Visual Object Detection with Score Refinement

Tao Li[1], Yu Wang[1], Zheng Zhang[1*], Xuezhuan Zhao[2], Lishen Pei[3]

[1] Department of Information Engineering, Open University of Henan, China
[2] Department of Information Engineering, Zhengzhou University of Aeronautics, China
[3] Department of Information Engineering, Henan University of Economics and Law, China
cvlablitao@163.com, cswangyuu@126.com, zhang_zhengfly@163.com, xuezhuansci@126.com, pls.cvlab@gmail.com

Abstract

Robustness of object detection against hard samples, especially small objects, has long been a critical and difficult problem that hinders development of convolutional object detectors. To address this issue, we propose Progressive Refinement Network to reduce classification ambiguity for scale robust object detection. In PRN, several orders of residuals for the class prediction are regressed from upper level contexts and the residuals are progressively added to the basic prediction stage by stage, yielding multiple refinements. Supervision signal is imposed at each stage and an integration of all stages is performed to obtain the final score. By supervision retaining through the context aggregation procedure, PRN avoids over dependency on higher-level information and enables sufficient learning on the current scale level. The progressive residuals added for refinements adaptively reduce the ambiguity of the class prediction and the final integration of all stages can further stabilize the predicted distribution. PRN achieves 81.3% mAP on the PASCAL VOC 2007 dataset and 31.7% AP (15.6% APS) on MS COCO dataset, which demonstrates the effectiveness and efficiency of the proposed method and its promising capability on scale robustness.

Keywords: Object detection, Scale robustness, Small objects

1 Introduction

The robustness of object detection for hard samples, especially small scale objects, has long been a challenging issue for the community in research of computer vision. Despite of the success in image classification driven by the remarkable representation power of deep convolutional neural networks (CNNs), the object detection task is far from being solved. One main reason is that modern convolutional detectors still have difficulties in dealing with the high ambiguity in classification of hard samples, especially small targets. To alleviate this problem, many approaches have been proposed, which generally include methods based on feature pyramid (multiscale feature fusion) and cascaded detectors.

Among a huge body of research works, the feature pyramid fusion methods (e.g., FPN [1], TDM [2], FSSD [3] and Deep Feature Pyramid Reconfiguration [4]) address the ambiguity in small object detection via aggregating the more semantic deeper features onto shallower layers. Typically a set of feature maps taken from backbone are fused together across scales, yielding a pyramidal representation on which the consequent detection operation is conducted. Although the features for detecting small objects are enhanced by the more discriminative information of upper layers, the fusion procedure itself is less controlled by supervision, which may cause over dependency on higher-level features, leaving the current level insufficiently learned. Moreover, higher-level features are less helpful for locating objects of smaller scale, but they are still involved in localization. For the second solution routine, known as cascaded detectors (like [5] and [6]), predictions are conducted more than once to obtain finer results. However, existing methods only do explicit refinement in the localization procedure. For the classification task, multiple predictions are generated by rescoring at different cascade stages, which means that classification scores at earlier stages are not considered along cascading. Thus the ambiguity in small object classification can not be maximally reduced.

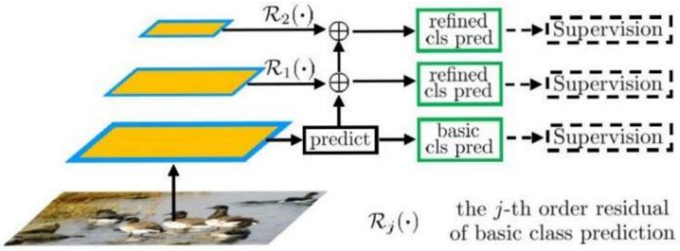

Figure 1. General illustration for the classification mechanism of the proposed approach
(Only the classification process for targets on a certain scale level is shown, whose corresponding feature map is drawn with an extra bolded border. The final score is an integration of different stages for stabilization.)

To avoid over dependency on higher-level features and decrease ambiguity for small objects by explicit modeling on the classification process, in this paper, we propose a novel approach for scale robustness enhancement, denoted as the Progressive Refinement Network (PRN). As shown in Figure 1, for the proposed PRN, classification score (i.e., the input of softmax classifier) is refined (rather than re-predicted) multiple times by progressively adding residuals regressed by upper

*Corresponding Author: Zheng Zhang; E-mail: zhang_zhengfly@163.com
DOI: 10.53106/160792642022092305025

level contexts, and supervision signal for classification is directly retained at each stage of refinement with a particular weight. This group of weights is then reused at test time to integrate the inputs of softmax at all the stages of refinement, which is followed by another softmax to yield the final classification score. In our approach, localization is dedicated to the current feature level or is decoupled from higher-level features.

By retaining supervision signals for classification through the process of context aggregation, PRN avoids over-dependency on higher-level information and enables sufficient optimization for features on lower levels, which we think is critical for detecting small objects. Since we do progressive score refinement rather than re-predict the score for many times, a classification distribution with less ambiguity can be obtained by adaptively adding residuals of the predictions stage by stage. Our integration manner for all the stages of refinement to get the final score is also helpful to stabilize the classification distribution and reduce the side effect of mis-refinement. Finally, decoupled from higher-level features, localization on smaller scales can be conducted on features with more spatial information.

To demonstrate the effectiveness of our approach, we apply PRN to the SSD [7] detection framework, yielding a light weight single stage detector. Experiments are conducted on the PASCAL VOC 2007 dataset [8] and the MS COCO dataset [9]. With PRN, a significant performance gain is obtained in comparison with the baseline. We prove that the proposed method is more effective than feature pyramid fusion and the ambiguity for small object classification is significantly reduced by progressive refinement. Using only the moderate sized VGG-16 network [10] as backbone, PRN reaches 81.3% mAP on the PASCAL VOC 2007 dataset and 31.5 AP, 15.6 APS on the MS COCO dataset. Being comparable with many state-of-the-arts with similar computational budget, the proposed PRN shows an extra promising capability for small object robustness.

Our main contributions are summarized as follows:

1) We propose PRN, which gains improved scale robustness for object detection, by progressively refining the classification score by adding multiple residuals adaptively along context aggregation and controlling the supervision along stages.

2) We apply PRN to the SSD framework and prove that our method is more effective than feature pyramid fusion.

3) We prove that the proposed method contributes to reducing ambiguity of small object classification, and experimental results demonstrate the scale robustness improved by PRN on the PASCAL VOC 2007 dataset and the MS COCO dataset.

This paper is organized as follows. Section 2 reviews related work. Section 3 elaborates our proposed method of object detection with score refinement. Section 4 presents experimental procedures, results, and analysis. Finally, conclusions and discussion are given in Section 5.

2 Related Work

2.1 One-stage Detectors and Two-stage Detectors

Modern convolutional detectors are divided by two main streams, known as the one-stage detectors and the two-stage detectors.

One-stage detectors or single-stage detectors run in a fully convolutional way to regress classification scores and localization offsets for the predefined cell anchors. Then interpreted bounding box predictions are post processed by removing duplicate detection to generate the final results. Stemming from YOLO [11], many detectors of this kind have been developed, such as SSD [7], DSSD [12], YOLO v2 [13], RON [14], RetinaNet [15], RefineDet [16], RBFNN [17] and so on. Since only one forward pass is required to generate all the predictions, one-stage methods are usually fast but less accurate.

Recently, based on one-stage detection paradigm, there has been a new surge of research trend that treats bounding boxes as points or point sets. This starts from Cornernet [18] where object bounding box is detected as a pair of keypoints using a single convolution neural network. Based on Cornernet, Duan et al. [19] proposed to detect each object as a triplet of keypoints. Similarly, Zhou et al. [20] proposed to model an object as a single point, i.e., the center point of its bounding box, which can also be used for 3D detection and pose estimation. Zhang et al. [21] proposed a learning to match approach to break IoU restriction in traditional object anchor IoU, allowing objects to match anchors in a flexible manner. Introduced by the RCNN [22] and Faster RCNN [23] family, two-stage detectors first extract a set of candidate boxes (i.e., region proposals), then classify and localize each proposal using a head subnet. Since the head network is applied many times to process the set of region proposals, two-stage detectors usually have lower detection efficiency but higher accuracy. Typical detectors are SPPNet [24], ION [25], R-FCN [26], Mask R-CNN [27], Light Head R-CNN [28], etc.

2.2 Approaches to Enhance Scale Robustness

Robust object detection for targets of various scales, especially small targets, has been an everlasting challenging problem. To remedy the issue, there have been many works proposed from different aspects. Besides methods based on image pyramid that has become less popular, feature pyramid fusion and cascading are two mainstream solution routines.

2.2.1 Image Pyramid

Enlarging the input scale is a straightforward way to enhance information for small scale targets. The most straightforward approach is known as the multiscale training and testing strategy, which is used in some relatively earlier works, like [3, 23, 29-31]. Recently, a scale normalization technique on image pyramid is proposed by SNIP [32], which improves scale robustness by filtering out extremely small and large samples on the pyramid. But the computational cost of SNIP is still large as an inevitable nature. And it is also because of this computation burden issue that approaches based on image pyramid are seldom adopted and studied recently.

2.2.2 Feature Pyramid Fusion

It is widely perceived that the lack of discriminative information on shallow layers is the main cause of low performance on small targets. Methods based on feature pyramid fusion are widely studied in many recent works by the community.

To start with, Lin et al. [1] proposed Feature Pyramid Network (FPN), which fuses the discriminative features from higher levels onto lower levels by iterative upsampling and element-wise addition. Then TDM [2] is proposed, which changes the fusion operation from element-wise addition to concatenation. Besides the two basic works, there are also some more complicated feature pyramid fusion methods proposed recently, like FSSD [3], Deep Feature Pyramid Reconfiguration [4], Parallel Feature Pyramid Network [33], etc. The common problem of methods based on feature pyramid fusion is that one must obtain a fused feature map first and then conduct object detection on it, without supervision directly retained on lower-level layers. Since higher-level features are easier to discriminate, it is easy to entail overdependence on higher-level information and make features on the exact level insufficiently learned.

2.2.3 Cascade Methods

Improving object detection by multiple predictions is another way to obtain finer results. Currently, most cascade methods are developed for two-stage detectors, like CRC [5], Cascade R-CNN [6] and so on. CRC introduces cascade rejection classifiers that reject easy negatives stage by stage to reduce the number of proposals. But once a sample is incorrectly by re-predicting along cascade stages without an explicit refinement mechanism. At test time, predictions at multiple stages are taken as an ensemble. Though Cascade R-CNN obtains high accuracy, it is also inefficient since it makes the head network more complicated and has an ensemble operation to cover predictions at several stages.

3 Method

3.1 Progressive Refinement

In this part, we introduce the progressive refinement approach, which acts as the key component of our work. In general, the main objective of PSR is to reduce the ambiguity of the predicted classification distributions for hard samples, mostly small targets. For this purpose, we add residuals onto the softmax input of the basic classification prediction stage by stage, yielding several classification predictions which are progressively refined along the procedure. The residuals used to refine the distribution at each stage are learned from some upper level contexts. In contrast to feature pyramid fusion in which the supervision signals are not directly retained for lower-level features (they impose the whole supervision onto an already fused representation), it imposes classification supervision signal for the refined output at each stage, using a weight factor.

In formulation, suppose we have *n* feature maps $\left[x_1, x_2, \dots, x_n\right]$ extracted by the backbone network, on which objects of different scales are distributed. We now specifically describe how PSR functions work for the object detection and classification on a certain feature map level x_j.

At the first step, we apply 1×1 convolution to each of backbone feature maps $x_2, x_3, \dots x_n$ yielding a new set of feature maps $h_2, h_3, \dots h_n$ with their channel numbers reduced to half. The generated $\left\{h_j\right\}_{j=2}^{n}$ are recognized as feature maps for refinement. Once obtained, they are commonly used for the classification refinement of any scale level and at any stage. Then for a certain scale level i, we apply a 3×3 convolution to the backbone feature map x_i. This yields the input of softmax for basic classification distributions of all the cell anchors on level i, denoted as z_i^0. z_i^0 is a tensor of shape $H_i \times W_i \times \left(K+1\right) A_i$, where $H_i \times W_i$ is the spatial size of x_i, $K+1$ is the number of classes (including the background class) and A_i denotes the number of anchors at each grid cell of x_i.

Suppose the progressive score refinement (PSR) is performed in *k* stages, then we pick up *k* feature maps from set $\left\{h_j\right\}_{j=2}^{n}$, which are $h_{i+1}, h_{i+2}, \dots h_{i+k}$ For the first stage of refinement, upper level contextual feature h_{i+1} is used to regress the first order residual of z_i^0 (the input of softmax classifier for basic classification prediction without refinement). We denote the first order residual of z_i^0 as $\Re_1\left(z_i\right)$, which is then added to the refined softmax input of the previous stage (here it is z_i^0, *i.e.*, stage 0 or no refinement), yielding a refined softmax input $z_i^1 = z_i^0 + \Re_1\left(z_i\right)$. This process is performed iteratively for *k* times. For the *j*-th stage of refinement $\left(1\le j\le k\right)$, the j-th order residual $\Re_j\left(z_i\right)$ is regressed from upper level context h_{i+j} by bilinear upsampling h_{i+j} to the size of x_i and applying a 3×3 convolution with dilation rate j. Through the progressive aggregation procedure, it generates the refined softmax input of each stage:

$$z_i^j = z_i^0 + \sum_{s=1}^{j} \Re_s\left(z_i\right) \quad \left(1\le j\le k\right) \tag{1}$$

After obtaining z_i^0 and its k refinements $z_i^1, z_i^2, \dots, z_i^k$, we apply softmax to each of them, yielding p_i^0 (the basic classification score) and $p_i^1, p_i^2, \dots, p_i^k$ (k classification scores which are progressively refined by the *k*-stage PSR). For each score prediction p_i^j $\left(0\le j\le k\right)$, a cross-entropy loss is computed with a weight factor λ_j $\left(0\le \lambda_j\le 1\right)$, which is the classification supervision signal imposed on stage *j* (denoted as $L_{cls,i}^{j}$):

$$L_{cls,i}^{j} = \lambda_j CE\left(p_i^j, y_i\right) \quad \left(0\le j\le k\right) \tag{2}$$

Summing up all the components across stage index j $\left(0 \le j \le k\right)$, the total classification loss for targets distributed onto x_i is:

$$L_{cls,i} = \sum_{j=0}^{k} L_{cls,i}^{j} = \sum_{j=0}^{k} \lambda_j CE\left(p_i^j, y_i\right) \qquad \textbf{(3)}$$

where λ_j $\left(0 \le \lambda_j \le k\right)$ controls the intensity of classification supervision signal at each stage and $\sum_{j=0}^{k} \lambda_j = 1$. Practically, we further simplify the loss form as follows:

$$\begin{aligned} L_{cls,i} &= \sum_{j=0}^{k} \lambda_j CE\left(p_i^j, y_i\right) \\ &= \sum_{j=0}^{k} \lambda_j sum\left(y_i \log\left(p_i^j\right)\right) \\ &= -sum\left(\sum_{j=0}^{k} y_i \log\left(p_i^j\right)^{\lambda_j}\right) \\ &= -sum\left(y_i \log \prod_{j=0}^{k} \left(p_i^j\right)^{\lambda_j}\right) \end{aligned} \qquad \textbf{(4)}$$

where $sum(\bullet)$denotes the sum of all elements of a tensor. This is to avoid some numerical issues caused by multiple $\log(\bullet)$s.

To get the final prediction of classification on scale level i, we first weight-sum the inputs of softmax across all the stages (including stage 0 which corresponds to the basic prediction) using the same group of weight factors $\left\{\lambda_j\right\}_{j=0}^{k}$ as we control the intensities of supervision signals, then apply another softmax to obtain the final classification score at level i. In formulation, the final prediction at level i is:

$$p_i = soft\max\left(\sum_{j=0}^{k} \lambda_j z_i^j\right) \qquad \textbf{(5)}$$

By using PSR, an explicit modeling of classification is established, making the context aggregation procedure manually controlled by the retaining of supervision signals along stages. This helps the detector find a balance among optimization on different semantic levels and allows more sufficient optimization on lower-level features, thus avoids the model from over-depending on higher level information which is far more apt to make classifier forcibly fit labels. The iterative adding up of prediction residuals can adaptively adjust the classification distributions, which reduces the ambiguity for predictions on a certain scale level. Moreover, the weighted integration of multiple stages of refinement to obtain the ultimate classification score develops an effective and efficient way to stabilize the final prediction and alleviate the side-effect of mis-refinements.

3.2 Decoupling Localization from Higher Level Features

From lower-level features to higher level features in the backbone network, subsampling like pooling and large stride convolution is performed several times. This makes many spatial details dropped progressively. Thus, there is less potential for higher level features to boost localization of objects on smaller scale levels where more position sensitive information is needed.

In our work, only the original backbone feature map at each scale level is used to localize objects. Therefore, higher level features are only used for classification refinements and are decoupled from localization.

According to the PSR modeling, it is also easy to fora similar Progressive Localization Refinement (PLR) procedure using higher level features. We do this as part of our experiments, which proves the hypothesis. Please see Section 4 for detail.

3.3 Overall Architecture

3.3.1 Network Architecture

We adopt SSD as the baseline framework for PRN to validate the effectiveness of our approach. The backbone network is VGGNet which is exactly the same as the original SSD.

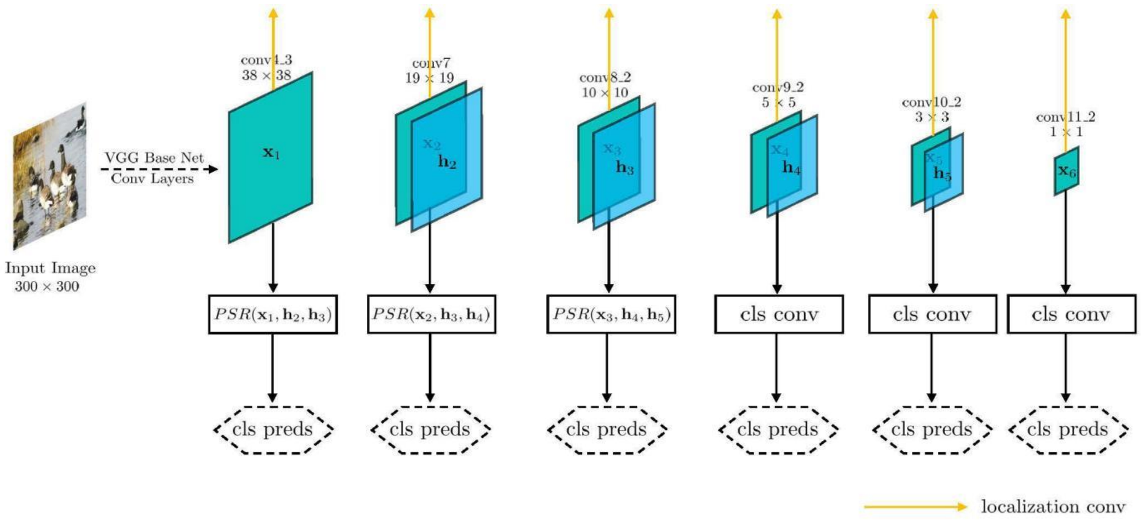

Figure 2. Overall architecture for the proposed method (PRN) on SSD framework

For input size 300×300, as shown in Figure 2, the SSD baseline extracts feature maps with 6 scale levels to detect objects of different scales. The feature maps extracted are conv4_3, conv7, conv8_2, conv9_2, conv10_2, and conv11_2. The corresponding spatial scales are 38, 19, 10, 5, 3, 1 and the channel numbers are 512, 1024, 512, 256, 256 and 256. We further denote the 6 extracted feature maps as $x_1, x_2, \dots, x_6$, correspondingly. Since small targets are commonly distributed onto lower levels, we conduct progressive score refinement for scale level 1, 2 and 3. The number of stages for refinement k is set to 2 and the weight factors to control the intensity of classification supervision signal at each stage are set as $\lambda_0 = \lambda_1 = 0.25$ and $\lambda_2 = 0.5$. Since the highest scale-level we do PSR on is level 3 and we do 2-stage PSR, the highest level of h_i needed is h_5 .Therefore, we perform 1×1 convolutions from x_2 to x_5 with their output channels halved, yielding h_2 to h_5 as feature maps for refinement. The h_6 is thus ignored for levels 4, 5 and 6. The classification modeling is kept the same as the original SSD. Localization is only conducted on backbone feature maps $x_1, x_2, \dots, x_6$.

For input size 512×512 where 7 levels are extracted by the baseline SSD, we apply 2-stage PSR for scale levels 1, 2, 3 and 4. Other settings are the same as those for input size. The obtained models of our method under the two input sizes are denoted as PRN300 and PRN512 respectively.

3.3.2 Training objective

For PRN, the total training objective has two components:

$$L = L_{cls} + \alpha L_{loc} \tag{6}$$

For the classification loss L_{cls}:

$$L_{cls} = \frac{1}{|S_+|}\left(\sum_{i=1}^{m} L_{cls,i} + \sum_{i=m+1}^{n} L_{cls,i}\right) \tag{7}$$

The first part corresponds to level 1 to m on which PSR is performed. For these levels, classification loss is defined as in Section 3.1. The second part corresponds to other levels where PSR is not performed. For these levels, classification loss is the same as baseline SSD. Under the settings of PRN300, $m=3$, $n=6$. And for PRN512, $m=4$, $n=7$.

The localization loss for bounding box regression L_{loc} is by the same definition as in the original SSD. Factor α is used to balance localization and classification, which is empirically set to 2.5 in our approach.

4 Experiments

For validation and further analysis, we conduct experiments on the PASCAL VOC 2007 dataset [8] and the MS COCO dataset [9], which have 20 and 80 categories respectively.

4.1 Implementation Details

We implement PRN using the deep learning framework PyTorch [34] on a single desktop computer with one Intel i5-6500 CPU, four Nvidia GTX 1080Ti GPUs and 64GB memory. Backbone VGGNet pre-trained on ImageNet [35] is used to initialize the models. For all the newly added layers, the uniform version of the MSRA method [36] is adopted for the random initialization of their weights. We apply the linear warm-up strategy at the beginning of training. Specifically, we set the learning rate to 1/3 of the base value at the first step, linearly increase it to the base learning rate in the following 300 steps and then retain it. Sampling strategy, hard example mining rules and data augmentation are kept the same as baseline SSD.

For PRN300, we do progressive score refinement (PSR) on the first three scale levels. And for PRN512, PSR is applied on the first four scale levels. For both PRN300 and PRN512, the number of stages k is 2 and the weight factors to control the classification supervision signals are $\lambda_0 = \lambda_1 = 0.25$, $\lambda_2 = 0.5$.

4.2 Results on PASCAL VOC 2007

For the PASCAL VOC 2007 dataset, we train our models using the combination of VOC 2007 trainval set and VOC 2012 trainval set and evaluate the models on VOC 2007 test set. A batch size of 32 is used for training and the optimization algorithm adopted is SGD, with momentum and weight decay set to 0.9 and 0.0001 respectively. The models are trained for 250 epochs (173k iterations) in total. We use a learning rate of 10^{-3} with the warm-up strategy described in Section 4.1 and then decay its value by 0.1 at iteration 100k and 140k. At test time, NMS (Non-Maximum Suppression) with 0.45 IoU threshold is applied to remove duplicate detection. The models are trained with 4 Nvidia GTX 1080Ti GPUs and tested on a single GPU. Both PRN300 and PRN512 are trained and tested.

The overall results on PASCAL VOC 2007 are shown in Table 1. For input size 300×300, PRN300 reaches 79.5% mAP (mean average precision), surpassing many other SSD like detectors, including DSSD321, STDN300 and FSSD300. It reaches comparable accuracy as Deep Feature Pyramid Reconfiguration (FP Reconfig.), which is based on feature pyramid fusion using a carefully handcrafted fusion transformation. Reaching high accuracy, PRN300 runs at a high speed (75.7 fps), which is only of a small speed drop compared to the baseline. For input size 512×512, PRN512 reaches 81.3% mAP, which also surpasses STDN513 and FSSD512. Despite the huge inferiority on backbone network, its accuracy is just slightly lower (by 0.2% mAP) than DSSD513 using the much powerful ResNet-101 as backbone, but it runs at a much higher speed. It is worth noticing that under the input size 512×512, our approach is slightly higher than Deep Feature Pyramid Reconfiguration (81.3% mAP vs. 81.1% mAP), which is exactly the opposite of input size 300×300.

Table 1. Overall results on the PASCAL VOC 2007 dataset
(All the entries listed use the combination of VOC 2007 trainval and VOC 2012 trainval as training data and use VOC 2007 test as test data.)

Method	Backbone	Input size	GPU	Speed (fps)	mAP (%)
Faster R-CNN [23]	ResNet-101 [38]	600 × 1000	K40	2.4	76.4
R-FCN [26]	ResNet-50	600 × 1000	-	-	77.0
SSD300 [7]	VGGNet	300 × 300	1080Ti	**83.3**	77.2
SSD512 [7]	VGGNet	512 × 512	1080Ti	**39.2**	79.8
YOLOv2 [13]	DarkNet-19	544 × 544	Titan X	40	78.6
DSSD321 [12]	ResNet-101	300 × 300	Titan X	9.5	78.6
DSSD513 [12]	ResNet-101	513 × 513	Titan X	5.5	**81.5**
STDN300 [37]	DenseNet-169 [39]	300 × 300	Titan Xp	41.5	78.1
STDN513 [37]	DenseNet-169	513 × 513	Titan Xp	28.6	80.9
FSSD300 [3]	VGGNet	300 × 300	1080Ti	65.8	78.8
FSSD512 [3]	VGGNet	512 × 512	1080Ti	35.7	80.9
FP Reconfig.300 [4]	VGGNet	300 × 300	Titan X	39.5	**79.6**
FP Reconfig.512 [4]	VGGNet	512 × 512	Titan X	-	81.1
PRN300	VGGNet	300 × 300	1080Ti	75.7	79.5
PRN512	VGGNet	512 × 512	1080Ti	38.3	81.3

4.3 Ablation Study

4.3.1 The Impact of Scale Levels

To demonstrate the effectiveness of the key component (PSR) of the proposed method, we increase the number of scale levels to do PSR from the first level to the fourth level for PRN300. Results are evaluated on the PASCAL VOC 2007 dataset. As shown in Table 2, after adding PSR on level 1, the mAP increases from 77.2% to 78.5%, which is the most significant. This is because objects of the smallest scales are distributed on this level and these objects suffer from high classification ambiguity most. After adding PSR on scale level 3, it yields the highest accuracy (79.5% mAP). Adding PSR on level 4 makes the mAP fall back to 78.8%. The most possible reason is that extra high level feature maps have less volume of information to support refinements since their spatial sizes are too small. Through the series of experiments to this step, the effectiveness of the proposed PSR approach is demonstrated. Finally, we remove the weight-sum integration for obtaining the final class prediction of PSR and replace it with using the prediction of the last stage instead. This drops the mAP by 0.4% from 79.5% to 79.1%, which reveals that the proposed weight-sum integration method for PSR to obtain the final class prediction can effectively reduce the side-effect of mis-refinements.

Table 2. Ablation study of PRN300 on the PASCAL VOC 2007 dataset
(All the models are trained with the combination of VOC 2007 trainval and VOC 2012 trainval, using the VOC 2007 test set for evaluation.)

Component	PRN300					
PSR on level 1		√	√	√	√	√
PSR on level 2			√	√	√	√
PSR on level 3				√	√	√
PSR on level 4					√	
Weight-sum integration		√	√	√	√	
mAP (%)	77.2	78.5	78.8	**79.5**	78.8	79.1

4.3.2 The Contribution of Higher-level Features

For Similar to PSR, it's also easy to do progressive refinement for localization using higher-level features. To ensure our hypothesis that higher levels contribute less for localization, we implement progressive localization refinement (PLR) on scale level 1, 2 and 3. For PLR, the definition and hyper parameters are all the same as PSR in this paper, with only the subject of refinement changed from the softmax input of classifier to localization offsets. To verify whether the localization refinement works, we take the localization output at each stage for evaluation respectively.

As shown in Table 3, adding PLR causes a significant performance drop (by 0.9% mAP). And as we take stage 0, 1 and 2 for localization output respectively, the mAP has little change or even slightly decreases, which supports that higher-level features contribute less for localization.

Table 3. Experimental results of the effect of localization refinement
(All the entries are trained on VOC 2007 trainval and VOC 2012 trainval, using VOC 2007 test for evaluation. PLR #i denotes the i-th stage of localization refinement is taken for evaluation. And PLR denotes the localization output is obtained by weight-sum integration across stages, which is similar to PSR.)

Method	PRN300	w/PLR	w/PLR# 0	w/PLR# 1	w/PLR# 2
mAP (%)	79.5	78.6	78.6	78.5	78.5

4.4 Quantitative Analysis on Classification Ambiguity

In this part, we show the classification ambiguity analysis quantitatively. To quantify the ambiguity of classification, we use the self-entropy or the self-information of the predicted probability distribution. Specifically, for a certain cell anchor, if the predicted classification probability distribution for this anchor is $(p_1, p_2, \ldots, p_{K+1})$, ($K+1$ is the number of classes, including background), then the classification ambiguity of this anchor (or sample) is defined as follows:

$$I(p) = -\sum_{i=1}^{K+1} p_i \log(p_i) \qquad \textbf{(8)}$$

The probable maximum value of $I(p)$ is $\log(K+1)$ when $p_1 = p_2 = \ldots = p_{(K+1)} = \dfrac{1}{K+1}$. In practice, we use this value ($\log(K+1)$) to normalize $I(p)$ as the final quantitative indicator so that it falls in range [0, 1].

For the analysis, we gather statistics of the normalized self-entropies for positive anchors on scale level 1, 2 and 3 respectively. On each occasion, we plot the cumulative distribution function of the normalized self-entropy over these anchors for the baseline and PRN300. Since lower value of self-entropy indicates less ambiguity on classification, the curve at top-left is better under this quantitative index. Results are shown in Figure 3. It can be seen from the figure that classification ambiguity is significantly reduced using PSR, especially for small scale levels.

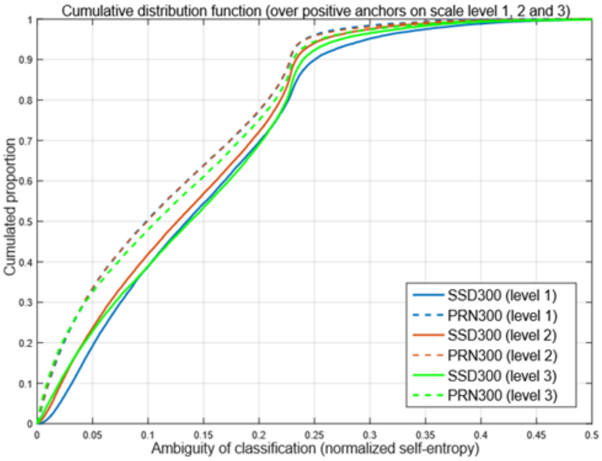

Figure 3. Cumulative distribution functions of normalize self-entropy
(The curve at top-left corresponds to a better model. Statistics are over positive anchors on scale level 1, 2 and 3 respectively. And the scope of statistics covers all the test images in the VOC 2007 test set.)

4.5 Comparison with Feature Pyramid Fusion

To further verify the effectiveness of PRN against feature pyramid fusion, we implement other two feature pyramid fusion methods FPN [1] and TDM [2] onto the baseline SSD300. For FPN, the fusion transformation defined in [1] is applied on the 6 backbone feature maps extracted by SSD300. The number of channels for lateral layers is set to 256 as is most commonly adopted. For TDM, modules as described in [2] are added onto the same 6 backbone maps with the channel numbers of lateral layers, top-down layers and out layers set to 128, 128 andf 256, respectively. Experimental results are evaluated on the PASCAL VOC 2007 dataset, under 300×300 input size. As shown in Table 4, our method (PRN300) surpasses the two feature pyramid fusion methods by large margins, which further demonstrates that the proposed method is more effective than feature pyramid fusion.

Table 4. Experimental results of feature pyramid fusion vs. PRN
(All the entries are trained on VOC 2007 trainval and VOC 2012 trainval, using VOC 2007 test for evaluation.)

method	SSD300	SSD300w /FPN	SSD300 w/TDM	PRN300
mAP (%)	77.2	78.3	78.2	**79.5**

4.6 Results on MS COCO

For the MS COCO dataset, the model is trained on the COCO2017 train set and tested on the COCO test-dev set. We train our model for 110 epochs (around 403k iterations) in total with a batch size of 32. The SGD optimizer is adopted with the momentum and weight decay set to 0.9 and 0.0001. We use a learning rate of 10^{-3} with the warm-up strategy described in Section 4.1 and then decay its value by 0.1 at iteration 295k and

370k. An NMS with 0.5 IoU threshold is applied at test time.

We evaluate PRN512 on the COCO test-dev set. Results are shown in Table 5. On COCO test-dev, PRN512 reaches 31.7 AP, which is much higher than the SSD512 baseline. It also reaches comparable result as Deep Feature Pyramid Reconfiguration (FP Reconfig.) and surpasses the SSD513 with a much more powerful backbone (ResNet-101). Despite that the AP of PRN512 is slightly lower than STDN513 and FSSD512, for the performance on small objects (APS), our result reaches the best (15.6) among the listed methods, suggesting a superiority for scale robustness on small objects, which is one of our main objectives.

Table 5. Experimental results on the COCO test-dev set (All the entries are trained on the COCO2017 train (trainval35k) set.)

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLOv2 [13]	DarkNet-19	21.6	44.0	19.2	5.0	22.4	35.5
SSD512 [7]	VGGNet	28.8	48.5	30.3	10.9	31.8	43.5
SSD513 [12]	ResNet-101	31.2	50.4	33.3	10.2	34.5	**49.8**
STDN513 [37]	DenseNet-169	**31.8**	51.0	33.6	14.4	**36.1**	43.4
FSSD512 [3]	VGGNet	**31.8**	**52.8**	33.5	14.2	35.1	45.0
FP Reconfig512 [4]	VGGNet	31.5	50.9	33.2	-	-	-
PRN512	VGGNet	31.7	51.0	**33.9**	**15.6**	33.5	44.0

4.7 Discussion on the Limitation of This Work

In this paper, we have proved that the progressive refinement strategy is effective for small object detection by collaboration with standard CNN object detection framework. However, there are several issues to explore considering the limitation of this work:

1）Given a fixed CNN framework, it still requires additional computation budget for feature extraction on small objects. To better facilitate real applications by gaining more improved efficiency, a scale-adaptive convolution scheme may be well complementary to our PSN framework.

2）Its general applicability in collaborating with new object detection pipeline, e.g., the transformer-based object detection framework, still needs to be investigated.

3）Beyond the standard visual object detection datasets, some real-world tiny object detection scenarios are also worth taken into consideration, despite that it is beyond the scope of this paper. For example, one can consider object detection task on the video taken by drones, where the object sizes depend on the height of the drones. Under this setting, the object scales may be much smaller than natural photographs, which brings more challenges to existing achievements.

5 Conclusion

In this paper, we propose the Progressive Refinement Network (PRN), a novel paradigm to deal with the ever-lasting issue of scale robustness in object detection. The main idea of PRN is to reduce ambiguity for small object classification by progressively adding residuals to the basic prediction, yielding a sequence of gradually refined classification pre dictions. To decrease the side-effect of mis-refinement, we also develop an integration mechanism to synchronously control the intensity of supervision signal at each stage and integrate the refined predictions at all the stages, which helps to stabilize the final classification distribution. Experimental results show that with progressive refinement conducted on several scale levels corresponding to small objects, reliable performance gain as well as a superiority for detecting small targets can be obtained, which demonstrates the effectiveness of PRN for improving scale robustness.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (61806073), and the Key Scientific and Technological Project of Henan Province (212102210160, 222102210256), and the Key Scientific Research Project in Colleges and Universities of Henan Province (21A520051).

References

[1] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature Pyramid Networks for Object Detection, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 2117-2125.

[2] A. Shrivastava, R. Sukthankar, J. Malik, A. Gupta, Beyond Skip Connections: Top-Down Modulation for Object Detection, *arXiv preprint*, arXiv:1612.06851, September, 2017.

[3] Z. Li, F. Zhou, FSSD: Feature Fusion Single Shot Multibox Detector, *arXiv preprint*, arXiv:1712.00960, May, 2018.

[4] T. Kong, F. Sun, W. Huang, H. Liu, Deep Feature Pyramid Reconfiguration for Object Detection, *2018 European Conference on Computer Vision (ECCV)*, Munich, Germany, 2018, pp. 169-185.

[5] F. Yang, W. Choi, Y. Lin, Exploit All the Layers: Fast and Accurate CNN Object Detector with Scale Dependent Pooling and Cascaded Rejection Classifiers, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 2129-2137.

[6] Z. Cai, N. Vasconcelos, Cascade R-CNN: Delving Into High Quality Object Detection, *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 6154-6162.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. -Y. Fu, A. -C. Berg, SSD: Single Shot MultiBox Detector, *2016 European Conference on Computer Vision (ECCV)*, Amsterdam, The Netherlands, 2016, pp. 21-37.

[8] M. Everingham, L. -V Gool, C. -K. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes (VOC) Challenge, *International journal of computer vision*, Vol. 88, No. 2, pp. 303-338, June, 2010.

[9] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, C. -L. Zitnick, Microsoft COCO: Common Objects in Context, *2014 European Conference on Computer Vision (ECCV)*, Zurich, Switzerland, 2014, pp. 740-755.

[10] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, *arXiv preprint*, arXiv:1409.1556, April, 2014.

[11] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779-788.

[12] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, A. C. Berg, DSSD: Deconvolutional Single Shot Detector, *arXiv preprint*, arXiv:1701.06659, January, 2017.

[13] J. Redmon, A. Farhadi, YOLO9000: Better, Faster, Stronger, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 7263-7271.

[14] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, Y. Chen, RON: Reverse Connection with Objectness Prior Networks for Object Detection, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 5936-5944.

[15] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal Loss for Dense Object Detection, *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017, pp. 2980-2988.

[16] S. Zhang, L. Wen, X. Bian, Z. Lei, S. Z. Li, Single-Shot Refinement Neural Network for Object Detection, *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 4203-4212.

[17] B. Yao, H. Zhou, J. Yin, G. Li, C. Lv, Small Sample Image Recognition Based on CNN and RBFNN, *Journal of Internet Technology*, Vol. 21, No. 3, pp. 881-889, May, 2020.

[18] H. Law, J. Deng, CornerNet: Detecting Objects as Paired Keypoints, *2018 European Conference on Computer Vision (ECCV)*, Munich, Germany, 2018, pp. 734-750.

[19] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, Q. Tian, CenterNet: Keypoint Triplets for Object Detection, *2019 IEEE International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 2019, pp. 6569-6578.

[20] X. Zhou, D. Wang, P. Krähenbühl, Objects as Points, *arXiv preprint*, arXiv:1904.07850, April, 2019.

[21] X. Zhang, F. Wan, C. Liu, R. Ji, Q. Ye, FreeAnchor: Learning to Match Anchors for Visual Object Detection, *2019 Conference on Neural Information Processing System (NeurIPS)*, Vancouver, Canada, 2019, pp. 147-155.

[22] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, 2014, pp. 580-587.

[23] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *2015 Conference on Neural Information Processing System (NeurIPS)*, Montreal, Canada, 2015. pp. 91-99.

[24] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, *2014 European Conference on Computer Vision (ECCV)*, Zurich, Switzerland, 2014, pp. 346-361.

[25] S. Bell, C.-L. Zitnick, K. Bala, R. Girshick, Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 2874-2883.

[26] J. Dai, Y. Li, K. He, J. Sun, R-FCN: Object Detection via Region-based Fully Convolutional Networks, *2016 Conference on Neural Information Processing System (NeurIPS)*, Barcelona, Spain, 2016, pp. 379-387.

[27] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask RCNN, *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017, pp. 2961-2969.

[28] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, J. Sun, Light-Head R-CNN: In Defense of Two-Stage Object Detector, *arXiv preprint*, arXiv:1711.07264, November, 2017.

[29] H. Yuan, H. Zhou, Z. Cai, S. Zhang, R. Wu, Dynamic Pyramid Attention Networks for multi-orientation object detection, *Journal of Internet Technology*, Vol. 3, No. 1, pp. 79-90, January, 2022.

[30] M. Eisenbach, D. Seichter, T. Wengefeld, H.-M. Gross, Cooperative multi-scale Convolutional Neural Networks for person detection, *2016 International Joint Conference on Neural Networks (IJCNN)*, Vancouver, BC, Canada, 2016, pp. 267-276.

[31] R. Girshick, Fast R-CNN, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Santiago, Chile, 2015, pp. 1440-1448.

[32] B. Singh, L.-S. Davis, An Analysis of Scale Invariance in Object Detection-SNIP, *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 3578-3587.

[33] S.-W. Kim, H.-K. Kook, J.-Y. Sun, M.-C. Kang, S.-J. Ko, Parallel Feature Pyramid Network for Object Detection, *2018 European Conference on Computer Vision (ECCV)*, Munich, Germany, 2018, pp. 234-250.

[34] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z.-D. Vito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in PyTorch, *2017 Conference on Neural Information Processing System (NeurIPS) Workshop*, Long Beach, CA, US, 2017, pp. 1-4.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, F.-F. Li, Imagenet: A large-scale hierarchical image database, *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, FL, USA, 2009, pp. 248-255.

[36] K. He, X. Zhang, S. Ren, J. Sun, Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Santiago, Chile, 2015, pp. 1026-1034.

[37] P. Zhou, B. Ni, C. Geng, J. Hu, Y. Xu, Scale-Transferrable Object Detection, *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 528-537.

[38] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770-778.

[39] G. Huang, Z. Liu, L.-V Maten, K.-Q. Weinberger, Densely Connected Convolutional Networks, *2017 IEEE Conference on Computer Vision and Pattern*

Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 4700-4708.

Biographies

Tao Li received the Ph.D. degree from University of Electronic Science and technology, Chengdu, China, in 2016. He is currently an associate professor in the Open University of Henan, China. His current research interests include machine learning and object detection, object classification in computer vision.

Yu Wang achieved his Master of Engineering degree from Henan University of Technology in 2018. He is currently a lecturer in the Open University of Henan, China. His current research fields are genetic algorithm, evolutionary computation and optimization theory.

Zheng Zhang received the M.E. degree in from Zhongyuan University of Technology, Zhengzhou, China, in 2017. She is currently a lecturer in the Open University of Henan China. Her current research interests include machine learning and object detection, object classification in computer vision.

Xuezhuan Zhao received the Ph.D. degree from the Chinese academy of sciences, China, in 2016. He is currently an associate professor in the Zhengzhou University of Aeronautics, China. His current research interests include object detection, saliency detection, in computer vision.

Lishen Pei received the Ph.D. degree from University of Electronic Science and technology, Chengdu, China, in 2016. She is currently a lecturer in the Henan University of Economics and Law, China. Her current research interests include object detection, action recognition in computer vision.