Hindawi
Mobile Information Systems
Volume 2022, Article ID 7556022, 11 pages
https://doi.org/10.1155/2022/7556022

Research Article

Curvature-Driven Deformable Convolutional Networks for End-To-End Object Detection

Xiaodong Gu [1] **and Ying Fu** [2]

[1]*School of Mathematics and Information Technology, Jiangsu Second Normal University, NanJing 210 013, China*
[2]*School of Foreign Languages, Jiangsu Second Normal University, NanJing 210 013, China*

Correspondence should be addressed to Xiaodong Gu; gu3xuan@qq.com

Received 9 September 2021; Revised 23 December 2021; Accepted 17 January 2022; Published 4 February 2022

Academic Editor: Jose M. Barcelo-Ordinas

Recently, deformable convolution networks have shown the superior performance in object detection due to its ability to adapt to the geometric variations of object. These methods learn the offset fields under the supervision of localization and recognition. Nevertheless, the spatial support of these networks may be inexact because the offsets are learned implicitly via extra convolutional layer. In this work, we present curvature-driven deformable convolutional networks (C-DCNets) that adopt explicit geometric property of the preceding feature maps to enhance the deformability of convolution operation and make the networks easier to focus on pertinent image region. To be consistent with postprocessing technology of object detection, we multiply the class prediction probability by the similarity of predicted boxes and ground truth boxes as the final class prediction probability and substitute it into the binary cross entropy loss function. The obtained loss function correlates the bounding box regression and classification. Experimental results on PASCAL VOC and COCO data set show that C-DCNets-based YOLOv4 with the proposed loss function outperforms state-of-the-art algorithms.

1. Introduction

Attention mechanisms make a neural network pay more attention to relevant parts of the image than irrelevant parts. Therefore, they can model long-range dependencies. Spatial transformer module [1] is a dynamic mechanism, which can actively spatially transform an image (or a feature map) to enhance the representations produced by CNNs. "Squeeze-and-Excitation Networks" (SENet) [2] improve the network representation by explicitly modeling the interdependencies between the channels of network's convolutional features. "Convolutional Block Attention Module" (CBAM) [3] applies channel attention modules and spatial attention modules sequentially so that each branch can learn "what to focus" and "where to focus" on the channel axis and spatial axis, respectively. "Selective Kernel Networks" (SKNets) [4] focus on the adaptive receptive field (RF) size of neurons by introducing the attention mechanisms. As a particular instantiation of spatial attention mechanisms [5–8], deformable convolutional networks can capture spatial transformation since it is utilized to exploit query content and relative position effectively. The current state-of-the-art methods for modeling geometric transformations are Deformable Convolutional Networks (DCNv1) [6], Deformable ConvNets v2 [7], and Point Set Representation for object detection (RepPoints) [8]. In DCNv1, there are two modules that aid CNNs in modeling geometric variations. One is deformable convolution, in which the grid sampling positions of standard convolution are shifted by 2D offsets learned via extra convolutional layer. The other is deformable RoIpooling, which adds 2D offsets to each bin position in the regular bin partition of previous RoI pooling [6]. The incorporation of these modules into a neural network gives it the ability to adjust its feature representation to object configuration, specifically by deforming network's sampling and pooling patterns to fit the object's structure. In DCNv2 [7], the learned offset fields and modulated amplitude control the sampling position together. However,

their spatial support may exceed the region of interest because the offsets and modulation scalar are learned implicitly by additional convolutional layer. In RepPoints [8], the point distance loss and the object recognition loss are adopted to learn the object localization, as deformable convolutions are operated on an irregular-form grid points and its recognition feedback can guide training for the positioning of these points. Compared with DCNv1 and DCNv2, RepPoints have more constraints on classification module, but its offset fields are still learned implicitly by convolutional layer. In order to further improve the deformation ability of Deformable Convolutional Networks, we introduce the intrinsic geometric property of the input feature maps, and a curvature-driven deformable convolutional networks (C-DCNets) are proposed, which use the offset learning guided by curvature fields of the preceding feature maps to focus the network on pertinent image region. The proposed method produces leading results on PASCAL VOC and COCO data set for object detection.

The goal of object detection is to predict a set of bounding boxes and category labels for each object of interest. But there are many near-duplicate predictions because of the anchor sets and the heuristics that cast target boxes to anchors. Traditional object detection pipelines [9, 10] assign foreground/background scores of each class for multiscale sliding windows based on the features calculated in each window. And deep-learning based object detectors employ region proposals [11, 12] generated by convolutional neural networks to replace sliding windows. For deep-learning based one-stage detectors (e.g., SSD [13], YOLOv1 ~ YOLOv4 [14–17]), they also use nonunique assignment rules between ground truth boxes and prediction boxes even if there are no region proposals. Hence, almost all state-of-the-art detectors [12–19] need postprocessing. Besides, considering the imbalance of positive and negative samples, feature imbalance, target imbalance, and image scene imbalance in target detection, researchers [20, 21] propose some preprocessing methods similar to sample augmentation to obtain a balanced learning representation. Nonmaximum suppression (NMS) [9–12] is a post-processing part of the object detection framework to avoid near-identical boxes. Its evaluation score is the product of Intersection over Union (IoU) and the class prediction probability, but these two loss functions are used in training to deal with box regression and classification separately. To be consistent with NMS, we multiply the class prediction probability by IoU of predicted boxes and targets as the final class prediction distribution and substitute it into the binary cross entropy loss function. The obtained loss function can achieve the best bipartite matching between the predicted boxes and ground truth boxes. The main contributions of this work are summarized as follows:

(1) Curvature-driven deformable convolutional networks (C-DCNets) are proposed, which make the spatial support of the networks adapt much more to saliency region

(2) A new loss function associated with bounding box regression and classification is proposed, in which the class prediction probability in the binary cross entropy loss function integrates the similarity of predicted boxes and targets

(3) We evaluate a C-DCNets based detection frameworks with the proposed loss function on PASCAL VOC and COCO data set, against a very competitive Faster R-CNN [12], YOLOv4 [17], DETR [22], and deformable DETR [23] baseline

The rest of this paper is organized as follows: In Section 2, the related works of attention mechanisms and post-processing techniques are reviewed. In Section 3, a curvature-driven deformable convolutional networks and a loss function associated with bounding box regression and classification are explained. In Section 4, the experimental results are given. Finally, Section 5 concludes this paper.

2. Related Work

2.1. Attention Mechanisms. Attention mechanisms are first studied in natural language processing (NLP) [24–28], where encoder-decoder attention modules are developed to facilitate neural machine translation. Certain key elements are given priority according to a given query to calculate the output for the query element. And then, self-attention modules are utilized for modeling intrasentence relations. When assigning the attention weight to a certain key for a given query, we need to consider the content of the query and the content of the key. The query content may be the features of a word in a sentence, and a key may be another word within the sentence. Besides, the relative position of the query and key should be considered. Shortly afterwards, attention mechanisms are becoming popular in computer vision [29–32]. Some works have even applied the attention mechanisms to SAR image [33–37]. In particular, [35] applies channel attention modules to ship classification in SAR images. Reference [36] combines traditional hand-crafted HOG features and CNN features to improve classification accuracy. A polarization fusion network with geometric feature embedding (PFGFE-Net) [37] is proposed for SAR ship classification. References [30–32] successfully extend relation networks and attention modules to the image domain, and a long-range object-object and pixel-pixel relations are modeled. Reference [30] establishes the relationship between objects through interaction of their appearance feature and geometry. In [31], the response of a pixel is calculated as a weighted sum of all pixel features. Reference [32] proposes a learnable region feature extractor, and the previous region feature extraction modules are unified from the perspective of the pixel-object relations. A common problem with such methods is that the aggregation weights and the aggregation operation need to be calculated on the elements in a pairwise fashion, which brings a high cost as the amount of calculation is quadratic to the number of elements. Different with the huge amount of calculation [6, 30–32], [7, 8] can be perceived as a special attention mechanism where only a sparse set of elements have nonzero aggregation weights. According to Zhu's [5] distinction of different attention factors based on how to obtain the

attention weight for a key considering that a query is determined, Deformable Convolutional Networks [6–8] utilize an attention mechanism based on the query content and relative position term; they operate more effectively and efficiently on object detection and semantic segmentation. The attended elements are specified by the learnable offsets [6] and the computational overhead is just linear to the number of elements. Modulation scalar is further introduced in [7]. RepPoints [8] is an object detection method that simultaneously models fine-grained localization information and identifies local areas significant for object classification. RepPoints can learn a geometric representation of objects. However, its deformability also depends only on implicit learning through an additional convolutional layer, and its spatial support may exceed the region of interest. To strength the deformability of convolution operation under irregular sampling grid, we propose curvature-driven deformable convolutional networks (C-DCNets) based on explicit geometric property of the preceding feature maps, where the curvature fields are utilized to guide the offsets learning, and the proposed C-DCNets modules are learned under the supervision of loss function that correlates the position accuracy and the class prediction probability.

2.2. Postprocessing Technology of the Object Detector. Most deep learning-based detectors use postprocessings such as nonmaximal suppression (NMS) to avoid near-duplicates boxes. The original NMS does not consider the context information. Greedy NMS [38] performs from high confidence score to low confidence score. Soft NMS [39] solves the problem of confidence score degradation caused by object occlusion. The DIoU NMS [40] adds the information of the center point distance to the bounding box screening process on the basis of Soft NMS. Learnable NMS methods [41] and relation networks [30] explicitly model relations between different prediction boxes with attention. In Fast NMS [42], each instance can decide to keep or discard in parallel, but it removes slightly too many boxes.Some algorithms use a global inference schemes to model interactions between all predicted bounding boxes. For constant-size set prediction, [43] uses deep neural networks to predict a set of class-agnostic bounding boxes along with a single score for each box. Reference [44] uses recurrent neural networks. End-to-end object detection with transformer (DETR) [22] is the first combination of bipartite matching loss and transformers with parallel decoding. And it uses Hungarian algorithm [45] to find a bipartite matching between prediction boxes and ground truth boxes, which enforces permutation-invariance, and guarantees that each target box has a unique match. However, it suffers from computational complexity and low performance of small object detection. In [23], Deformable transformer (Deformable DETR) is proposed for end-to-end object detection, whose attention modules only concern a small set of key sampling points around a reference point. It combines the advantage of the sparse spatial sampling of deformable convolution and the relation modeling capability of transformers. For each query, multiscale deformable attention checks multiple sampling points from multiscale inputs. It has superior performance in small object detection without the help of FPN [46]. However, due to the lack of global inference mode, deformable DETR still uses traditional NMS to improve its performance. And the complexity of the deformable transformer is very high when the number of object queries is large. No matter which kind of NMS, its evaluation score is the product of class prediction probability and IoU.

3. Curvature-Driven Deformable Convolutional Networks for End-To-End Object Detection

3.1. Curvature-Driven Deformable Convolutional Networks. Geometric priors [47–49] play an essential role in Bayesian theory. Gradient priors and curvature flow are widely used in image denoising, restoration, super resolution, and other fields. Gradient reflects the first derivative of the image, which is easy to be affected by noise, and curvature describes the degree of curvature of an image levelset, which reflects the change of the first derivative. Compared with the gradient information, curvature fields reflect the trend of sampling points towards the salient region of the image. In this paper, we apply the curvature of the preceding feature maps to the learnable offset to obtain the final offset; the larger the curvature, the larger the final displacement.

3.2. Curvature-Driven Deformable Convolution. The definition of a curvature of an image levelset is as follows:

$$\begin{aligned}&\operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\\&=\left(\frac{\partial}{\delta x}\cdot\vec{i}+\frac{\partial}{\delta y}\cdot\vec{j}\right)\left(\overline{\phi}_x\cdot\vec{i}+\overline{\phi}_y\cdot\vec{j}\right),\end{aligned}\tag{1}$$

and the curvature vector is

$$\text{Curv}=\overline{\phi}_{xx}\vec{i}+\overline{\phi}_{xx}\vec{j},\tag{2}$$

where

$$\begin{aligned}\overline{\phi}_{xx}&=\frac{|\nabla\phi|^2\phi_{xx}-\left(\phi_x\phi_{xx}+\phi_y\phi_{yx}\right)\phi_x}{|\nabla\phi|^3},\\\overline{\phi}_{yy}&=\frac{|\nabla\phi|^2\phi_{yy}-\left(\phi_y\phi_{yy}+\phi_x\phi_{xy}\right)\phi_x}{|\nabla\phi|^3}.\end{aligned}\tag{3}$$

Given a convolutional kernel of K sampling locations, let w_k and p_k denote the weight and prespecified offset for the k-th location, respectively. For example, $K=9$ and $p_k\in(-1,-1),(-1,0)\ldots(1,1)$ defines a 3×3 convolutional kernel of dilation 1. Let $\mathbf{X}(p)$ and $\mathbf{Y}(p)$ denote the features at location p from the input feature maps $\mathbf{X}$ and output feature maps $\mathbf{Y}$, respectively. The curvature-driven deformable convolution can be expressed as

$$\mathbf{Y}(p) = \sum_{k=1}^{K} w_k \cdot \mathbf{X}\left(p + p_k + \triangle p_k \cdot \mathrm{Curv}_k\right), \tag{4}$$

where $\triangle p_k$ is the learnable offset for the k-th location and Curv_k is the curvature vector at the k-th location. We use bilinear interpolation to compute $\mathbf{X}(p + p_k + \triangle p_k \cdot \mathrm{Curv}_k)$:

$$\mathbf{X}\left(p + p_k + \triangle p_k \cdot \mathrm{Curv}_k\right) = \sum_{q} G\left(q, p + p_k + \triangle p_k \cdot \mathrm{Curv}_k\right) \cdot X(q), \tag{5}$$

where q enumerates all integral spatial locations in the feature maps X, and $G(q, p + p_k + \triangle p_k \cdot \mathrm{Curv}_k)$ is the two-dimension bilinear interpolation kernel:

$$\begin{aligned} &G\left(q, p + p_k + \triangle p_k \cdot \mathrm{Curv}_k\right) = \\ &g\left(q_x, \left(p + p_k + \triangle p_k \cdot \mathrm{Curv}_k\right)_x\right) \cdot \\ &g\left(q_y, \left(p + p_k + \triangle p_k \cdot \mathrm{Curv}_k\right)_y\right), \end{aligned} \tag{6}$$

where $G(a, b) = \max(0, 1 - |a - b|)$. The curvature fields are generated from the input feature maps $\mathbf{X}$ and $\triangle p_k$ is obtained via a convolution layer applied over the same feature maps. The convolution kernel is of the same spatial resolution and dilation as those of the current convolutional layer. The output offset fields have the same spatial resolution with the input feature maps, and the product of point-by-point multiplication of offset fields and curvature fields will be superimposed to the normal grid sampling positions in the standard convolution. The channel dimension $2K$ corresponds to K offsets $\{\triangle p_k\}_{k=1}^{K}$. During training, both the convolutional kernels and the offsets are learned simultaneously. To learn offsets, the gradients are backpropagated through the bilinear operations equations (5) and (6). The added convolution layer and fully connection layer for offset learning are initialized with zero weights. Their learning rates are set to 0.1 times of the learning rate for the existing layers.

Figure 1 shows the 3×3 deformable convolution. Figure 2 shows the sampling locations ($9 = 729$ red points in each image) in three levels of 3×3 deformable filters for activation units (green points). The receptive field and the sampling locations in the standard convolution are fixed no matter how many levels convolution are stacked. The sampling locations in DCNv1 and DCNV2 (shown in the middle) are adaptively adjusted according to the scale and shape of the object. The normalized modulation amplitudes in DCNv2 are obtained by additional convolutional layers learning; therefore, the offsets and modulation scalar in dcnv2 are entangled with each other. Compared with DCNv1, there is no big difference in the change of sampling positions because that the modulation scalar acts on the whole convolution term. The sampling locations in our C-DCNets are shown at the bottom. It can be seen from the figure that the sampling locations in our C-DCNets are more concentrated in the salient region of the image.

3.3. Curvature-Driven Deformable RoI Pooling. Given the input feature map $\mathbf{X}$ and a RoI of size $w \times h$ and top-left corner p_0, RoI pooling divides the RoI into $k \times k$ (k is a free parameter) bins and outputs a $k \times k$ feature map $\mathbf{Y}$. For $(i, j) - th$ bin $0 \le i, j \le k$, we have

$$\mathbf{Y} = \sum_{p \in bin(i,j)} \frac{X\left(p_0 + p_{i,j}\right)}{n_{i,j}}, \tag{7}$$

where $n_{i,j}$ is the number of pixels in the bin. In curvature-driven deformable ROI pooling, offsets $\{\triangle p_{i,j} \cdot \mathrm{Curv}_{i,j} | 0 \le i, j \le k\}$ are added to the spatial binning positions.

$$\mathbf{Y} = \sum_{p \in bin(i,j)} \frac{\mathbf{X}\left(p_0 + p_{i,j} + \triangle p_{i,j} \cdot \mathrm{Curv}_{i,j}\right)}{n_{i,j}}. \tag{8}$$

Equation (8) is implemented by bilinear interpolation equations (5) and (6). Figure 3 shows the process of obtaining offsets. First, RoI pooling equation (7) generates the pooled feature maps. Second, curvature fields can be obtained from the input feature maps, and a fully connection layer generates the normalized offsets $\triangle \widehat{p}_{i,j}$, which are then transformed to the offsets $\Delta p_{i,j}$ in (8) by element-wise product with the $ROI's$ width and height, as $\triangle p_{i,j} = \triangle \widehat{p}_{i,j} \odot (w, h)$. The effect of curvature-driven deformable RoI pooling is shown in Figure 4. The regular grid structure in the standard RoI pooling will no longer be maintained, and the deformation ability of the sampled grid will be enhanced.

3.4. Loss Function Associated with Bounding Box Regression and Classification. Traditional object detection pipelines employ nonmaximum suppression (NMS) for selecting the best prediction bounding box with the maximum score and remove spurious neighboring detection boxes. First, it sorts all detection boxes on the basis of their scores. The detection box M with the maximum score is selected and all other detection boxes with a significant overlap (using a predefined threshold) with M are suppressed. The evaluation score of NMS is the product of class prediction probability and IoU. But the training loss function in object detection networks uses the loss of position accuracy for bounding box regression and the loss of classification for recognition separately, which is not consistent with the evaluation score of NMS.A linear combination of the l_1 loss and the generalized IoU loss [50],

$$L_{\mathrm{box}}\left(b_{ij}, \widehat{b}\right) = \xi_{\mathrm{iou}} L_{\mathrm{iou}}\left(b_{ij}, \widehat{b}\right) + \xi_{L_1} \left\| b_{ij} - \widehat{b} \right\|_1, \tag{9}$$

is used in DETR, where $\xi_{iou}, \xi_{L_1} \in \mathbb{R}$ are hyperparameters. These two losses are normalized by the number of objects inside the batch. The generalized IoU loss $L_{\mathrm{iou}}(b_{ij}, \widehat{b}) = [1 - (\mathrm{IoU} - (B_{(b_{ij}, \widehat{b})} / (b_{ij} \cup \widehat{b}))/|b_{ij}, \widehat{b}|)]$, where $\mathrm{IoU} = |b_{ij} \cap \widehat{b}| / |b_{ij} \cup \widehat{b}|$, $|\cdot|$ means area. The union and intersection of box coordinates are used as shorthands for the boxes themselves. The areas of unions or intersections are

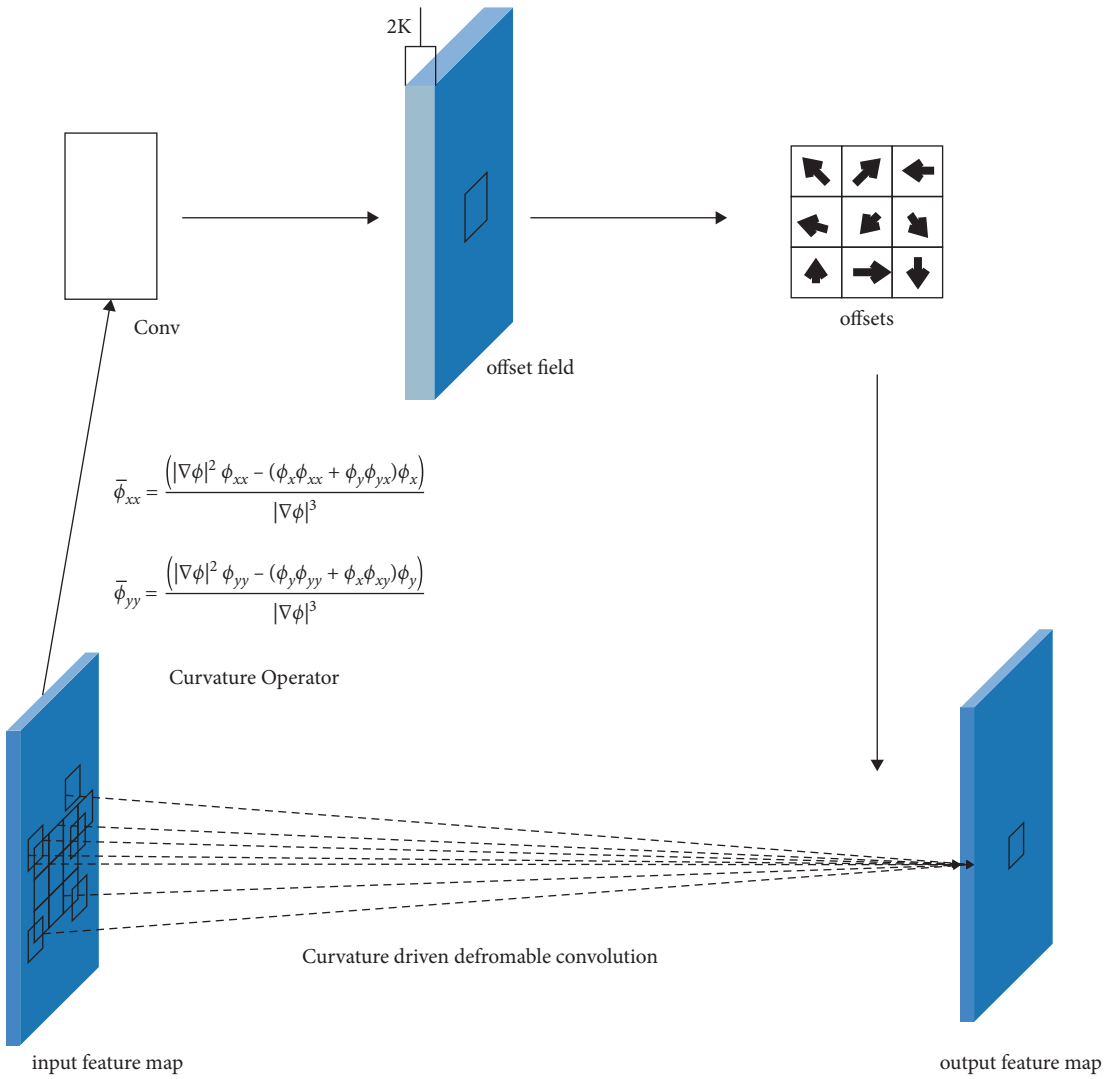

Figure 1: Illustration of 3×3 curvature-driven deformable convolution.

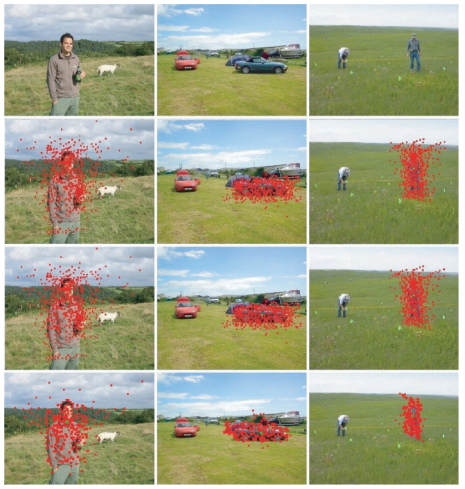

Figure 2: The sampling locations ($93 = 729$ red points in each image) in three levels of 3×3 deformable filters for activation units (green points). Top: original image; the second row: DCNv1; the third row: DCNv2; bottom: C-DCNets.

computed by min/max of the linear functions of b_{ij} and $\widehat{b}$, which makes the loss sufficiently well behaved for stochastic gradients. $B(b_{ij}, \widehat{b})$ means the largest box containing b_{ij} and $\widehat{b}$ (the areas involving B are also computed based on min/max of linear functions of the box coordinates). Different with DETR, our problem is not N-to-N matching. Generally speaking, the number of prediction bounding boxes is much larger than the number of ground truth boxes. For YOLOv3/YOLOv4, the last several convolutional layers predict 3-d tensor $S \times S \times [3 * (4 + 1 + C)]$ where the input image is divided into $S \times S$ grid and C is the classes number.To correlate the bounding box regression and classification, we multiply the class prediction probability by IoU of predicted boxes and ground truth boxes as the final class prediction distribution and substitute it into the binary cross entropy loss function:

$$-\sum_{i=0}^{S^2} \mathbb{I}_{ij}^{Obj} \sum_{c \in \text{classes}} \left[\text{IoU} * \widehat{P}_i^j \log\left(P_i^j\right) + \left(1 - \text{IoU} * \widehat{P}_i^j\right) \log\left(1 - P_i^j\right) \right], \tag{10}$$

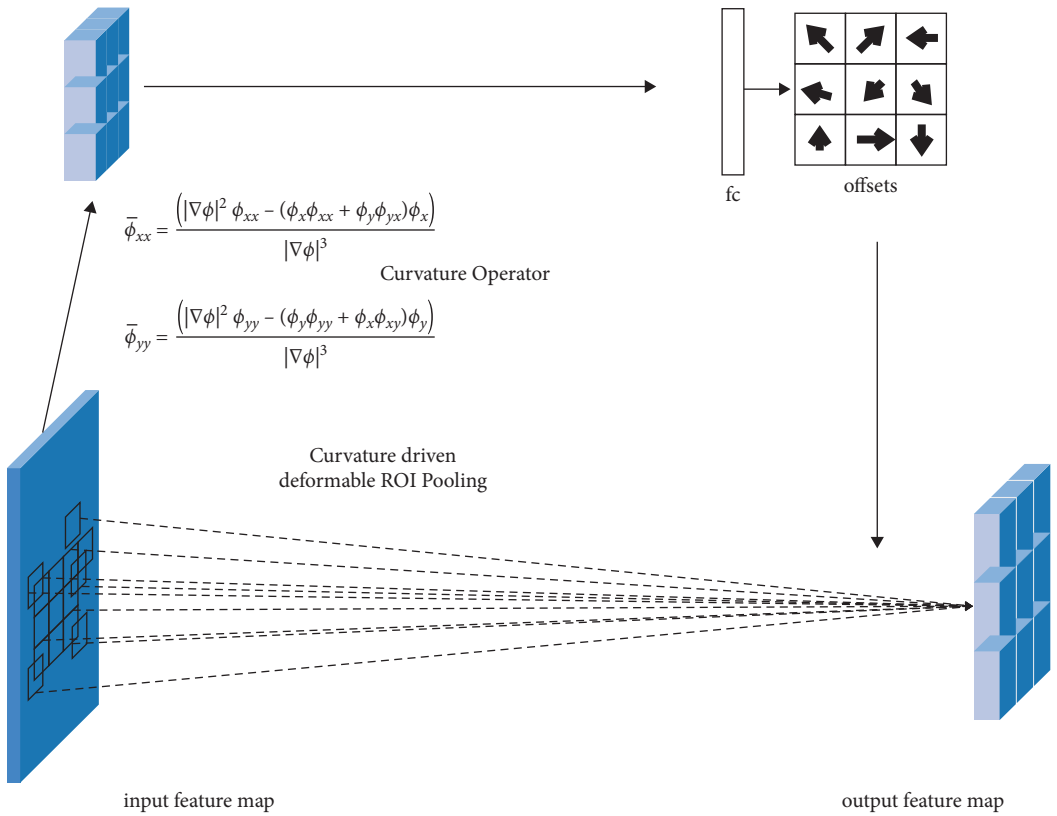

Figure 3: Illustration of 3×3 curvature-driven deformable RoI pooling.

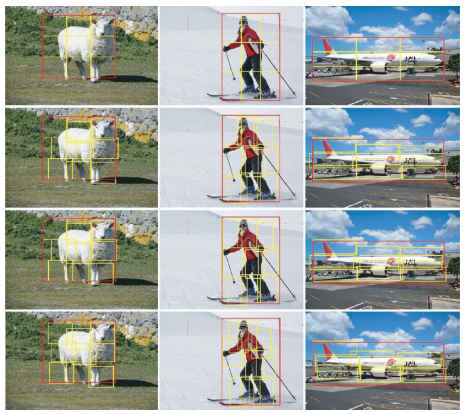

Figure 4: Illustration of offset parts in (curvature-driven) deformable RoI pooling in R-FCN [11] and 3×3 binsm(yellow) for an input RoI (red). Top: original RoI; the second row: DCNv1; the third row: DCNv2; bottom: our C-DCNets.

where $\mathbb{I}_{ij}^{Obj}$ denotes if object appears in cell i and $\mathbb{I}_{ij}^{Obj}$ denotes that the jth bounding box predictor in cell i is responsible for that prediction. We replace the original binary cross entropy loss function in YOLOv4 with (10) and original $L_{iou}(b_{ij}, \widehat{b})$ with $L_{box}(b_{ij}, \widehat{b})$; therefore the whole loss function in YOLOv4 is

$$\mathbf{L}_{\text{box}}\left(b_{ij},\widehat{b}\right)-\sum_{i=0}^{S^2}\mathbb{I}_{ij}^{\text{Obj}}\sum_{c\in\text{classes}}\left[\text{IoU}*\widehat{P}_i^j\log\left(P_i^j\right)+\left(1-IoU*\widehat{P}_i^j\right)\log\left(1-P_i^j\right)\right]$$
$$-\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{I}_{ij}^{\text{Obj}}\left[\widehat{C}_i^j\ln\left(C_i^j\right)+\left(1-\widehat{C}_i^j\right)\ln\left(1-C_i^j\right)\right]-\lambda_{\text{noobj}}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{I}_{ij}^{\text{noobj}}\left[\widehat{C}_i^j\ln\left(C_i^j\right)+\left(1-\widehat{C}_i^j\right)\ln\left(1-C_i^j\right)\right]. \tag{11}$$

And,

$$-\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{I}_{ij}^{Obj}\left[\widehat{C}_i^j\ln\left(C_i^j\right)+\left(1-\widehat{C}_i^j\right)\ln\left(1-C_i^j\right)\right]-$$
$$\lambda_{\text{noobj}}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{I}_{ij}^{\text{noobj}}\left[\widehat{C}_i^j\ln\left(C_i^j\right)+\left(1-\widehat{C}_i^j\right)\ln\left(1-C_i^j\right)\right] \tag{12}$$

is the loss of confidence predictions for boxes. The maximum matching between the prediction bounding boxes and ground truth boxes can be obtained by the loss function associated class prediction probability and position accuracy because this loss function is directly representative of the core evaluation metric. And DCNets modules in detection network are also supervised by this loss function. Figure 5 shows the example results from COCO validation using YOLOv4 trained employing (left to right) original loss function and the proposed loss function (11).

4. Experimental Results

4.1. Ablation Study. We use PASCAL VOC 2007 [51] and COCO 2017 data set [52] and follow their original protocol. For PASCAL VOC 2007, training is performed on the union of VOC 2007 trainval and VOC 2012 trainval and evaluating on VOC 2007 test set. For COCO, our models are trained and evaluated on the 120k images of the COCO 2017 trainval and 20k images of the COCO 2017 test-dev set. For evaluation, we use AP^{bbox} ($AP50 + AP55 + \cdots + AP90 + AP95$)/10, where AP50 means mean average precision (mAP) with IoU threshold of 0.50 and AP95 means mAPs with IoU threshold of 0.95 as performance measurement. We do not deliberately distinguish between small objects, middle objects, and large objects because Faster-RCNN and YOLOv3/YOLOv4 are superior to small object detection. And the latest transformer based target detector [53] shows excellent small target detection performance. ImageNet [54] pretrained ResNet-50 [55] is utilized as the backbone. For comparison with the same standard, we report results of Faster-RCNN/YOLOv4 with the backbone ResNet-50 even though YOLOv4 recommends CSPDarknet53 backbone. In training and inference, parameter setting and training strategy mainly follow DCNv1 [6] and DCNv2 [7] except the image resolution, iterations, and learning rates. The images are resized to have a shorter side of 600 pixels. A total of 30k and 50k iterations are performed on PASCAL VOC and COCO, respectively. The learning rates are 10^{-3}. DCNv1 [6] and DCNv2 [7] show that the more regular 3×3 convolutions are replaced, the better the final result is. According to DCNv2, employing deformable layers in the conv3 ~ conv5 stages achieves the best tradeoff between accuracy and efficiency for object detection on COCO. To construct different deformable convolutional networks, we replace the layers of 3×3 convolution in the conv3 ~ conv5 stage in YOLOv4 and Faster R-CNN with deformable/modulated-deformable/curvature-driven-deformable conv layers. And aligned RoIpooling is replaced by deformable/modulated-deformable/curvature-driven-deformable RoIpooling. In our experiments, the models are trained with 2 Nvidia GTX 2080Ti GPUs.

The comparison results of DCNv1, DCNv2, and curvature-driven deformation modeling on PASCAL VOC data set are shown in Table 1, and the comparison results on COCO data set are shown in Table 2. On PASCAL VOC, the DCNv1 obtains an increase of 2.1% ~ 2.5% in AP^{bbox} scores compared to the baseline, and DCNv2 module obtains further gains about 1.9% ~ 1.9% on the basis of DCNv1. On COCO, the DCNv1 obtains an AP^{bbox} score of 40.1% for Faster R-CNN and 42.3% for YOLOv4 when the layers of 3×3 convolution in the conv3 ~ conv5 stage and the aligned RoIpooling layer are replaced by their deformable counterparts, which are higher than the baseline about 0.6% ~ 0.7%, respectively. DCNv2 obtains further gains about 1.1% ~ 1.3% in AP^{bbox} scores with a small increase in parameters addition and FLOPs. The accuracies of DCNv1 and DCNv2 are lower than that reported in [6, 7]; the main reason is that the model we trained is slightly worse. Compared with the significant improvements of PASCAL VOC by DCNv1 and DCNv2, the improvements on COCO are not very significant. The reason is that COCO is larger and more challenging, which makes it more difficult to learn the offsets and modulation scalar implicitly. As shown in the table, our C-DCNets get better results. On PASCAL VOC, our curvature-driven deformation model yields a 75.2% AP^{bbox} on Faster R-CNN and 76.5% AP^{bbox} for YOLOv4, which is 3.4%% and 3.5% higher than that of the DCNv1 for Faster R-CNN and YOLOv4, respectively. On COCO, our curvature-driven deformation model yields a 43.3% AP^{bbox} on Faster R-CNN and 45.2% AP^{bbox} for YOLOv4, which is 3.2% and 2.9% higher than that of the DCNv1 for Faster R-CNN and YOLOv4, respectively. Note that the parameter quantity of our C-DCNets is the same as that of DCNv1 model, FLOPs are slightly increased with DCNv1 model, and the performance is better than that of DCNv2. The improvement of performance is mainly due to the stronger deformation ability of our model.

Extensive ablation studies in object detection are performed to validate the efficacy and efficiency of the combination of the C-DCNets and the proposed loss function (11). We apply YOLOv4 model with C-dconv@c3 ~ c5+Cdpool and replace YOLOv4's original loss function with the proposed loss function (11). For DETR, we

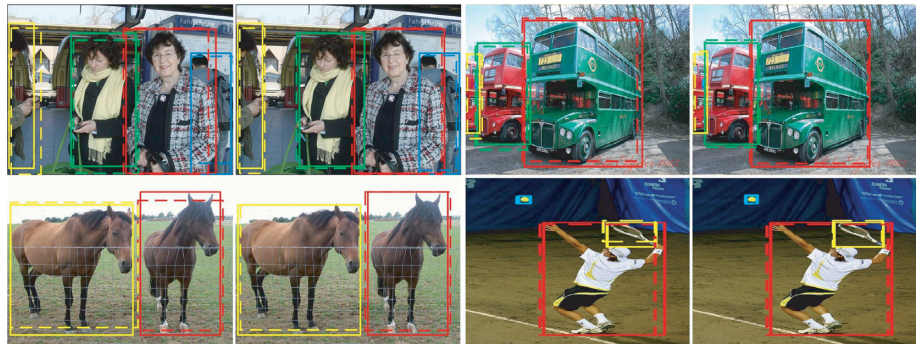

Figure 5: Example results from COCO validation using YOLOv4 trained employing (left to right) original loss function and the proposed loss function (11). Ground truth is shown by a solid line and predictions are represented with dashed lines.

Table 1: Detection results on PASCAL VOC 2007 test set. The detectors are Faster R-CNN and YOLOv4.

Method	Shorter side (600)	Faster R-CNN			YOLOv4		
		AP^{bbox}	AP_{50}	AP_{75}	AP^{bbox}	AP_{50}	AP_{75}
Baseline	Regular	69.7	78.6	62.5	70.5	79.8	63.4
Deformation	dconv@c3 ~ c5+ dpool(DCNv1) [6]	71.8	81.4	65.3	73.0	82.5	65.9
Modulated deformation	mdconv@c3 ~ c5+ Mdpool [7]	73.7	83.8	68.5	74.9	84.3	69.6
Curvature-driven Deformation	**C-dconv@c3 ~ c5+ Cdpool**	**75.2**	**85.4**	**71.6**	**76.5**	**86.2**	**73.5**

The input images are of shorts side 600 pixels. In the setting column, "(m)dconv" and "(m)dpool" stand for (modulated) deformable convolution and (modulated) deformable RoIpooling, respectively. "C-dconv" and "C-dpool" stand for curvature-driven deformable convolution and curvature-driven deformable RoIpooling, And dconv@c3 ~ c5 stands for applying deformable conv layers at stages conv3 ~ conv5, C-dconv@c3 ~ c5″ stands for applying curvature-driven deformable conv layers at stages conv3 ~ conv5. The bold value means the best value of each item.

Table 2: Ablation study on DCNv1, DCNv2, and our C-DCNets.

Method	Shorter side (600)	Faster R-CNN					YOLOv4				
		AP^{bbox}	AP_{50}	AP_{75}	# param	FLOP	AP^{bbox}	AP_{50}	AP_{75}	# param	FLOP
Baseline	Regular	39.4	60.8	42.4	51.30 M	100.1 G	41.7	63.5	44.7	26.8 M	146.4 G
Deformation	dconv@c3 ~ c5+ dpool(DCNv1) [6]	40.1	62.8	43.6	52.70 M	102.8 G	42.3	64.9	46.1	28.5 M	150.5 G
Modulated deformation	mdconv@c3 ~ c5+ Mdpool [7]	41.4	63.0	44.1	65.5 M	146.2 G	43.4	65.2	47.6	36.3 M	178.2 G
Curvature-driven Deformation	**C-dconv@c3 ~ c5+ Cdpool**	**43.3**	**65.2**	**47.4**	**52.7 M**	**109.1 G**	**45.2**	**67.0**	**49.8**	**28.5 M**	**165.4 G**

The input images are of shorts side 600 pixels. And all settings are consistent with Table 1. The bold value means the best value of each item.

choose ResNet-50-based DETR model with 3 encoder, 3 decoder layers, and width 256 because of the limitation of our GPU configuration. DETR model is trained for 100 epochs on 2 Nvidia GTX 2080Ti GPUs, and batch size is set as 4. Other parameters mainly follow DETR [22]. For deformable DETR, we just run one-stage mechanism with single-scale inputs, the backbone is ResNet-50, and the number of object queries is set as 100. Other hyperparameter setting and training strategy mainly follow deformable DETR [23]. Our DETR has a lower performance than published results because the baseline DETR has 6 encoder, 6 decoder layers, and width 256 with long training schedule. Deformable DETR achieves the best performance with 10 × less training epochs compared with DETR. Even through deformable DETR greatly reduces the amount of computation, its complexity is much great than YOLOv4, YOLOv4 with C-DCNets, and YOLOv4 with C-DCNets and the proposed loss function (11). Actually, deformable transformer will degenerate to deformable convolution when multiscale attention is not applied, and $K = 1$. The comparison results of different end-to-end detectors are shown in Table 3. It can be seen from the table that YOLOv4 with C-DCNets and the proposed loss function achieves better results with low complexity.

Table 3: Comparison results of different end-to-end detectors.

Model	GFLOPS/FPS	# Params	AP^{bbox}	AP_{50}	AP_{75}
YOLOv4	146.4/42	26.8	41.7	63.5	44.7
YOLOv4 with C-dconv@c3 ~ c5+Cdpool	165.4/40	28.5	45.2	67.0	49.8
YOLOv4 with C-dconv@c3 ~ c5+Cdpool and the proposed loss function	168.5/40	28.5	**46.0**	**69.2**	**51.3**
DETR	80/19	37.4	38.8%	59.9	41.4
Deformable DETR	173/16	40	43.8%	65.2	48.5

The first row shows results YOLOv4 baseline. The second row shows YOLOv4 models with C-dconv@c3 ~ c5+Cdpool. The third row shows results for YOLOv4 models with C-dconv@c3 ~ c5+Cdpool and the proposed loss function (11). The fourth row shows results for DETR model and the last row shows results for deformable DETR. Results are reported on the COCO 2017 validation set. The bold value means the best value of each item.

5. Conclusion

In this paper, curvature-driven deformable convolution networks (C-DCNets) are proposed. The deformation ability of convolution operation with irregular grid is further enhanced. The final offset fields are not only driven by the task goal, but also guided by the curvature fields of the preceding feature maps, which deform networks sampling and pooling patterns to fit the object's structure. To be consistent with the evaluation score of postprocessing of detection network, a new loss function associated with bounding box regression and classification is proposed, in which the class prediction probability in the binary cross entropy loss function integrates the similarity of the predicted boxes and targets. Experimental results on PASCAL VOC 2007 and COCO 2017 data sets show that C-DCNets based YOLOv4 with the proposed loss function outperforms state-of-the-art detectors without bells and whistles.

Data Availability

The experimental data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (No. 61 701 201).

References

[1] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proceedings of the Conference on Neural Information Processing Systems(NIPS)*, Montreal, Canada, December 2015.

[2] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 2011–2023, 2020.

[3] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: convolutional block attention module," Edited by V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., in *Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science, V. Ferrari, M. Hebert, C. Sminchisescu*, vol. 11211, Springer, Cham, Switzerland, 2018.

[4] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 510–519, Long Beach, CA, USA, 2019.

[5] X. Zhu, D. Cheng, Z. Zhang, S. Lin, and J. Dai, "An empirical study of spatial attention mechanisms in deep networks," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6687–6696, Long Beach, CA, USA, 2019.

[6] J. Dai, H. Qi, Y. Xiong et al., "Deformable convolutional networks," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 764–773, Venice, Italy, 2017.

[7] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable ConvNets V2: more deformable, better results," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9300–9308, Long Beach, CA, USA, 2019.

[8] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "RepPoints: point set representation for object detection," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9656–9665, Seoul, South Korea, 2019.

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the Computer Vision and Pattern Recognition, 2005*, vol. 1, pp. 886–893, San Diego, CA, USA, June 2005.

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[11] Y. Li, K. He, and J. Sun, "R-fcn: object detection via region based fully convolutional networks," *Advances in Neural Information Processing Systems*, vol. 29, pp. 379–387, 2016.

[12] S. Ren, K. He, R. Girshick, and S. Jian, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Proceedings of the Neural Information Processing Systems*, pp. 91–99, USA, December 2015.

[13] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *Proceedings of the European conference on computer vision*, pp. 21–37, Florence, Italy, October 2016.

[14] J. Redmon, S. K. Divvala, and R. Girshick, "You only look once: unified, real-time object detection," in *Proceedings of the Computer Vision and Pattern Recognition*, pp. 779–788, Bostony, USA, June 2016.

[15] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the Computer Vision and Pattern Recognition*, pp. 6517–6525, Honolulu, HI, USA, July 2017.

[16] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," in *Proceedings of the Computer Vision And Pattern Recognition*, Salt Lake City, UT, USA, June2018.

[17] A. Bochkovskiy, C. Y. Wang, and H. Liao, "YOLOv4: optimal speed and accuracy of object detection," 2020.

[18] T. Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proceedings of the Computer Vision and Pattern Recognition*, Honolulu, HI, USA, July 2017.

[19] X. Zhou, D. Wang, and P. Krahenbuhl, "Objects as points," 2019, https://arxiv.org/abs/1904.07850.

[20] T. Zhang, X. Zhang, C. Liu et al., "Balance learning for ship detection from synthetic aperture radar remote sensing imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 182, pp. 190–207, 2021.

[21] T. Zhang, X. Zhang, J. Shi et al., "Balance scene learning mechanism for offshore and inshore ship detection in SAR images," *IEEE Geoscience and Remote Sensing Letters*, vol. 2022, Article ID 3033988, 2022.

[22] N. Carion, F. Massa, S. Gabriel, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end Object Detection with Transformers," in *Proceedings of the ECCV*, 2020.

[23] X. Zhu, W. Su, and L. Lu, "Deformable DETR: deformable transformers for end-to-end object detection," 2020, https://arxiv.org/abs/2010.04159.

[24] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the ICLR*, San Diego, CA, USA, May 2015.

[25] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the EMNLP*, Lisbon, Portugal, September 2015.

[26] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the ICML*, Sydney, Australia, August 2017.

[27] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Proceedings of the NIPS*, Long Beach, CA, USA, December 2017.

[28] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 2, New Orleans, LO, USA, June 2018.

[29] J. Fu, J. Liu, H. Tian et al., "Dual attention network for scene segmentation," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3141–3149, Long Beach, CA, USA, June 2019.

[30] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June 2018.

[31] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June 2018.

[32] J. Gu, H. Hu, L. Wang, Y. Wei, and J. Dai, "Learning region features for object detection," in *Proceedings of the ECCV*, Munich, Germany, September 2018.

[33] T. Zhang and X. Zhang, "ShipDeNet-20: an only 20 convolution layers and," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 7, pp. 1234–1238, 2021.

[34] T. Zhang, X. Zhang, J. Shi, and S. Wei, "HyperLi-Net: a hyper-light deep learning network for high-accurate and high-speed ship detection from synthetic aperture radar imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 167, pp. 123–153, 2020.

[35] T. Zhang and X. Zhang, "Squeeze-and-Excitation laplacian pyramid network with dual-polarization feature fusion for ship classification in SAR images," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[36] T. Zhang, X. Zhang, X. Ke et al., "HOG-ShipCLSNet: a novel deep learning network with HOG feature fusion for SAR ship classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 2021, Article ID 3082759, 22 pages, 2021.

[37] T. Zhang and X. Zhang, "A polarization fusion network with geometric feature embedding for SAR ship classification," *Pattern Recognition*, vol. 123, Article ID 108365, 2021.

[38] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 580–587, Columbus, OH, USA, June 2014.

[39] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS improving object detection with one line of code," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 5562–5570, Seoul, Korea, October 2017.

[40] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU Loss: faster and better learning for bounding box regression," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, New York, USA, February 2020.

[41] J. H. Hosang, R. Benenson, and B. Schiele, "Learning non-maximum suppression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

[42] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Real-time instance segmentation," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9156–9165, Seoul, Korea, October 2019.

[43] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, June 2014.

[44] O. Vinyals, S. Bengio, and M. Kudlur, "Order matters: sequence to sequence for sets," in *Proceedings of the ICLR*, San Juan, Puerto Rico, May 2016.

[45] H. W. Kuhn, "The Hungarian Method for the Assignment Problem," *Journal of Physics Conference Series*, vol. 1377, no. 1, Article ID 012046, 1955.

[46] T. -Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944, Miami, FL, USA, 2017.

[47] J. Shen, S. H. Kang, and T. F. Chan, "Euler's elastica and curvature-based inpainting," *SIAM Journal on Applied Mathematics*, vol. 63, no. 2, pp. 564–592, 2003.

[48] F. Zana and J.-C. Klein, "Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation," *IEEE Transactions on Image Processing*, vol. 10, no. 7, pp. 1010–1019, 2001.

[49] F. Mokhtarian and A. K. Mackworth, "A theory of multiscale, curvature-based shape representation for planar curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 8, pp. 789–805, 1992.

[50] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: a metric and a loss for bounding box regression," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 658–666, Long Beach, CA, USA, June 2019.

[51] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[52] T.-Y. Lin, M. Maire, S. Belongie et al., "Microsoft coco: common objects in context," in *Proceedings of the European conference on computer vision*, pp. 740–755, Springer, Munich, Germany, September 2014.

[53] Z. Liu, Y. Lin, Y. Cao et al., "Hierarchical vision transformer using shifted windows," 2021, https://arxiv.org/abs/2103.14030.

[54] D. Jia, W. Dong, R. Socher, L. J. Li, K. Li, and L. F. Fei, "Imagenet: a large-scale hierarchical image database," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami, Florida, USA, June 2009.

[55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.