Single-Shot Bidirectional Pyramid Networks for High-Quality Object Detection

Xiongwei Wu[†], Daoxin Zhang[†‡], Jianke Zhu[‡], Steven C.H. Hoi[†§]
[†]School of Information Systems, Singapore Management University, Singapore
[‡]College of Computer Science and Technology, Zhejiang University, Hangzhou, China
[§]DeepIR Inc., Beijing, China
{chhoi,dxzhang,xwwu.2015@phdis}@smu.edu.sg;{dxzhang,jkzhu}@zju.edu.cn

Abstract

Recent years have witnessed many exciting achievements for object detection using deep learning techniques. Despite achieving significant progresses, most existing detectors are designed to detect objects with relatively low-quality prediction of locations, i.e., often trained with the threshold of Intersection over Union (IoU) set to 0.5 by default, which can yield low-quality or even noisy detections. It remains an open challenge for how to devise and train a high-quality detector that can achieve more precise localization (i.e., IoU>0.5) without sacrificing the detection performance. In this paper, we propose a novel single-shot detection framework of Bidirectional Pyramid Networks (BPN) towards high-quality object detection, which consists of two novel components: (i) a Bidirectional Feature Pyramid structure for more effective and robust feature representations; and (ii) a Cascade Anchor Refinement to gradually refine the quality of predesigned anchors for more effective training. Our experiments showed that the proposed BPN achieves the best performances among all the single-stage object detectors on both PASCAL VOC and MS COCO datasets, especially for high-quality detections.

1. Introduction

Object detection is one of fundamental research problems in computer vision and has been extensively studied in literature[22, 8, 16]. Recent years have witnessed remarkable progresses for object detection after exploring the family of powerful deep learning techniques. Currently, the state-of-the-art deep learning based object detection frameworks can be generally divided into two major groups: (i) two-stage detectors, such as the family of Region-based CNN (R-CNN) and their variants[7, 22, 8] and (ii) one-stage detectors, such as SSD and its variants[21, 19]. 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 the 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, i.e., with a default IoU threshold of 0.5. When the goal is to achieve higher quality localization precision (IoU>0.5), the detection performance often drops significantly[2]. One naive solution is to increase the IoU threshold during training. This however is not effective since a high IoU will lead to significantly less amount of positive training samples and thus make the training results prone to overfitting, especially for single-shot SSD-like detectors. This work is motivated to investigate an effective single-shot detection scheme towards high-quality object detection.

In this paper, we aim to develop a novel high-quality detector by following the family of single-stage SSD-like detectors due to their significant advantage in computational efficiency. In particular, we realize that the existing SSD-style detector has two critical drawbacks for high-quality object detection tasks. First, the single-shot feature representations may not be discriminative and robust enough for precise localization. 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). As a summary, our main contributions include the following:

- A novel framework of Bidirectional Pyramid Networks (BPN) for single-shot object detector that is designed directly towards high-quality detection;

- A novel Bidirection Feature Pyramid Structure that improves the vanilla Feature Pyramid by adding a Reverse Feature Pyramid in order to fuse both deep and shadow features towards more effective and robust representations;
- A novel Cascaded Anchor Refinement scheme to gradually improve the quality of predefined anchors which are often inaccurate at the beginning;
- 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 maintaining the advantage of computational efficiency.

2. Related Work

Object detection has been extensively studied for decades [29, 8, 4]. In early stage of research studies, object detection was based on sliding windows, and dense image grids are encoded with hand-crafted features followed by training classifiers to find and locate objects. Viola and Jones [29] proposed a pioneering cascaded classifiers by AdaBoost with Haar feature 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, 25, 15], deep learning based approaches have been actively explored for object detection, especially for the region-based convolutional neural networks (R-CNN) and its variants [22, 16, 7]. Currently deep learning based detectors can be generally divided into two groups: (i) two-stage RCNN-based methods and (ii) one-stage SSD-based methods. RCNN-based methods, such as RCNN [8], Fast RCNN [7], Faster RCNN [22], R-FCN [3], first generate a sparse set of proposals followed by region classifiers and location regressors. Two-stage detectors usually achieve better detection performance and report state-of-the-art results on many common benchmarks because the proposals are often carefully generated (e.g., by selective search [28] or RPN [22]) and usually well match the target objects. However, they often suffer from very slow inference speed due to the expensive two-stage detection approach. Unlike the two-stage RCNN-based methods, SSD-style methods, such as SSD [19], YOLO [20], YOLOv2 [21]), ignores the proposal generation step by directly making prediction with manually pre-defined anchors and thus reduce the inference time significantly towards real-time speed. However, the manually predefined anchors are often sub-optimal and sometimes ill-designed, in which few of them can tightly match the objects. Thus, SSD-style detectors [19] are difficult to precisely locate objects towards high-quality detection.

In literature, most object detection studies were focused on the detection with relatively low localization quality, with a default IoU threshold of 0.5. There are only a few related studies for high-quality detection. LocNet[6] learns a postprocessing network for location refinement, which however does not optimize the whole system end-to-end and is not designed for high-quality detection tasks. MultiPath Network [30] proposed to learn multiple detection branches for different quality thresholds. However, it still suffers insufficient training samples and it is computationally slow due to the nature of two-stage detectors. Cascaded RCNN [2] learns regressors in a cascaded way, which gradually increases qualified proposal numbers towards high-quality detection. However, it is still based on two-stage RCNN and its slow inference speed is a critical drawback, especially the feature re-extraction step is operated by time-consuming ROI Pooling or ROI Warping.

Our work is also related to the studies for multi-scale feature fusion, which has been proved to be an effective and important structure for object detection with different scales. ION [1] extracts region features from different layers by ROI Pooling operation; HyperNet [14] directly concatenates features at different layers using deconvolution layers. FPN [16] and DSSD[5] fuses features of different scales with lateral connection in a bottom-up manneer, which effectively improve the detection of small objects. However, the vanilla feature pyramid [16] only considers boosting shallow layer features with deep layer features, but ignores the fact that the instance information in shallow layer features can be helpful to deep semantic layer features. We overcome this limitation by the proposed Bidirectional Feature Pyramid structure.

3. Single-Shot Detector for High-Quality Detection

3.1. Motivation

Our goal is to investigate single-shot detectors for high-quality object detection tasks. For existing object detectors, a group of anchors are often generated/pre-defined on the feature maps densely or sparsely, followed by location regression and object category classification. The object class label of each anchor is assigned according to anchor's jaccard overlap with objects. Two-stage RCNN-based detectors generate anchors in their first step, and assign positive label to anchors whose overlaps with objects are higher than IoU threshold. However, in one-stage SSD-based detectors, anchors are manually designed and thus the majority of these anchors fail to match objects with qualified IoU threshold(0.5 etc.). This problem becomes more severe in training detectors for high IoU thresholds(0.7 etc.) since the number of positive anchors decreases significantly as IoU thresholds increase, and it leads to overfitting problem.

Table 1: Average matched anchor number per image whose jaccard overlaps with objects are higher than IoU Threshold. Our cascaded anchor refiner improves anchor quality gradually. Underlined entry is the number we use for training(discussed in section4.2).

IoU Threshold	Original SSD	FPN(once refined)	BPN(twice refined)
0.5	**13.85**	191.11	383.09
0.6	5.08	**141.68**	316.86
0.7	3.01	100.52	**252.16**
0.8	2.84	62.12	176.26
0.9	2.84	22.18	64.21

In the first column of Table 1, we count the number of positive anchors per image for different IoU thresholds in training SSD-style detector. In default setting(IoU threshold is 0.5), the positive anchor number per image is only 13.85. When we increase IoU threshold from 0.5 to 0.7, only 3 positive anchors left for training, which cannot provide sufficient information to effectively train detectors. Our motivation is to improve anchor quality by cascaded refine predesigned anchor in different predict levels. The second and third column in Table 1 illustrate the matched anchor number after once and twice being refined respectively. With sufficient matched anchors, we are able to train high quality detector in Bi-directional feature pyramid. Please refer Section 4.2 for details.

3.2. Proposed Framework of Bidirectional Pyramid Networks

In this paper, we propose a novel framework of Bidirectional Pyramid Networks (BPN) to overcome the above drawbacks of SSD-style detectors towards high-quality detection tasks. In particular, to address the weak feature representation issue of SSD-style detectors, the idea is to explore the structure of Feature Pyramid Networks (FPN) [16] in improving the typical SSD-style feature representations, in which we propose a novel Bidirectional Feature Pyramid structure that can further boost the effectiveness of FPN structure. To address the anchor quality issue, the key idea and challenge is to devise an effective yet efficient scheme for refining the quality of the anchors before training the classifiers and regressors, in which we explore a cascade learning and refinement approach without suffering the computational drawback of two-stage detectors.

Figure 1 gives an overview of the proposed single-shot Bidirectional Pyramid Networks (BPN) for high-quality object detection, where the backbone network can be any typical CNN network, such as Alexnet[15], GoogleNet[27], VGG[25], ResNet[10], etc, as shown in the blue branch of Figure 1. For simplicity, we choose VGG-16 as backbone network in our study.

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 bottom-up (the purple branch of Figure 1) and reverse feature pyramid in top-down (the green branch of Figure 1). These three-level branches not only aggregate multi-level features to provide robust feature representations, but also enable multi-quality training and cascaded Anchor refinement. For the joint training with multiple quality levels, the Cascaded Anchor Refinement optimizes anchors from the previous branch and send them to the next branch.

The above two key components, Bidirectional Feature Pyramid and Cascaded Anchor Refinement, are nicely integrated in the framework and can be trained end-to-end to achieve high-quality detections in a coherent and synergic manner. In the following, we present each of these two components in detail.

3.3. Cascaded Anchor Refinement

We denote the depth of feature maps for prediction as L, where $L \in \{1,2,3,4\}$ in our settings, and the levels of quality $Q \in \{1,2,3,\ldots\}$ with the corresponding IoU thresholds as $\mathrm{IoU}(Q) \in \{0.5, 0.6, 0.7, ...\}$. 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 is 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 Figure 1 for details).

In order to increase the number of positive anchors and improve their quality as well, we denote the Cascaded Anchor Refinement ("CAR") used in quality Q, depth L as CAR_L^Q. In particular, CAR has two parts: location regressor Reg_L^Q and categorical classifiers Cls_L^Q. At each level of quality, regressors receive the processed anchors from the previous level of quality for further optimization (A_L^1 is the manually defined anchors):

$$A_L^Q = Reg^Q(A_L^{Q-1}; F_L^Q), \quad Q = 2,3,\ldots, L = 1,2,\ldots \tag{1}$$

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

$$C_L^Q = Cls^Q(F_L^Q), \quad Q = 1,2,3\ldots, L = 1,2,\ldots \tag{2}$$

Therefore, the training loss at quality level Q can be represented as:

$$L^Q = \frac{1}{N_{\text{Q}}} * \sum_L \sum_i \left(L_{\text{Cls}}^Q(\{C_{L_i}^Q\}, \{l_{L_i}\}) + \lambda * L_{\text{Reg}}^Q(\{A_{L_i}^Q\}, \{g_{L_i}\}) \right) \tag{3}$$

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

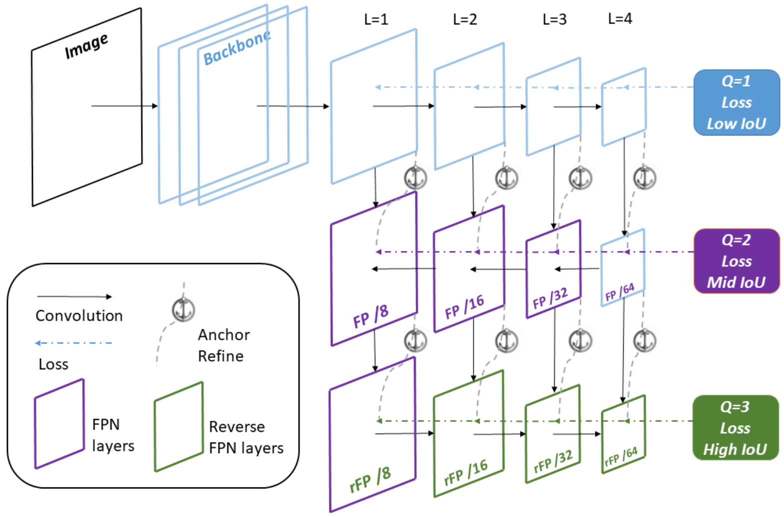


Figure 1: The proposed framework of Bidirectional Pyramid Networks (BPN) for single-shot high-quality detection. The Cascaded Anchor Refinement (*CAR*) are utilized for relocating anchors, each of which is responsible for a certain quality level of detection. Training sample quality increases when anchor refinement cascades (with higher IoU). *FPN* denotes Feature Pyramid block, and *rFPN* denotes Reverse Feature Pyramid block.

a mini-batch, l_{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. $L^Q_{\text{Cls}}(.)$ is softmax loss function over multiple classes confidences and $L^Q_{\text{Reg}}(.)$ is the Smooth L1-loss which is also used in [19]. The total training loss is the summation of losses at all the quality levels:

$$L_{\text{BPN}} = \sum_Q L^Q \tag{4}$$

3.4. Bidirectional Feature Pyramid Structure

In order to improve the power of feature representation of SSD-style detectors, we apply Feature Pyramid Networks (FPN) [16], which exploits the inherents multi-scale, pyramidal hierarchy of deep convolutional networks to construct the representation of feature pyramids with marginal extra cost. Specifically, FPN fuse deep semantically-strong features with shallow semantically-weak but instance-strong features. However, we found aggregating features in the reverse direction is of great importance similarly. This results in the proposed Bidirectional Feature Pyramid structure that consists of both FPN and reverse FPN to make the feature representation considerably more effective and robust.

Reverse FPN can enjoy the merit from following aspects: 1). Compared with stacked CNN for image classification, reverse FPN reduces the *distance* from shallow features to deep features by using much fewer convolution filters and thus effectively keeps spatial information; 2) Lateral connections *reuse* different shallow layer features to reduce information attenuation from shallow features to deep features; 3) Our cascaded-style detector structure can naturally use this structure. Figure 2 gives the illustration of the proposed Bidirectional Feature Pyramid structure.

Specifically, Figure 2(a) is the vanilla Feature Pyramid block that fuses features in bottom-up with lateral connection. It worth noticing that there is no strengthen of the deepest feature layer from Feature Pyramid (the right diagram of Figure1). Thus, we further build the Reverse Feature Pyramid by top-down aggregation (as shown in Figure 2 (b)) with lateral connection to enhance deep layer features. The formulation of Feature Pyramid (FP) and reverse Feature Pyramid (rFP) can be represented as:

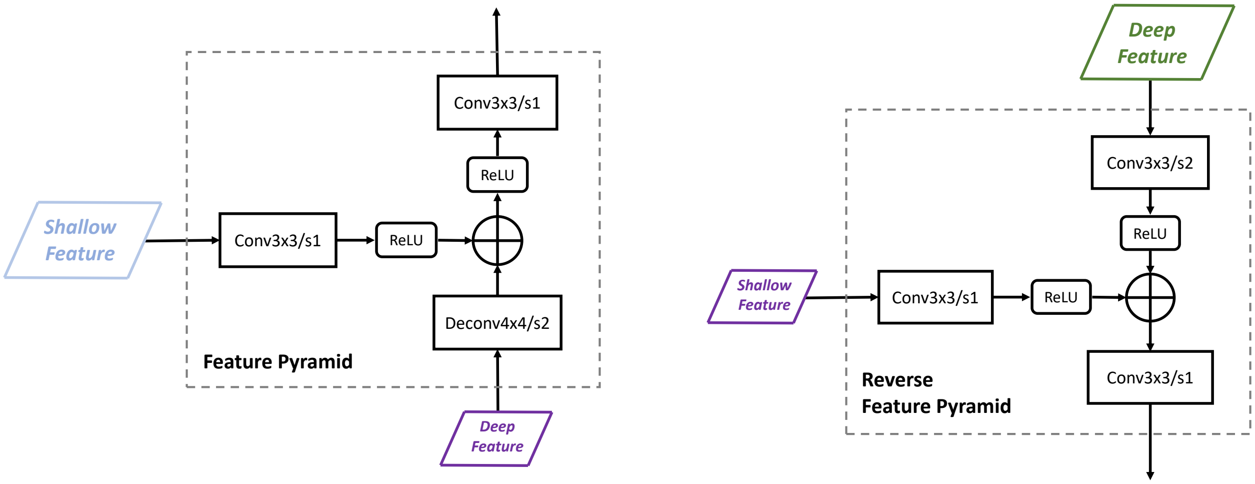
$$\text{FP}: \quad F^Q_L = \text{Deconv}_{s2}(F^Q_{L+1}) \oplus \text{Conv}(F^{Q-1}_L) \tag{5}$$

$$\text{rFP}: \quad F^Q_L = \text{Conv}_{s2}(F^Q_{L-1}) \oplus \text{Conv}(F^{Q-1}_L) \tag{6}$$

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.

3.5. Implementation Details

Backbone Architecture: We choose VGG16 [25] pre-trained on ImageNet as the backbone network in our experiments. We follow [19] to transform the last two fully-connected layers "fc6" and "fc7" to convolutional layers



(a) Feature Pyramid

(b) Reverse Feature Pyramid

Figure 2: The proposed Bidirectional Feature Pyramid Network Structure

"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 (dabbed 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 to stable the training process.

Data Augmentation: We follow the data augmentation strategies in [19] to make the detectors robust to objects with multiple scales and colors. Specifically, images are randomly expanded or cropped with additional photometric distortion to generate more 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 VGG16 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 and 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 etc.)

Anchor Design: Originally a group of anchors are predesigned manually. For each prediction feature block, one scale-specific set of anchors with three aspect ratios are associated. In our approach, we set the scale of anchors as 4 times 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 Details: We use the "xavier" method in [9] to randomly initialize the parameters in extra added layers in VGG16. We set the mini-batch size as 32 in training and the whole network is optimized via the SGD optimizer. We set the momentum to 0.9 and weight decay to 0.005. The initial learning rate is set to 0.001. The learning policy is different for different datasets. For PASCAL VOC dataset, the models are totally finetuned for 120k iterations and we decrease the learning rate to 1e-4 and 1e-5 after 80k and 100k iterations, respectively. For MSCOCO dataset, the models are finetuned for 400k iterations and we decrease the learning rate to 1e-4 and 1e-5 after 280k and 360k iterations, respectively . All the detectors are optimized end-to-end.

Sampling Strategy: The ratio of positive and negative anchors are imbalanced after anchor matching step, so proper sampling strategy is necessary to stable training process. In this paper, we sample a subset of negative anchors to keep the ratio of positive and negative anchors as 1:3 in training process. In order to stable training and fast convergency, instead of randomly sampling negative anchors, we sort the negative anchors according to their confidence loss values and select the hardest ones for training. In different quality levels, the IoU thresholds for training are different. In this paper, we set three quality levels: low, mid and high qualities with IoU thresholds as 0.5, 0.6 and 0.7.

Inference: During the inference phase, different quality CARs make prediction and send the refined anchors into next quality level. We take the predictions from CARs in all qualities, to make sure that it is suitable for all the low-, mid- and high-quality detection.

4. Experiments

We conduct extensive experiments on two public benchmarks Pascal VOC and MSCOCO. Pascal VOC has 20 categories and MSCOCO has 80 categories. The evaluation metric is mean average precision which is widely used in evaluating object detection. The whole framework is implemented in Caffe[12] Platform.

4.1. Pascal VOC

We use Pascal VOC2007 trainval set and Pascal VOC2012 trainval set as our training set, and VOC2007 test set as testing set, with overall 16k images for training and 5k images for testing. All models are based on VGG16 architecture since ResNet-101 has limited gain in this dataset[5].

We set BPN with two resolutions input(320x320 and 512 x512) 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). From Table 2, our BPN320 gets 80.3%, 75.5% and 66.1% accuracy in low, mid and high quality detection scenario, which has already outperformed many detectors(SSD320 and Faster RCNN etc.). Further we build BPN512 by increasing input size to 512 and BPN512 gets 81.9%, 77.6% and 68.3% in three quality scenarios, which are state-of-the-art results. Our BPN is one-stage detector, so it can also enjoy the merit for real time inference. BPN320 can make inference with 32.4fps while BPN512 with 18.9fps on Titan XP GPU cards, which has significant strength over two-stage detectors. Notably, BPN has very clear advantage in high quality detection scenario(IoU=0.7).

4.2. Ablation Study

In this section, we conduct a series of ablation study 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 listed in Table 3.

Proposal Quality Improved by CAR: In this section, we validate the effectiveness of CAR blocks to improve anchor quality. In Table 1, we count the positive anchor numbers per image on different IoU thresholds in original SSD, FPN and BPN respectively. In original SSD, anchors are generated manually and only a few anchors matched objects, which is hard to train detectors effectively. In FPN anchors have been refined by CAR once, and the matched number increases significantly in all IoU thresholds. Further in BPN where anchors has been refined by CAR twice, more high quality anchors are generated. Notably, after refined by CAR we have sufficient positive training samples in high quality levels so that we could conduct gradually increasing training positive IoU thresholds (0.5, 0.6 and 0.7). This experiment shows our CAR blocks can gradually improve anchor qualities and generates more qualified training samples.

Bidirectional Feature Pyramid: To validate the effectiveness of the Bidirectional Feature Pyramid, we remove CAR from BPN and compare this model(dabbed as BPN w / o CAR) with vanilla SSD and SSD w / FPN. Bidirectional Feature Pyramid is built based on vanilla SSD and all three models are fine-tuned with IoU threshold as 0.5. In Table 3, SSD w / FPN outperforms vanilla SSD because deep semantic features boost feature representations. Further, BPN w / o CAR outperforms SSD w / FPN in all quality scenarios, which proves the effectiveness of Bidirectional Feature Pyramid.

Level of Cascaded: In this section, we validate the cascaded level of CAR is important in training high quality detectors. We list the results in Table 3. Firstly, a vanilla SSD model is 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 the fact that insufficient positive training samples cause overfitting problem. Secondly, we keep only one CAR block of BPN(dabbed as BPN w / AR), and train this model with 0.5 IoU threshold. The results show the detection results improves significantly compared with BPN w / o CAR in low and mid quality scenarios, but not obvious in high quality case(63.6% vs 63.4%). We further train BPN w / AR with 0.7 IoU threshold and this model(Row 6) also presents overfitting problem but less severe compared with vanilla SSD. It represents the fact that anchor refiner can boost detection performance by refine anchor quality but only one refiner could not directly boost the model. Thirdly, considering results above, we add two more CAR blocks and joint optimize CAR with different quality settings (0.5,0.5,0.7) and (0.5,0.6,0.7), which utilize high quality anchors for training. This two models(Row 7 and Row 8) achieve evident growth especially in high quality scenario(IoU=0.6 and IoU=0.7, etc.). In conclusion, a single Anchor Refiner is very effective in addressing overfitting problem in SSD model but to improve the detection performance in high quality scenarios, Cascaded Anchor Refiner(CAR) is required.

4.3. MSCOCO

In addition to PASCAL VOC, we also evaluate BPN on MSCOCO [18]. COCO contains 80 classes objects and about 120k images in `trainval` set. We use `trainval35k` set for training and test on `test-dev` set.

Table 4 shows the results on MS COCO test-dev set. BPN320 with VGG-16 achieves 29.6% AP and when using larger input image size 512, the detection accuracy of BPN 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. Because the objects in COCO dataset are with various scales, so we also applied multi-scale testing based on BPN320 and BPN512 to reduce the impact of input size. The improve version BPN320++ and BPN512++ achieves 35.4%

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. For some algorithms there are no public released model, so we emit the results with IoU with 0.7. Bold fonts indicate the best mAP.

Method	Backbone	Input size	FPS	mAP (%)		
				IoU@0.5	IoU@0.6	IoU@0.7
two-stage:						
Fast R-CNN [7]	VGG-16	$\sim 1000 \times 600$	0.5	70.0	62.4	49.4
Faster R-CNN [22]	VGG-16	$\sim 1000 \times 600$	7	73.2	67.7	54.4
OHEM [24]	VGG-16	$\sim 1000 \times 600$	7	74.6	68.9	55.9
HyperNet [14]	VGG-16	$\sim 1000 \times 600$	0.88	76.3	-	-
Faster R-CNN [10]	ResNet-101	$\sim 1000 \times 600$	2.4	76.4	69.5	57.3
ION [1]	VGG-16	$\sim 1000 \times 600$	1.25	76.5	-	-
LocNet [6]	VGG-16	$\sim 1000 \times 600$	-	77.5	-	64.5
R-FCN [3]	ResNet-101	$\sim 1000 \times 600$	9	80.5	73.2	61.8
CoupleNet [32]	ResNet-101	$\sim 1000 \times 600$	8.2	81.7	76.6	66.8
one-stage:						
YOLO [20]	GoogleNet [27]	448×448	45	63.4	-	-
RON384 [13]	VGG-16	384×384	15	75.4	66.8	54.2
SSD300 [19]	VGG-16	300×300	46	77.3	72.3	61.3
DSOD300 [23]	DS/64-192-48-1[23]	300×300	17.4	77.7	73.4	63.6
YOLOv2 [21]	Darknet-19	544×544	40	78.6	69.1	56.5
SSD512 [19]	VGG-16	512×512	19	79.8	74.7	64.0
RefineDet320 [31]	VGG-16	320×320	40.3	80.0	74.2	63.6
RefineDet512 [31]	VGG-16	512×512	24.1	81.8	76.9	66.0
BPN320(ours)	VGG-16	320×320	32.4	80.3	75.5	66.1
BPN512(ours)	VGG-16	512×512	18.9	**81.9**	**77.6**	**68.3**

Table 3: 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. 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 w / FPN	0.5	77.4	72.1	61.6
BPN w / o CAR	0.5	78.1	72.7	63.4
BPN w / AR	0.5	80.0	74.2	63.6
BPN w / AR	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**

and 37.9% AP, which is the state-of-the-art performance of one-stage detectors.

Different from Pascal VOC, deeper backbone such as ResNet can further improve detection accuracy than VGG16 model. As [31] claimed, batch normalization in ResNet requires at least four images per GPU to obtain precise statistic information and stable training process. And its deep network structure further enlarges the memory and computation cost. Thus, we only report the results of VGG16-based model due to GPU memory and computation limitation.

4.4. Quantitative Results and Error Analysis

We show some quantitative results on Pascal VOC in figure 3. The improvement of Cascaded Anchor Refiner and the error analysis are shown in this section. We compare the results of BPN with other single-stage detector RefineDet[31] and SSD[19]. Results of BPN are illustrated as blue boxes. Red and Green boxes present the results from RefineDet[31] and SSD[19] respectively. These models are trained with backbone VGG16 and with the same training setting. It can be found that the predictions of BPN match the object boundary more precisely. Also in some hard cases SSD or RefineDet fail to detect objects while BPN can still work due to stronger features.

We also analyze the performance of BPN320 by the detection analysis tool to better understand the detection result as well as for further improvement. In Figure 4, the first row shows the percentage of error type of the top false positive. The second row shows the fraction of detections

Table 4: Detection results on MS COCO `test-dev` set. Bold fonts indicate the best performance.

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
two-stage:							
Fast R-CNN [7]	VGG-16	19.7	35.9	-	-	-	-
Faster R-CNN [22]	VGG-16	21.9	42.7	-	-	-	-
OHEM [24]	VGG-16	22.6	42.5	22.2	5.0	23.7	37.9
ION [1]	VGG-16	23.6	43.2	23.6	6.4	24.1	38.3
OHEM++ [24]	VGG-16	25.5	45.9	26.1	7.4	27.7	40.3
R-FCN [3]	ResNet-101	29.9	51.9	-	10.8	32.8	45.0
CoupleNet [32]	ResNet-101	34.4	54.8	37.2	13.4	38.1	50.8
Faster R-CNN by G-RMI [11]	Inception-ResNet-v2[26]	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 [16]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN w Cascade RCNN [2]	VGG16	26.9	44.3	27.8	8.3	28.2	41.1
R-FCN w Cascade RCNN [2]	ResNet-50	30.9	49.9	32.6	10.5	33.1	46.9
R-FCN w Cascade RCNN [2]	ResNet-101	33.3	52.6	35.2	12.1	36.2	49.3
one-stage:							
YOLOv2 [21]	DarkNet-19[21]	21.6	44.0	19.2	5.0	22.4	35.5
SSD300 [19]	VGG-16	25.1	43.1	25.8	6.6	25.9	41.4
RON384++ [13]	VGG-16	27.4	49.5	27.1	-	-	-
SSD321 [5]	ResNet-101	28.0	45.4	29.3	6.2	28.3	49.3
DSSD321 [5]	ResNet-101	28.0	46.1	29.2	7.4	28.1	47.6
SSD512 [19]	VGG-16	28.8	48.5	30.3	10.9	31.8	43.5
SSD513 [5]	ResNet-101	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [5]	ResNet-101	33.2	53.3	35.2	13.0	35.4	51.1
RefineDet320 [31]	VGG-16	29.4	49.2	31.3	10.0	32.0	44.4
RefineDet512 [31]	VGG-16	33.0	54.5	35.5	16.3	36.3	44.3
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

that are correct (Cor) or different false positive types: localization issue(Loc), confusion with similar categories (Sim), with background (BG) or other errors(Oth). In the first row, BPN320 produces much less background errors and localization errors compared with vanilla SSD and RefineDet. The bi-directional feature pyramid strongly boost feature maps and thus is more robust to complex background information. The cascaded anchor refiners optimize object locations which leads to more high quality predictions. However, we notice confusion with similar categories error(Sim) occupies high error ratio in BPN. Similar categories(cow and cats etc.) make the detector confused and it means the sampling strategy is not optimal. In this paper, we adopt the same sampling method with previous work[19] and we argue more effective sampling strategy such as Focal Loss[17] can further improve the detection results. We leave this as future work. The second row of Figure 4 indicates the majority of BPN's confident detections are correct.

5. Conclusions

This paper proposed a novel single-stage detector framework cascaded Bidirectional Feature Pyramid Networks (BPN) towards high-quality object detection with two major components: a Bidirectional Feature Pyramid structure for more effective and robust feature representations and a Cascade Anchor Refinement to gradually refine the quality of predesigned anchors for more effective training. The proposed method achieves state-of-the-art results on Pascal VOC and MSCOCO dataset with real-time inference speed. Future work includes more empirical studies on better backbone networks and other object detection tasks.

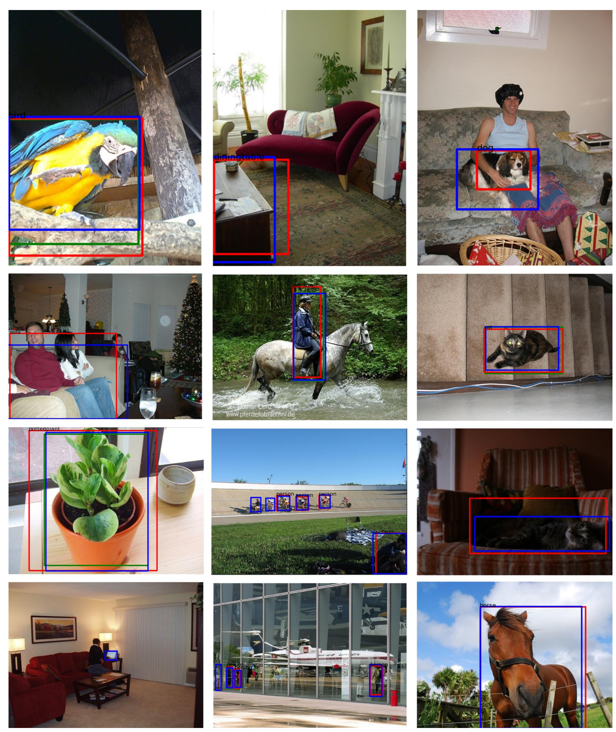


Figure 3: Qualitative detection results of our method (blue) on Pascal VOC compared with other single-shot methods RefineDet [31](red) and SSD [19](green). Bounding boxes with confidence scores less than 0.4 are ignored. If there is a missing color, it indicates the corresponding detector fails to detect the object.

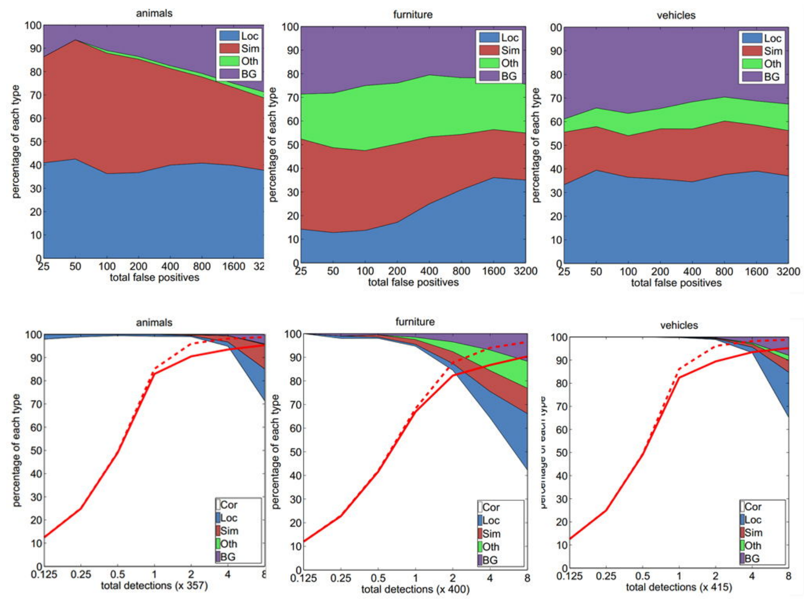


Figure 4: Visualization of error analysis for the proposed BPN320 detector on “animals”, “vehicles”, and “furniture” classes on the VOC 2007 test set.

References

[1] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 7, 8

[2] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. *arXiv preprint arXiv:1712.00726*, 2017. 1, 2, 8

[3] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems*, 2016. 2, 7, 8

[4] 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*, 2010. 2

[5] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017. 2, 6, 8

[6] S. Gidaris and N. Komodakis. Locnet: Improving localization accuracy for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 7

[7] R. Girshick. Fast r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, 2015. 1, 2, 7, 8

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1, 2

[9] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010. 5

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 3, 7, 8

[11] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 8

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of 22nd ACM intl. conference on Multimedia*, 2014. 6

[13] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen. Ron: Reverse connection with objectness prior networks for object detection. *arXiv preprint arXiv:1707.01691*, 2017. 7, 8

[14] T. Kong, A. Yao, Y. Chen, and F. Sun. Hypernet: Towards accurate region proposal generation and joint object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 7

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012. 2, 3

[16] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 3, 4, 8

[17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017. 8

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. 6

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *European Conference on Computer Vision*, 2016. 1, 2, 4, 5, 7, 8

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 7

[21] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016. 1, 2, 7, 8

[22] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015. 1, 2, 7, 8

[23] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, and X. Xue. Dsod: Learning deeply supervised object detectors from scratch. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017. 7

[24] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 7, 8

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2, 3, 4

[26] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017. 8

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3, 7

[28] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *Intl. Journal of Computer Vision (IJCV)*, 2013. 2

[29] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 2004. 2

[30] S. Zagoruyko, A. Lerer, T.-Y. Lin, P. O. Pinheiro, S. Gross, S. Chintala, and P. Dollár. A multipath network for object detection. *arXiv preprint arXiv:1604.02135*, 2016. 2

[31] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li. Single-shot refinement neural network for object detection. *arXiv preprint arXiv:1711.06897*, 2017. 7, 8

[32] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, and H. Lu. Couplenet: Coupling global structure with local parts for object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017. 7, 8