same color for whole objects. The left, right, top, and bottom part boxes are colored with cyan, orange, purple, and red, respectively. The last row shows the examples of inaccurate detections.

Authorized licensed use limited to: Inha University. Downloaded on August 10,2023 at 06:07:49 UTC from IEEE Xplore. Restrictions apply.

for a fair comparison, and only change the training schedule.[6] For training and evaluation, we use the VOC07/12 trainval sets and VOC07 test set, respectively.

In Table XI, we compare ours with many recent detectors. We can achieve the best results with the R50-FPN backbone only even though using the deeper backbones produces the better detection scores. Compared to IQDET [85], our detector improve $\mathbf{AP}$, $\mathbf{AP_{50}}$, and $\mathbf{AP_{75}}$ by 5.95%, 1.47%, and 7.18% for the same R50-FPN. The significant improvements also prove the effects of the proposed methods.

I. Qualitative Results

We compare the detection results of our Cascade D-PRD with Cascade Mask R-CNN in Fig. 8. For easier comparison, we mark the false positives and false negatives with the colored and dotted circles. We know that the inaccurate detections occur in the Cascade Mask R-CNN for the small objects and occluded ones. On the other hand, our detector can handle the issues due to the powerful deformable part learning and feature aggregation methods.

Fig. 9 shows the visualization results. We also depict detected whole object regions with colored masks and its four-part regions with different color boxes. Although our object part regions do not correctly match with actual object parts, object part regions can be 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 the global context around the object.

In the last row in Fig. 9, we present some inaccurate detection results. As can be seen, these failures occurred due to the inaccurate classification rather than the localization problem. We expect that this problem is caused due to the global context learning based on the deformable part models. Specifically, since the context between an object and its background can be learned, our detector could predict the wrong classes as the boat, teddy bear, and bench by considering the relationship between the objects and the vicinity of the object.

VI. Conclusion

Detecting and segmenting object regions are still challenging problems due to occlusions and small object sizes. To resolve this, we present a deformable part model-based detector (D-PRD) for improving object detection. We propose a deformable part region network that can transform decomposed part regions according to the geometric transformation of an object. In order to learn discriminative semantic feature representation, we propose a feature aggregation tree. Based on the bottom-up feature aggregation starting from leaf nodes, our tree network can produce the strongest features at the top levels. Here, when aggregating the lower-level features, we exploit the spatial and channel attention for finding the more meaningful semantic features suitable for detection. We achieve this by maximizing the mutual information between the joint and marginal distributions for part features. For learning our D-PRD without extra supervision of part boxes, we present several detection losses for part models. By extending D-PRD to a multi-stage refinement scheme, we present a new cascade detector embedded with the deformable part region and feature aggregation tree networks that have shown remarkable detection results.

To verify our methods, we have implemented our D-PRDs with FAT and R-DAD (*i.e.* feature aggregation methods) and several backbones. We have compared our methods extensively with other recent detection methods on the several benchmark challenges such as PASCAL VOC and MSCOCO datasets. Without bells and whistles (*e.g.* multi-scale testing, model ensemble, etc), we have proved that our deformable detector is superior to other state-of-the-art detectors. When using the multi-scale testing, our Cascade D-PRD shows the best scores among many recent detectors. With the Swin-L backbone and self-learning [41], our Cascade D-PRD achieves the remarkable box AP scores of 57.9. Moreover, we have made the extensive ablation study to show effectiveness and robustness of our methods. To this end, we have implemented several versions of D-PRDs and Cascade D-PRDs by adding our method one-by-one or changing structures, model components, loss functions, etc. We believe that our deformable part learning and feature aggregation methods would become the crucial guideline for deformable part learning of modern detectors.

[6] Followed by the default setting provided by Detectron2.

References

[1] P. Dollár, R. Appel, S. J. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.

[2] P. F. Felzenszwalb, R. B. Girshick, and D. A. McAllester, "Cascade object detection with deformable part models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2241–2248.

[3] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Efficient subwindow search: A branch and bound framework for object localization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2129–2142, Dec. 2009.

[4] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.

[5] J. Pont-Tuset, P. Arbelaez, J. T. Barron, F. Marqués, and J. Malik, "Multiscale combinatorial grouping for image segmentation and object proposal generation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 128–140, Jan. 2017.

[6] J. Carreira and C. Sminchisescu, "CPMC: Automatic object segmentation using constrained parametric min-cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1312–1328, Jul. 2012.

[7] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 391–405.

[8] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2189–2202, Nov. 2012.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–14.

[10] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[12] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1106–1114.

Authorized licensed use limited to: Inha University. Downloaded on August 10,2023 at 06:07:49 UTC from IEEE Xplore. Restrictions apply.

[14] R. B. Girshick, "Fast R-CNN," in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[15] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[16] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[17] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[18] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 9157–9166.

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

[20] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 2999–3007.

[21] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 734–750.

[22] J. Dai et al., "Deformable convolutional networks," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 764–773.

[23] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[24] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6154–6162.

[25] K. Chen et al., "Hybrid task cascade for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4974–4983.

[26] X. Lu, B. Li, Y. Yue, Q. Li, and J. Yan, "Grid R-CNN," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7363–7372.

[27] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 9627–9636.

[28] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 936–944.

[29] C. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD : Deconvolutional single shot detector," 2017, *arXiv:1701.06659*.

[30] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8759–8768.

[31] Q. Zhao et al., "M2Det: A single-shot object detector based on multi-level feature pyramid network," in *Proc. Conf. Assoc. Advance. Artif. Intell.*, 2019, pp. 9259–9266.

[32] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "NAS-FPN: Learning scalable feature pyramid architecture for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7036–7045.

[33] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10781–10790.

[34] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–16.

[35] C. Guo, B. Fan, Q. Zhang, S. Xiang, and C. Pan, "AugFPN: Improving multi-scale feature learning for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12592–12601.

[36] R. B. Girshick, F. N. Iandola, T. Darrell, and J. Malik, "Deformable part models are convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 437–446.

[37] L. Wan, D. Eigen, and R. Fergus, "End-to-end integration of a convolutional network, deformable parts model and non-maximum suppression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 851–859.

[38] H. Zhang, H. Chang, B. Ma, S. Shan, and X. Chen, "Cascade RetinaNet: Maintaining consistency for single-stage object detection," in *Proc. Brit. Mach. Vis. Conf.*, 2019, Art. no. 227.

[39] Y. Fang et al., "Instances as queries," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 6910–6919.

[40] T. Vu, K. Haeyong, and C. D. Yoo, "SCNet: Training inference sample consistency for instance segmentation," in *Proc. Conf. Assoc. Advance. Artif. Intell.*, 2021, pp. 2701–2709.

[41] B. Zoph et al., "Rethinking pre-training and self-training," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 3833–3845.

[42] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask Scoring R-CNN," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6409–6418.

[43] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[44] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9759–9768.

[45] S. Zagoruyko et al., "A multipath network for object detection," in *Proc. Brit. Mach. Vis. Conf.*, pp. 1–12.

[46] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 354–370.

[47] T. Vu, H. Jang, T. X. Pham, and C. D. Yoo, "Cascade RPN: Delving into high-quality region proposal network with adaptive convolution," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 1430–1440.

[48] H. K. Cheng, J. Chung, Y.-W. Tai, and C.-K. Tang, "CascadePSP: Toward class-agnostic and very high-resolution segmentation via global and local refinement," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8890–8899.

[49] S. Li, L. Ke, K. Pratama, Y. Tai, C. Tang, and K. Cheng, "Cascaded deep monocular 3D human pose estimation with evolutionary training data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6172–6182.

[50] T. Zhou, W. Wang, S. Qi, H. Ling, and J. Shen, "Cascaded human-object interaction recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4262–4271.

[51] S.-H. Lee and S.-H. Bae, "AFI-GAN: Improving feature interpolation of feature pyramid networks via adversarial training for object detection," *Pattern Recognit.*, vol. 138, 2023, Art. no. 109365.

[52] S. Kim, H. Kook, J. Sun, M. Kang, and S. Ko, "Parallel feature pyramid network for object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 239–256.

[53] S. Woo, J. Park, J. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.

[54] Y. Lee and J. Park, "CenterMask: Real-time anchor-free instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13906–13915.

[55] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-YOLOv4: Scaling cross stage partial network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 13029–13038.

[56] Z. Huang, W. Ke, and D. Huang, "Improving object detection with inverted attention," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 1–5, 2020, pp. 1294–1302.

[57] T.-I. Hsieh, Y.-C. Lo, H.-T. Chen, and T.-L. Liu, "One-shot object detection with co-attention and co-excitation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 2721–2730.

[58] Q. Fan, W. Zhuo, C.-K. Tang, and Y.-W. Tai, "Few-shot object detection with attention-RPN and multi-relation detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4013–4022.

[59] Y. Gu, L. Wang, Z. Wang, Y. Liu, M.-M. Cheng, and S.-P. Lu, "Pyramid constrained self-attention network for fast video salient object detection," in *Proc. Conf. Assoc. Advance. Artif. Intell.*, 2020, pp. 10869–10876.

[60] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.

[61] Z. Sun, S. Cao, Y. Yang, and K. M. Kitani, "Rethinking transformer-based set prediction for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3611–3620.

[62] Z. Dai, B. Cai, Y. Lin, and J. Chen, "UP-DETR: Unsupervised pre-training for object detection with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1601–1610.

[63] L. Zhang and K. Ma, "Improve object detection with feature-based knowledge distillation: Towards accurate and efficient detectors," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–14.

[64] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 1–9.

[65] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[66] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.

Authorized licensed use limited to: Inha University. Downloaded on August 10,2023 at 06:07:49 UTC from IEEE Xplore. Restrictions apply.

[67] C.-H. Lin and S. Lucey, "Inverse compositional spatial transformer networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2252–2260.
[68] T. Yang, X. Zhang, Z. Li, W. Zhang, and J. Sun, "MetaAnchor: Learning to detect objects with customized anchors," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 318–328.
[69] W. Ke, T. Zhang, Z. Huang, Q. Ye, J. Liu, and D. Huang, "Multiple anchor learning for visual object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10206–10215.
[70] M. I. Belghazi et al., "Mutual information neural estimation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 531–540.
[71] S. Nowozin, B. Cseke, and R. Tomioka, "F-GAN: Training generative neural samplers using variational divergence minimization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 271–279.
[72] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.
[73] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.
[74] Z. Chen, X. Jin, B. Zhao, X. Wei, and Y. Guo, "Hierarchical context embedding for region-based object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 633–648.
[75] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Occlusion-aware R-CNN: Detecting pedestrians in a crowd," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 657–674.
[76] C. Zhou and J. Yuan, "Bi-box regression for pedestrian detection and occlusion estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 138–154.
[77] S.-H. Bae, "Object detection based on region decomposition and assembly," in *Proc. Conf. Assoc. Advance. Artif. Intell.*, 2019, pp. 8094–8101.
[78] T. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[79] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2015.
[80] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5987–5995.
[81] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," 2019. [Online]. Available: https://github.com/facebookresearch/detectron2
[82] R. D. Hjelm et al., "Learning deep representations by mutual information estimation and maximization," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–24.
[83] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 618–626.
[84] N. Samet, S. Hicsonmez, and E. Akbas, "HoughNet: Integrating near and long-range evidence for bottom-up object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 406–423.
[85] Y. Ma, S. Liu, Z. Li, and J. Sun, "IQDet: Instance-wise quality distribution sampling for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1717–1725.
[86] W. Ke, T. Zhang, Z. Huang, Q. Ye, J. Liu, and D. Huang, "Multiple anchor learning for visual object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10206–10215.
[87] Z. Dong, G. Li, Y. Liao, F. Wang, P. Ren, and C. Qian, "CentripetalNet: Pursuing high-quality keypoint pairs for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10519–10528.
[88] C. Zhu, F. Chen, Z. Shen, and M. Savvides, "Soft anchor-point object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 91–107.
[89] Q. Cai, Y. Pan, Y. Wang, J. Liu, T. Yao, and T. Mei, "Learning a unified sample weighting network for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 14161–14170.
[90] L. Chen, A. Hermans, G. Papandreou, F. Schroff, P. Wang, and H. Adam, "MaskLab: Instance segmentation by refining object detection with semantic and direction features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4013–4022.
[91] H. Zhang, H. Chang, B. Ma, N. Wang, and X. Chen, "Dynamic R-CNN: Towards high quality object detection via dynamic training," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 260–275.
[92] J. Cao, H. Cholakkal, R. M. Anwer, F. S. Khan, Y. Pang, and L. Shao, "D2Det: Towards high quality object detection and instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11485–11494.
[93] S.-H. Bae, "Deformable part region learning for object detection," in *Proc. Conf. Assoc. Advance. Artif. Intell.*, 2022, pp. 95–103.
[94] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "Global context networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Dec. 24, 2020, doi: 10.1109/TPAMI.2020.3047209.
[95] G. Ghiasi et al., "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2918–2928.
[96] X. Zhou, V. Koltun, and P. Krähenbühl, "Probabilistic two-stage detection," 2021, *arXiv:2103.07461*.
[97] H. Zhang et al., "DINO: DETR with improved denoising anchor boxes for end-to-end object detection," in *Proc. Int. Conf. Learn. Representations*, 2023, pp. 1–19.
[98] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.
[99] J. Noh, W. Bae, W. Lee, J. Seo, and G. Kim, "Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 9725–9734.
[100] K. Kim and H. S. Lee, "Probabilistic anchor assignment with IoU prediction for object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 355–371.

Seung-Hwan Bae (Member, IEEE) received the BS degree in information and communication engineering from Chungbuk National University, in 2009 and the MS and PhD degrees in information and communications from the Gwangju Institute of Science and Technology (GIST), in 2010 and 2015, respectively. He was a senior researcher with Electronics and Telecommunications Research Institute (ETRI) in Korea from 2015 to 2017. He was an assistant professor with the Department of Computer Science and Engineering at Incheon National University, Korea from 2017 to 2020. He is currently an associate professor with the Department of Computer Engineering, Inha University. His research interests include object tracking, object detection, generative model learning, continual learning, on-device ML, etc.

Authorized licensed use limited to: Inha University. Downloaded on August 10,2023 at 06:07:49 UTC from IEEE Xplore. Restrictions apply.