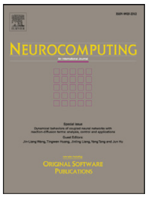
Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Single-shot bidirectional pyramid networks for high-quality object detection

Xiongwei Wu[a,*], Doyen Sahoo[c], Daoxin Zhang[a,b], Jianke Zhu[a], Steven C.H. Hoi[a,c]

[a] *School of Information Systems, Singapore Management University, Singapore*
[b] *College of Computer Science and Technology, Zhejiang University, China*
[c] *Salesforce Research Asia, Singapore*

ARTICLE INFO

Article history:
Received 17 July 2019
Revised 17 January 2020
Accepted 24 February 2020
Available online 12 March 2020

Communicated by zhi yong Liu

Keywords:
Object detection
Deep learning
Computer vision
Anchor refinement

ABSTRACT

Recent years have witnessed significant advances in deep learning based object detection. Despite being extensively explored, most existing detectors are designed to detect objects with relatively low-quality prediction of locations, i.e., they are often trained with the threshold of Intersection over Union (IoU) set as 0.5. This can yield low-quality or even noisy detections. Designing high quality object detectors which have a more precise localization (e.g. IoU > 0.5) remains an open challenge. In this paper, we propose a novel single-shot detection framework called Bidirectional Pyramid Networks (BPN) for high-quality object detection. It comprises two novel components: (i) Bidirectional Feature Pyramid structure and Anchor Refinement (AR). The bidirectional feature pyramid structure aims to use semantic-rich deep layer features to enhance the quality of the shallow layer features, and simultaneously use the spatially-rich shallow layer features to enhance the quality of deep layer features, leading to a stronger representation of both small and large objects for high quality detection. Our anchor refinement scheme gradually refines the quality of pre-designed anchors by learning multi-level regressors, giving more precise localization predictions. We performed extensive experiments on both PASCAL VOC and MSCOCO datasets, and achieved the best performance among all single-shot detectors. The performance was especially superior in the regime of high-quality detection.

1. Introduction

Object detection is a fundamental research problem in computer vision. Recent years have witnessed remarkable progress in object detection algorithms catalyzed by the success of powerful deep learning techniques [1–3]. Currently, the state-of-the-art deep learning based object detection frameworks can be generally categorized into two major groups: (i) two-stage detectors, such as the family of Region-based CNN (R-CNN) [2] and their variants [1,4] and (ii) one-stage detectors, such as SSD [5] and its variants [6,7]. Two-stage RCNN-based detectors first learn to generate a sparse set of proposals followed by training region classifiers, while one-stage SSD-like detectors directly make categorical prediction of objects based on the predefined anchors on the feature maps without a proposal generation step. Two-stage detectors usually achieve better detection performance and often report state-of-the-art results on benchmark data sets, while one-stage detectors are significantly more efficient and thus more suitable for many real-word practical/industrial applications where fast/real-time detection speed is of crucial importance.

Despite being studied extensively, most existing object detectors are designed for achieving localization with relatively low-quality precision (e.g. Intersection over Union (IoU) threshold of 0.5 is considered good enough). When the goal is to achieve higher quality localization precision (IoU > 0.5), the detection performance often drops significantly [8]. A naive solution to address this issue is to increase the IoU threshold when selecting positive samples (e.g., from 0.5 to 0.7) during training, such that the detector is trained on only high quality examples. Unfortunately, such a strategy will lead to very few (positive) training samples, and will consequently lead to overfitting, especially for single-shot SSD-like detectors. In addition, most object detectors aim to use the strength of deep features for object localization. This can have adverse effects as deep features (while being semantically rich) lack detailed information about the spatial location of the objects.

In this paper, we aim to develop a novel high-quality single-shot detector. We follow the family of single-stage SSD-like detec-

[*] Corresponding author.
E-mail addresses: xwwu.2015@phdis.smu.edu.sg (X. Wu), dsahoo@salesforce.com (D. Sahoo), dxzhang@zju.edu.cn (D. Zhang), jkzhu@zju.edu.cn (J. Zhu), chhoi@smu.edu.sg, shoi@salesforce.com (S.C.H. Hoi).

https://doi.org/10.1016/j.neucom.2020.02.116
0925-2312/© 2020 Elsevier B.V. All rights reserved.

tors, and design an approach that makes it amenable for high quality detection. We identify two critical drawbacks of SSD-like detectors for learning high quality detectors: first, the single-shot feature representations may not be discriminative and robust enough for precise localization; and second, the singe-stage detection scheme relies on the predefined anchors which are very rigid and often inaccurate. To overcome these drawbacks for high-quality object detection tasks, in this paper, we propose a novel single-shot detection framework named "Bidirectional Pyramid Networks" (BPN). Specifically, BPN uses a novel Bidirectional Pyramid Structure, that boosts the vanilla feature pyramid [3] by reinforcing it with a Reverse Feature Pyramid to fuse both deep and shallow features to learn more effective and robust representations. Unlike Feature Pyramid Network (FPN) which aims to enhance the shallow features with semantically rich deep features, the Reverse FPN aims to enhance the deep features with spatially rich shallow features, thereby improving the representation for better localization. BPN is also augmented with a novel Anchor Refinement scheme that learns to gradually improve the quality of predefined anchors which are often inaccurate at the beginning. Specifically, we train the bounding box regressors at different levels of qualtiy (IoU thresholds), and in an incremental manner, feed the bounding box predictions of a specific quality into the predictions of the next higher quality. We conducted extensive experiments on PASCAL VOC and MSCOCO showed that the proposed method achieved the state-of-the-art results for high-quality object detection while still maintaining the advantage of computational efficiency of single shot detectors.

2. Related work

Object detection has been extensively studied for decades [2,9]. In early stages of research, object detection was based on sliding windows, and dense image grids were encoded by hand-crafted features, which were followed by training classifiers to find and locate objects. Viola and Jones [9] proposed cascaded classifiers by AdaBoost with Haar features for face detection and obtained excellent performance with high efficiency. After the remarkable success of applying Deep Convolutional Neural Networks on image classification tasks [10–12], deep learning based approaches have been actively explored for object detection, in particular, the region-based convolutional neural networks (R-CNN) [2] and its variants [1,3,4]. Currently deep learning based detectors can be generally categorized into two groups: (i) two-stage RCNN-based methods and (ii) one-stage SSD-based methods. RCNN-based methods, such as RCNN [2], Fast RCNN [4], Faster RCNN [1], and R-FCN [13], first generate a sparse set of proposals followed by region classifiers and location regressors. Two-stage detectors usually achieve better detection performance (than one-stage detectors) and report state-of-the-art results on many common benchmarks. This is largely because the proposals are often carefully generated (e.g., by selective search [14] or RPN [1]) and the proposed regions tightly bound the objects in the image. However, they often suffer from very slow inference speed due to having two-stages to perform detection. Unlike the two-stage RCNN-based methods, SSD-style methods (one-stage detectors), such as SSD [5], YOLO [15], YOLOv2 [6]), ignore the proposal generation step by directly making predictions with manually designed pre-defined anchors and thus reduce the inference time significantly, enabling real-time detection. However, these anchors are often sub-optimal and sometimes ill-designed, and are unable to preciely match with the location of the objects in the image. Thus, SSD-style detectors [5] often struggle in the regime of high quality detection.

In literature, most object detection studies have focused on detection with relatively low localization quality, with a default IoU threshold of 0.5. There are have been limited efforts for high-quality detection. LocNet [16] learns a single postprocessing network for location refinement without changing the distribution of hypotheses in different quality stages. Their method is only optimal for the initial anchor distribution, while our method learns multi-level anchor refinements for different quality stages. MultiPath Network [17] proposed to learn multiple detection branches for different quality thresholds. However, this model suffered from not having sufficient training samples. Moreover, it was computationally slow by virtue of being a two-stage detectors. Cascaded RCNN [8] learned regressors in a cascaded way, which refined the proposal predictions sequentially. However, this was also based on two-stage RCNN which prevented its use in real time object detection. Moreover, they consider only refining the anchor quality, and ignore the quality of feature representation for high quality detection.

Our work is also related to studies for multi-scale feature fusion, which has proved to be an effective structure for object detection with different scales. ION [18] extracted region features from different layers by ROI Pooling; HyperNet [19] directly concatenated features at different layers using deconvolution layers. FPN [3] and DSSD [20] fused features of different scales with lateral connection in a bottom-up manner, which effectively improved the detection of small objects. However, the vanilla feature pyramid [3] only considers boosting shallow layer features with deep layer features, but does not consider that shallow layer features could be helpful to deep semantic layer features by enriching them with crucial spatial information. We overcome this limitation by the proposed Bidirectional Feature Pyramid structure, where a reverse Feature Pyramid fuses the spatial information from shallow features with the deep leayer features. Moreover, none of these methods aim to refine the bounding box predictions, and are often susceptible to obtaining low quality predictions. In contrast, our anchor refinement strategy improves the model's ability to make high quality predictions.

3. Single-shot high-quality object detection

To train a detector, predefined anchors are often used. These anchors are generated densely or sparsely across the image, and the goal is to predict the class of object and the appropriate corrections to the original anchor localization. Each anchor is assigned to some object class label (including background) according to the anchor's Jaccard overlap score with ground-truth objects, a.k.a. "Intersection over Union" (IoU). When an anchor matches with the object for a given threshold, it is termed as a positive anchor. These positive anchors serve as ground truth for training. For objects that do not meet this threshold with any anchor, the best anchor is assigned as a positive anchor during the training stage. Our aim is to devise a new single-shot detector for high-quality object detection tasks by overcoming the drawbacks of state-of-the-art detectors. We tackle this challenge from both *feature representation* and *anchor-refining* perspectives. Existing single-shot object detectors, feature representations may not be discriminate and robust enough for precise localization, as they rely primarily on the deep layer features which while being semantically-rich, lack spatial information. We propose to strengthen deep layer features with spatially rich shallow feature to improve the localization performance. Second, for many state-of-the-art detectors, a group of anchors are often generated/pre-defined on the feature maps densely or sparsely, followed by location regression and object classification prediction. Due to the scale variance of the objects, and several downsampling steps from the original image, the manually designed anchors will often not be able to find a good match with the ground truth object locations. This issue becomes more prominent when we aim to train high-quality detectors with a high IoU threshold (e.g., 0.7) since the number of positive anchors would decrease significantly

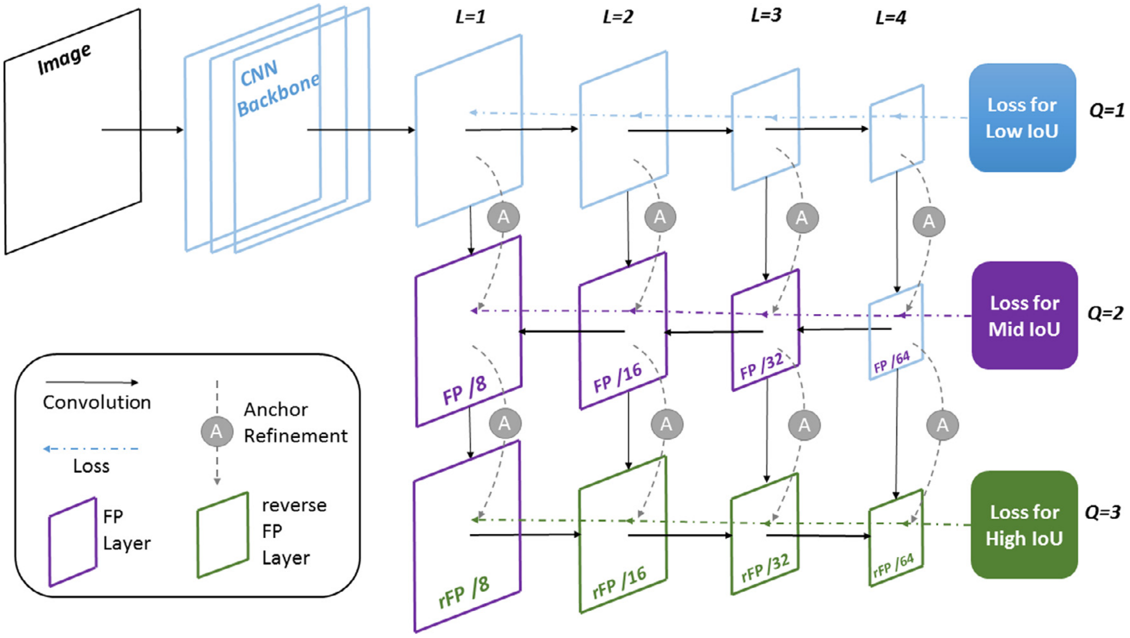


Fig. 1. The proposed framework of Bidirectional Pyramid Networks (BPN) for single-shot high-quality detection. *FP* denotes Feature Pyramid building block, and *rFP* denotes the Reverse Feature Pyramid building block. Bidirectional Feature Pyramid block generates more robust and discriminative feature map and the Anchor Refinement (*AR*) is utilized for relocating anchors, each level of which is responsible for a certain quality of detection. Training sample quality improves as the Anchor Refinement progresses (with higher IoU).

as IoU increases. This would consequently result in poor detection performance due to overfitting. Thus, we propose a novel anchor refinement procedure to improve the localization prediction.

3.1. Framework of bidirectional pyramid networks

We propose a novel framework called Bidirectional Pyramid Networks (BPN) to overcome the above drawbacks of SSD-style detectors, with the aim of developing a high-quality object detector. To address the weak feature representation issue of SSD-style detectors, we adapt the structure Feature Pyramid Networks (FPN) [3] and develop a novel Bidirectional Feature Pyramid structure that significantly boosts the effectiveness of Feature Pyramid(FP) structure. To address the issue of anchor quality, the key idea is to devise an effective yet efficient multi-level learning scheme to refine the quality of the anchors. We have classifiers and regressors at multiple levels, and for each level we train the classifier and regressor to refine anchors, before training the classifiers and regressors in the next level. Fig. 1 gives an overview of the proposed single-shot Bidirectional Pyramid Networks (BPN) for high-quality object detection, where the backbone network (as shown in the blue branch of Fig. 1) can be any CNN network, such as Alexnet [12], GoogleNet [21], VGG [11], ResNet [10], etc. For simplicity, we choose VGG-16 and ResNet-101 as backbone networks.

Similar to typical single-shot detectors, at the lowest quality level with the default IoU=0.5, the proposed BPN detector makes the prediction based on the predefined anchors. Then, the features are further enhanced by the Bidirectional Feature Pyramid which aggregates features from different depths. It consists of standard feature pyramids in a bottom-up fashion (the purple branch of Fig. 1) and reverse feature pyramid in a top-down fashion (the green branch of Fig. 1). These three-level branches not only aggregate multi-level features to provide robust feature representations, but also enable multi-quality training. For the joint training with multiple quality levels, the Anchor Refinement scheme with multi-level learning optimizes anchors from the previous level/branch and sends them to the next level/branch.

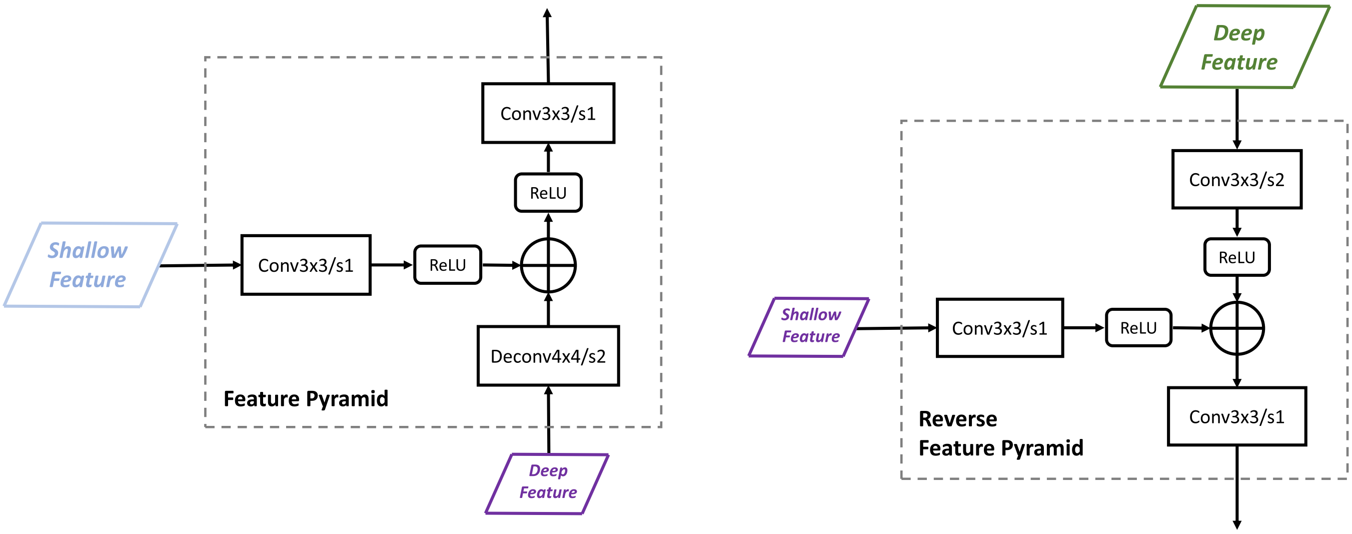
The above two key components, Bidirectional Feature Pyramid and Anchor Refinement, are seamlessly integrated in the proposed framework and can be trained end-to-end to achieve high-quality detection in a synergic manner. In the following, we present the detailed functioning of these components.

3.2. Bidirectional feature pyramid structure

We denote the index of feature maps for prediction as L, where $L \in \{1, 2, 3, 4\}$ in our setting, and the levels of quality $Q \in \{1, 2, 3, \ldots\}$ with the corresponding IoU thresholds as $\text{IoU}(Q) \in \{0.5, 0.6, 0.7, \ldots\}$. The feature map in depth L for quality Q prediction is denoted as F_L^Q, and anchors for training quality Q detector in depth L are denoted as A_L^Q. Specifically for this work, we choose three types of detectors with different quality levels: *Low*, *Mid* and *High* with the corresponding IoU threshold as 0.5, 0.6 and 0.7, respectively (See Fig. 1 for details).

In order to improve the power of feature representation of SSD-style detectors, we apply Feature Pyramids (FP) [3], which exploits the inherent multi-scale and pyramidal hierarchy of deep convolutional networks to construct the representation of feature pyramids. Specifically, FPN fuses semantically-strong deep layer features with shallow features which are semantically-weak but spatially-strong. The idea is to strengthen the features by helping them with stronger semantic information. We propose to augment this structure via a reverse Feature Pyramid (rFP), where the deep features are strengthened by the spatially strong shallow features.

Reverse Feature Pyramid has several strengths. First, the deep feature representations are enhanced to for better localization of large objects in the high quality scenario; second, compared to stacked CNN for image classification, rFP reduces the *distance* from shallow features to deep features by using much fewer convolution filters and thus more effectively preserves spatial information. Finally, the lateral connections *reuse* different shallow layer features to reduce information attenuation from shallow features to deep features. We demonstrate this concept in Fig. 2. Specifically, Fig. 2(a) is the vanilla Feature Pyramid building block that fuses features in a bottom-up manner with lateral connections. It is worth noting that there is no strengthening of the



(a) Feature Pyramid(*FP*) building block (b) Reverse Feature Pyramid(*rFP*) building block

Fig. 2. Proposed bidirectional feature pyramid structure.

deepest feature layer from the Feature Pyramid (the right diagram of Fig. 1). Thus, we further build the Reverse Feature Pyramid by top-down aggregation (as shown in Fig. 2 (b)) with lateral connections to enhance deep layer features with rich spatial information.

The formulations of Feature Pyramid (FP) and reverse Feature Pyramid (rFP) can be represented as:

$$\mathrm{FP}: \quad F^Q_L = \mathrm{Deconv}_{s2}(F^Q_{L+1}) \oplus \mathrm{Conv}(F^{Q-1}_L) \tag{1}$$

$$\mathrm{rFP}: \quad F^Q_L = \mathrm{Conv}_{s2}(F^Q_{L-1}) \oplus \mathrm{Conv}(F^{Q-1}_L) \tag{2}$$

where Deconv_{s2} denotes the deconvolution operation for feature map up-sampling with stride 2 and Conv denotes convolution operation. $\oplus$ denotes element-wise summation. In this paper, we use 3 × 3 convolution kernels with 256 channels to build the Feature Pyramid and Reverse Feature Pyramid in our BPN detector.

3.3. Anchor refinement

In order to both increase the number of positive anchors during training and improve their quality, we propose the Anchor Refinement ("AR"). We denote the anchors used at quality Q, depth L as AR^Q_L. In particular, AR has two parts: location regressor Reg^Q_L and a categorical classifier Cls^Q_L. At each level of quality, regressors receive the processed anchors from the previous level of quality for further optimization (A^1_L is the set of manually defined anchor):

$$A^Q_L = Reg^Q(A^{Q-1}_L; F^Q_L), \quad Q = 2, 3, \ldots, L = 1, 2, \ldots \tag{3}$$

A set of offsets is learned from the regressors to adjust the location of the predicted bounding boxes. Different from vanilla SSD, these bounding boxes are conditioned on the refined anchors and are be used as new anchors in next stage.

Categorical classifiers learn to predict categorical confidence scores and assign them to these anchors:

$$C^Q_L = Cls^Q(F^Q_L), \quad Q = 1, 2, 3 \ldots, L = 1, 2, \ldots \tag{4}$$

Thus, the training loss at quality level Q can be written as:

$$\ell^Q = \frac{1}{N_Q} * \sum_L \sum_i \Big(\ell^Q_{\mathrm{Cls}}(\{C^Q_{L_i}\}, \{t_{L_i}\}) + \lambda * \ell^Q_{\mathrm{Reg}}(\{A^Q_{L_i}\}, \{g_{L_i}\}) \Big) \tag{5}$$

where N_Q is the positive sample number at quality level Q, L_i is the index of anchor in depth L feature map within a mini-batch, t_{L_i} is the ground truth class label of anchor L_i, g_{L_i} is the ground truth location and size of anchor L_i, λ is the balance weighting parameter which is simply set to 1 in our settings. $L^Q_{\mathrm{Cls}}(.)$ is softmax loss function over multiple classes confidences and $L^Q_{\mathrm{Reg}}(.)$ is the Smooth L1-loss which is also used in [5]. The total training loss is the summation of losses at all the quality levels:

$$\ell_{\mathrm{BPN}} = \sum_Q \ell^Q \tag{6}$$

3.4. Implementation details

CNN backbone architecture: We choose VGG16 [11] and ResNet-101 [10] pre-trained on ImageNet as the backbone networks in our experiments. For VGG16, we follow [5] to transform the last two fully-connected layers "fc6" and "fc7" to convolutional layers "conv_fc6" and "conv_fc7" via reducing parameters. To increase receptive fields and capture large objects, we attached two additional convolution layers after the VGG16 (denoted as conv6_1 and conv6_2). Due to different scale norm in different feature maps, we re-scale the norms of the first two feature blocks to 10 and 8 respectively. For ResNet-101, we added one extra residual block "res6" at the end of the network.

Data augmentation: We adopt the augmentation strategies in [5] to make the detectors robust to objects with the changes in scale and color. Specifically, images are randomly expanded or cropped with additional photometric distortion to generate additional training samples.

Feature blocks for prediction: In order to detect objects at different scales, we use multiple feature maps for prediction. The vanilla convolution feature blocks in backbone are used for low-quality detection, feature pyramid blocks are used for mid-quality detection, and the reverse feature pyramid blocks are used for high-quality detection. We use four feature blocks with stride 8, 16, 32 and 64 pixels in training each quality detector. In VGG16, conv4_3, conv5_3, conv_fc7, conv6_2 and their corresponding feature pyramid blocks FP3, FP4, FP5 and FP6, and reverse feature pyramid blocks rFP3, rFP4, rFP5 and rFP6 are used, while in ResNet-101, res3b3, res4b22, res5c, res6 and their corresponding feature pyramid blocks and reverse feature pyramid blocks are used.

Table 1
Detection results on PASCAL VOC dataset. All the methods were trained on VOC2007 and VOC2012 `trainval` sets and tested on VOC2007 `test` set.

Method	Backbone	Input size	FPS	mAP (%)		
				IoU@0.5	IoU@0.6	IoU@0.7
Two-stage Detectors:						
Fast R-CNN [4]	VGG-16	~ 1000 × 600	0.5	70.0	62.4	49.4
Faster R-CNN [1]	VGG-16	~ 1000 × 600	7	73.2	67.7	54.4
OHEM [23]	VGG-16	~ 1000 × 600	7	74.6	68.9	55.9
HyperNet [19]	VGG-16	~ 1000 × 600	0.88	76.3	-	-
Faster R-CNN [10]	ResNet-101	~ 1000 × 600	2.4	76.4	69.5	57.3
ION [18]	VGG-16	~ 1000 × 600	1.25	76.5	-	-
LocNet [16]	VGG-16	~ 1000 × 600	-	77.5	-	64.5
R-FCN [13]	ResNet-101	~ 1000 × 600	9	80.5	73.2	61.8
R-FCN Cascade [8]	ResNet-101	~ 1000 × 600	7	81.0	75.8	66.7
CoupleNet [24]	ResNet-101	~ 1000 × 600	8.2	81.7	76.6	66.8
One-stage Detectors:						
RON384 [25]	VGG-16	384 × 384	15	75.4	66.8	54.2
SSD300 [5]	VGG-16	300 × 300	46	77.3	72.3	61.3
DSOD300 [26]	DS/64-192-48-1	300 × 300	17.4	77.7	73.4	63.6
YOLOv2 [6]	Darknet-19	544 × 544	40	78.6	69.1	56.5
SSD512 [5]	VGG-16	512 × 512	19	79.8	74.7	64.0
RefineDet320 [7]	VGG-16	320 × 320	40.3	80.0	74.2	63.6
RefineDet512 [7]	VGG-16	512 × 512	24.1	81.8	76.9	66.0
RFBNet300 [27]	VGG-16	300 × 300	83.0	80.7	75.5	65.5
RFBNet512 [27]	VGG-16	512 × 512	38.0	82.2	-	-
BPN320(ours)	VGG-16	320 × 320	32.4	80.3	75.5	66.1
BPN512(ours)	VGG-16	512 × 512	18.9	**82.2**	**77.6**	**68.3**

Anchor design: Originally a group of anchors are pre-designed manually. For each prediction feature block, one scale-specific set of anchors with three aspect ratios isssociated. In our approach, we set the scale of anchors as 4 times that of the feature map stride and set the aspect ratios as 0.5, 1.0 and 2.0 to cover different scales of objects. We first match each object to the anchor box with the best overlap score, and then match the anchor boxes to any ground truth with overlap higher than the quality thresholds.

Optimization: We use “Xavier” method in [22] to randomly initialize the parameters in extra added layers in VGG16 and ResNet-101. We set the mini-batch size as 32 in training and the whole network is optimized via the SGD optimizer (momentum=0.9, weight decay=0.005, and initial learning rate=0.001). The training strategy varies a bit for different datasets. For PASCAL VOC dataset, the models are completely finetuned for 120k iterations and we decrease the learning rate to 10^{-4} and 10^{-5} after 80k and 100k iterations, respectively. For MSCOCO, the models are finetuned for 400k iterations and we decrease the learning rate to 10^{-4} and 10^{-5} after 280k and 360k iterations, respectively. All the detectors were trained and optimized end-to-end.

Sampling strategy: The ratio of positive and negative anchors are imbalanced after the anchor matching step, so proper sampling strategy is necessary to address this imbalance. We sample a subset of negative anchors to keep the ratio of positive and negative anchors as 1:3 in training process. To achieve faster convergence, instead of randomly sampling negative anchors, we sort the negative anchors according to the loss sufferred by them and select the hardest ones for training. Different IoU thresholds are used for different quality levels. We use three quality levels (low, mid and high) for IoU as 0.5, 0.6 and 0.7, respectively.

Inference: During the inference phase, the anchor refinement different quality stage makes prediction and send the refined anchors to the next quality stage. We take the predictions from AR in all quality stages to ensure they are suitable for all the low-, mid- and high-quality detection.

4. Experiments

We conduct extensive experiments on two publicly available benchmark datasets: Pascal VOC and MSCOCO. The evaluation metric for the detector performance is mean average precision which is widely used in evaluating object detection.

4.1. Pascal VOC experiment

We use Pascal VOC2007 trainval set and Pascal VOC2012 trainval set as our training set, and VOC2007 test set as testing set. There are 16k images for training and 5k images for testing. All models are based on VGG16 architecture as ResNet-101 has limited benefits for this dataset [20]. We train BPN with two resolutions of the input (320 × 320 and 512 × 512) and compare them with the state-of-the-art methods on low, mid and high quality detection scenarios (IoU thresholds as 0.5, 0.6 and 0.7, respectively).

We show the comparison of performance of our proposed method BPN320 and BPN512 against several state of the art two-stage and one-stage baseline detectors in Table 1. BPN320 obtains an accuracy of 80.3%, 75.5% and 66.1% in low, mid and high quality detection scenario respectively, which outperforms many detectors (e.g., SSD320, Faster RCNN, etc.). BPN512 achieves the state-of-the-art results of 82.2%, 77.6% and 68.3% for three scenarios respectively. Notably, BPN has clear advantage in high quality detection scenario(IoU=0.7). BPN is one-stage detector, and can thus be used for real-time inference. BPN320 can perform inference at 32.4fps while BPN512 at 18.9fps on a Titan XP GPU.

4.2. Ablation studies

In this section, we conduct a series of ablation studies to analyze the impact of different components of BPN. We use VOC2007 and VOC2012 `trainval` set as our training set and test on VOC2007 `test` set. We use mean average precision on three different IoU thresholds (0.5, 0.6 and 0.7) as our evaluation metric. The results are shown in Table 2.

Bidirectional feature pyramid: To validate the effectiveness of the Bidirectional Feature Pyramid, we remove all Anchor Refinement components from BPN leaving only one classifier, and compare this model (called as BPN w / o AR) with vanilla SSD and SSD+FP. Bidirectional Feature Pyramid is built based on vanilla SSD and all three models are fine-tuned with IoU threshold as 0.5. In Table 2,

Table 2
Detection results on PASCAL VOC dataset. For VOC 2007, all methods are trained on VOC 2007 and VOC 2012 `trainval` sets and tested on VOC 2007 `test` set. Original SSD uses six feature maps for prediction, while we use four feature maps to be consistent with BPN, so the detection result of SSD here is a bit lower. "Training IoU" denotes IoU thresholds trained for different stages ("-" means no classifier in this stage). Bold fonts indicate the best mAP.

	Training IoU	mAP@IoU=0.5	mAP@IoU=0.6	mAP@IoU=0.7
SSD	(0.5, -, -)	76.3	71.0	60.4
SSD	(0.7, -, -)	68.4	61.9	50.8
SSD+FP	(-,0.5, -)	77.4	72.1	61.6
BPN w / o AR	(-, -,0.5)	78.1	72.7	63.4
SSD+FP+AR	(0.5, 0.5, -)	80.0	74.2	63.6
SSD+FP+AR	(0.5, 0.7, -)	78.1	73.7	63.1
BPN	(0.5, 0.5, 0.7)	80.0	75.1	65.4
BPN	(0.5, 0.6, 0.7)	**80.3**	**75.5**	**66.1**

we can see that SSD+FP outperforms vanilla SSD because deep semantic features boost feature representations. Further, BPN w / o AR outperforms SSD+FP in all quality scenarios, demonstrating its effectiveness.

Levels of AR: We aim to validate if the level of AR is important for training high-quality detectors. We show the results in Table 2. Firstly, a vanilla SSD was trained with 0.7 IoU threshold. This model (row 2) performs much worse than the baseline (row 1) trained with 0.5 IoU threshold in all three quality levels, which validates that insufficient positive training samples causes overfitting. Second, we keep a single level of AR block on SSD+FP (called "SSD+FP+AR"), and train this model with 0.5 IoU threshold. We can see that the detection results improve significantly compared with "BPN w/o AR" in low and mid quality scenarios, and is similar in the high-quality scenario (63.6% vs 63.4%). We further train "SSD+FP+AR" with 0.7 IoU threshold and this model (row 6) also suffers from overfitting issues but it is less severe compared to vanilla SSD. This shows that Anchor Refinement can boost detection performance by refining anchor quality. However, a single level of AR was not enough to boost the performance of the model. Finally, to the above model, we add one more level AR blocks and jointly optimize AR with different quality settings (0.5,0.5,0.7) and (0.5,0.6,0.7), which utilize high quality anchors for training. These two models (row 7 and row 8) further improve the performance significantly especially for high quality scenario (IoU=0.6 and IoU=0.7, etc.). In summmary, single level of AR is effective in addressing overfitting issues with SSD, and multi-level of AR are critical for enhancing the detection performance in high-quality scenarios.

Proposal quality improved by anchor refinement: In this section, we validate the effectiveness of the Anchor Refinement blocks to improve the anchor quality. In Fig. 3, we count the number of positive anchors per image for training under different IoU thresholds for SSD, SSD+FP+AR and BPN. For SSD, anchors are generated manually and only a few anchors matched objects under high IoU threshold metric, which makes it hard to train effective detectors. For SSD+FP+AR, anchors have been refined by AR once, and the number of positive anchors increases significantly under all IoU thresholds. Further in BPN where anchors are refined by AR twice, more high quality anchors are generated on more robust feature maps. Notably, after being refined by AR we have sufficient positive training samples even under high IoU metrics, so that we could conduct gradually increasing training positive IoU thresholds (0.5, 0.6 and 0.7). These results show that our AR blocks can gradually improve anchor qualities and generate more positive anchors for training.

Time analysis: As shown in Table 1, BPN shows significant speed advances compared with two-stage detectors and thus in this part we analyze the time complexity. For two-stage object detectors, the inference time consists of three parts: backbone convolution computation (T_{conv}), proposal generation (T_{proposal}), and region-wise operation (T_{region}, including region classification and region regression). Assume we have R regions to predict, the time complexity of two-stage detector is:

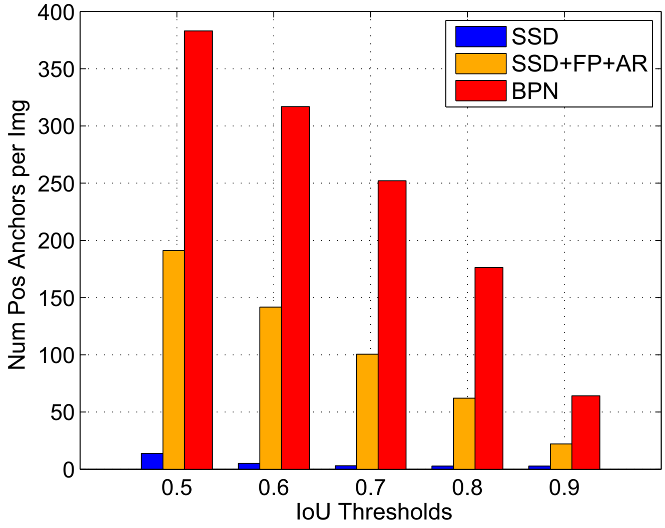


Fig. 3. Average *positive* anchor number per image by different approaches under different "IoU Threshold" metric.

$$\mathrm{T}_{\text{two-stage}} = T_{\text{conv}} + T_{\text{proposal}} + T_{\text{region}} \times R \tag{7}$$

Notably, region operation is operated across *all* R regions ($R = 300$ by default), which makes two-stage detectors slow. BPN is the one-stage detector and avoids the unshared region operation. BPN has additional two blocks: rFP and anchor refinement. For rFP, it only requires additional 4 convolution layers computation and for anchor refinement, only simple coordinate transformation is involved. Compared with the unshared region operation, the additional computation cost of BPN can be negligible:

$$\mathrm{T}_{\text{BPN}} = T_{\text{conv}} + T_{\text{proposal}} + T_{\text{rFP}} + T_{\text{AR}} \tag{8}$$

$$T_{\text{rFP}} + T_{\text{AR}} \ll T_{\text{proposal}} \times R \tag{9}$$

Thus our BPN is much faster than two-stage methods.

4.3. MSCOCO experiment

We also evaluate the performance of BPN on the MSCOCO data set [42], which has objects from 80 classes and about 120k images in `trainval` set. We use `trainval35k` set for training and test on `test-dev` set. Table 3 shows the results on MS COCO test-dev set. BPN320 with VGG-16 achieves 29.6% AP and when using larger

Table 3
Detection results on MS COCO `test-dev` set.

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage Detectors:							
Fast R-CNN [4]	VGG-16	19.7	35.9	-	-	-	-
Faster R-CNN [1]	VGG-16	21.9	42.7	-	-	-	-
OHEM [23]	VGG-16	22.6	42.5	22.2	5.0	23.7	37.9
ION [18]	VGG-16	23.6	43.2	23.6	6.4	24.1	38.3
OHEM++ [23]	VGG-16	25.5	45.9	26.1	7.4	27.7	40.3
R-FCN [13]	ResNet-101	29.9	51.9	-	10.8	32.8	45.0
CoupleNet [24]	ResNet-101	34.4	54.8	37.2	13.4	38.1	50.8
Faster R-CNN by G-RMI [28]	Inception-ResNet-v2	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN+++ [10]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [3]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Cascade RCNN w R-FCN [8]	ResNet-101	33.3	52.6	35.2	12.1	36.2	49.3
DeNet-101(wide) [29]	ResNet-101	33.8	53.4	36.1	12.3	36.1	50.8
DeNet [29]	ResNet-101	33.8	53.4	36.1	12.3	36.1	50.8
D-FCN [30]	Aligned-Inception-ResNet	37.5	58.0	-	19.4	40.1	52.5
Regionlets [31]	ResNet-101	39.3	59.8	-	21.7	43.7	50.9
Mask-RCNN [32]	ResNeXt-101	39.8	62.3	43.4	22.1	43.2	51.2
Soft-NMS [33]	Aligned-Inception-ResNet	40.9	62.8	-	23.3	43.6	53.3
Fitness NMS [34]	ResNet-101	41.8	60.9	44.9	21.5	45.0	57.5
Cascade RCNN w FPN [8]	ResNet-101	42.8	62.1	46.3	23.7	45.5	55.2
One-stage Detectors:							
YOLOv2 [6]	DarkNet-19	21.6	44.0	19.2	5.0	22.4	35.5
SSD300 [5]	VGG-16	25.1	43.1	25.8	6.6	25.9	41.4
RON384++ [25]	VGG-16	27.4	49.5	27.1	-	-	-
SSD321 [20]	ResNet-101	28.0	45.4	29.3	6.2	28.3	49.3
DSSD321 [20]	ResNet-101	28.0	46.1	29.2	7.4	28.1	47.6
SSD512 [5]	VGG-16	28.8	48.5	30.3	10.9	31.8	43.5
SSD513 [20]	ResNet-101	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [20]	ResNet-101	33.2	53.3	35.2	13.0	35.4	51.1
FPN-Reconfig [35]	ResNet-101	34.6	54.3	37.3	-	-	-
RetinaNet500 [36]	ResNet-101	34.4	53.1	36.8	14.7	38.5	49.1
RetinaNet800 [36]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RefineDet320 [7]	VGG-16	29.4	49.2	31.3	10.0	32.0	44.4
RefineDet512 [7]	VGG-16	33.0	54.5	35.5	16.3	36.3	44.3
RefineDet320 [7]	ResNet-101	32.0	51.4	34.2	10.5	34.7	50.4
RefineDet512 [7]	ResNet-101	36.4	57.5	39.5	16.6	39.9	51.4
ExtremeNet [37]	Hourglass-104	40.2	55.5	43.2	20.4	43.2	53.1
FCOS [38]	ResNeXt-101	42.1	62.1	45.2	25.6	44.9	52.0
FoveaBox [39]	ResNeXt-101	42.1	61.9	45.2	24.9	46.8	55.6
CenterNet-HG [40]	Hourglass-104	42.1	61.1	45.9	24.1	45.5	52.8
CornerNet511 [41]	Hourglass-104	40.5	56.5	43.1	19.4	42.7	53.9
CornerNet511++ [41]	Hourglass-104	42.1	57.8	45.3	20.8	44.8	56.7
BPN320	VGG-16	29.6	48.4	32.3	9.6	32.5	44.3
BPN512	VGG-16	33.1	53.1	36.3	15.7	37.0	44.2
BPN320++	VGG-16	35.4	55.3	38.5	19.0	37.9	47.0
BPN512++	VGG-16	37.9	58.0	41.5	21.9	41.1	48.1
BPN512	ResNet-101	37.6	59.1	40.5	18.7	42.2	50.8
BPN512++	ResNet-101	42.3	62.8	46.3	25.7	46.1	53.2

input image size 512, the detection accuracy of BPN512 reaches 33.1%, which is better than all other VGG16-based methods. Notably, we notice in high quality detection metric AP_{75}, BPN is clearly better than other detectors. As the objects in COCO dataset are of various scales, we also applied multi-scale testing based on BPN320 and BPN512 to reduce the impact of input size. The improved version BPN320++ and BPN512++ achieve 35.4% and 37.9% AP, which is the state-of-the-art performance among one-stage detectors. Different from Pascal VOC, using a deeper backbone such as ResNet could further improve detection accuracy compared to VGG16. Thus we report BPN512 with ResNet-101. Single BPN512 achieves 37.6% AP and when using multi-scale and flip horizontal inference, it improves to 42.3% AP, which is the state-of-the-art performance among one-stage detectors. Notably, BPN512++ achieves 46.3% on AP_{75}, which outperforms all other one-stage detectors significantly under high-quality metric.

5. Conclusions

In this paper, we proposed a novel single-stage detector framework Bidirectional Feature Pyramid Networks (BPN) for high-quality object detection. It comprises two novel major components: a Bidirectional Feature Pyramid structure for more effective and robust feature representations and an Anchor Refinement component to gradually refine the quality of pre-designed anchors for more effective training. The proposed method achieves state-of-the-art results on Pascal VOC and MSCOCO dataset while enjoying real-time inference speed.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

CRediT authorship contribution statement

Xiongwei Wu: Conceptualization, Methodology, Software, Writing - original draft, Writing - review & editing. **Doyen Sahoo:**

Investigation, Writing - review & editing. **Daoxin Zhang:** Visualization, Software, Writing - original draft. **Jianke Zhu:** Supervision. **Steven C.H. Hoi:** Supervision, Investigation, Writing - review & editing.

References

[1] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: Proceedings of the Advances in Neural Information Processing Systems, MIT Press, 2015.

[2] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, 2014.

[3] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, 2017.

[4] R. Girshick, Fast R-cnn, in: Proceedings of the International Conference on Computer Vision, IEEE, 2015.

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: Single shot multibox detector, in: Proceedings of the European Conference on Computer Vision, Springer, 2016.

[6] J. Redmon, A. Farhadi, Yolo9000: Better, faster, stronger, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, 2017.

[7] S. Zhang, L. Wen, X. Bian, Z. Lei, S.Z. Li, Single-shot refinement neural network for object detection, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, 2018.

[8] Z. Cai, N. Vasconcelos, Cascade r-cnn: Delving into high quality object detection, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, 2018.

[9] P. Viola, M.J. Jones, Robust real-time face detection, International Journal of Computer Vision (2004).

[10] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, 2016.

[11] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, International Conference on Learning Representations (2015).

[12] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the Advances in Neural Information Processing Systems, MIT Press, 2012.

[13] J. Dai, Y. Li, K. He, J. Sun, R-fcn: Object detection via region-based fully convolutional networks, in: Proceedings of the Advances in Neural Information Processing Systems, MIT Press, 2016.

[14] J.R. Uijlings, K.E. Van De Sande, T. Gevers, A.W. Smeulders, Selective search for object recognition, International Journal of Computer Vision (2013).

[15] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real–time object detection, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, 2016.

[16] S. Gidaris, N. Komodakis, LocNet: Improving localization accuracy for object detection, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, 2016.

[17] S. Zagoruyko, A. Lerer, T.-Y. Lin, P.O. Pinheiro, S. Gross, S. Chintala, P. Dollár, A multipath network for object detection, in: Proceedings of the BMVC, British Machine Vision Association, 2016.

[18] S. Bell, C. Lawrence Zitnick, K. Bala, R. Girshick, Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, 2016.

[19] T. Kong, A. Yao, Y. Chen, F. Sun, Hypernet: Towards accurate region proposal generation and joint object detection, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, 2016.

[20] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, A.C. Berg, DSSD: Deconvolutional single shot detector, arXiv:1701.06659 (2017).

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, 2015.

[22] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, JMLR, 2010.

[23] A. Shrivastava, A. Gupta, R. Girshick, Training region-based object detectors with online hard example mining, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, 2016.

[24] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, H. Lu, Couplenet: Coupling global structure with local parts for object detection, in: Proceedings of the International Conference on Computer Vision, IEEE, 2017.

[25] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, Y. Chen, Ron: Reverse connection with objectness prior networks for object detection, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, 2017.

[26] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, X. Xue, Dsod: Learning deeply supervised object detectors from scratch, in: Proceedings of the International Conference on Computer Vision, IEEE, 2017.

[27] S. Liu, D. Huang, a. Wang, Receptive field block net for accurate and fast object detection, in: Proceedings of the European Conference on Computer Vision, Springer, 2018.

[28] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al., Speed/accuracy trade-offs for modern convolutional object detectors, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, 2017.

[29] L. Tychsen-Smith, L. Petersson, Denet: Scalable real-time object detection with directed sparse sampling, in: Proceedings of the International Conference on Computer Vision, IEEE, 2017.

[30] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, Proceedings of the International Conference on Computer Vision, IEEE, 2017.

[31] H. Xu, X. Lv, X. Wang, Z. Ren, R. Chellappa, Deep regionlets for object detection, Springer, 2018.

[32] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-cnn, in: Proceedings of the International Conference on Computer Vision, IEEE, 2017.

[33] N. Bodla, B. Singh, R. Chellappa, L.S. Davis, Soft-NMS – improving object detection with one line of code, Proceedings of the IEEE International Conference on Computer Vision, IEEE, 2017.

[34] L. Tychsen-Smith, L. Petersson, Improving object localization with fitness NMS and bounded iou loss, Proceedings of the Computer Vision and Pattern Recognition, IEEE (2018).

[35] T. Kong, F. Sun, W. Huang, H. Liu, Deep feature pyramid reconfiguration for object detection, in: Proceedings of the European Conference on Computer Vision, Springer, 2018.

[36] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the International Conference on Computer Vision, IEEE, 2017.

[37] X. Zhou, J. Zhuo, P. Krähenbühl, Bottom-up object detection by grouping extreme and center points, IEEE, 2019.

[38] Z. Tian, C. Shen, H. Chen, T. He, FCOS: fully convolutional one-stage object detection, Proceedings of the International Conference on Computer Vision, IEEE, 2019.

[39] T. Kong, F. Sun, H. Liu, Y. Jiang, J. Shi, Foveabox: Beyond anchor-based object detector, arXiv:1904.03797 (2019).

[40] X. Zhou, D. Wang, P. Krähenbühl, Objects as points, in: arXiv:1904.07850 2019.

[41] H. Law, J. Deng, Cornernet: Detecting objects as paired keypoints, in: Proceedings of the European Conference on Computer Vision, Springer, 2018.

[42] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: Proceedings of the European Conference on Computer Vision, Springer, 2014.

Xiongwei Wu received the bachelor’s degree in computer science from Zhejiang University, Zhejiang, P.R. China. He is currently the PhD student in the School of Information Systems, Singapore Management University, Singapore, supervided by Prof. Steven Hoi. His research directions mainly focus on object detection and deep learning.

Doyen Sahoo is a Research Scientist at Salesforce Research Asia. Prior to this, he was serving as Adjunct faculty in Singapore Management University, and was also a Research Fellow at the Living Analytics Research Center. He works on Online Learning, Deep Learning and related machine learning applications. He obtained his PhD from Singapore Management University, and B.Eng from Nanyang Technological University.

Daoxin Zhang is a master student in College of Computer Science and Technology, Zhejiang University supervised by Professor Jianke ZHU. He is also Research assistant in School of Information Systems, Singapore Management University supervised by Associate Professor Steven C.H. HOI. His primary research topic is deep learning based computer vision including recognition, segmentation and super resolution etc. Prior to starting graduate study, Daoxin completed his B.Eng. in Mathematics from Zhejiang University.

Jianke Zhu is an Associate Professor in College of Computer Science at Zhejiang University. He received his Ph.D. degree in Computer Science and Engineering from The Chinese University of Hong Kong. He was a postdoc in BIWI Computer Vision Lab at ETH Zurich. His research interests include computer vision and multimedia information retrieval. He is a senior member of the IEEE.

Prof. Steven C.H. Hoi is currently Managing Director of Salesforce Research Asia at Salesforce, located in Singapore. He has been also a tenured Associate Professor of the School of Information Systems at Singapore Management University, Singapore. Prior to joining SMU, he was a tenured Associate Professor of the School of Computer Engineering at Nanyang Technological University, Singapore. He received his Bachelor degree in Computer Science from Tsinghua University, Beijing, China, in 2002, and both his Master and Ph.D degrees in Computer Science and Engineering from the Chinese University of Hong Kong, in 2004 and 2006, respectively.