arXiv:2003.07021v1 [cs.CV] 16 Mar 2020

Extended Feature Pyramid Network for Small Object Detection

Chunfang Deng, Mengmeng Wang, Liang Liu, and Yong Liu†

Zhejiang University
{dengcf, mengmengwang, leonliuz}@zju.edu.cn, yongliu@iipc.zju.edu.cn

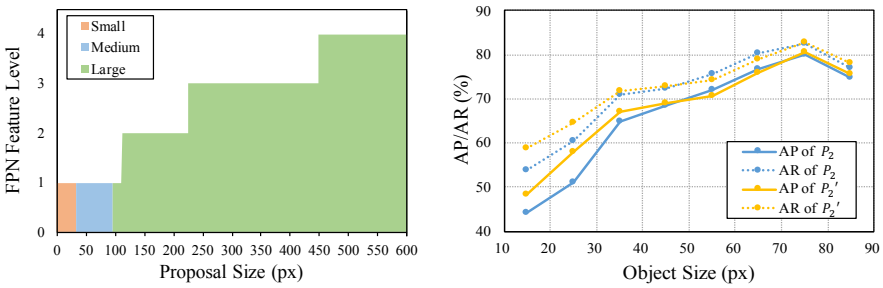
Abstract. Small object detection remains an unsolved challenge because it is hard to extract information of small objects with only a few pixels. While scale-level corresponding detection in feature pyramid network alleviates this problem, we find feature coupling of various scales still impairs the performance of small objects. In this paper, we propose extended feature pyramid network (EFPN) with an extra high-resolution pyramid level specialized for small object detection. Specifically, we design a novel module, named feature texture transfer (FTT), which is used to super-resolve features and extract credible regional details simultaneously. Moreover, we design a foreground-background-balanced loss function to alleviate area imbalance of foreground and background. In our experiments, the proposed EFPN is efficient on both computation and memory, and yields state-of-the-art results on small traffic-sign dataset Tsinghua-Tencent 100K and small category of general object detection dataset MS COCO.

Keywords: Small Object Detection, Feature Pyramid Network, Feature Super-Resolution

1 Introduction

Object detection is a fundamental task of many advanced computer vision problems such as segmentation, image caption and video understanding. Over the past few years, rapid development of deep learning has boosted the popularity of CNN-based detectors, which mainly include two-stage pipelines [8,7,28,5] and one-stage pipelines [24,27,20]. Although these general object detectors have improved accuracy and efficiency substantially, they still perform poorly when detecting small objects with a few pixels. Since CNN uses pooling layers repeatedly to extract advanced semantics, the pixels of small objects can be filtered out during the downsampling process.

Utilization of low-level features is one way to pick up information about small objects. Feature pyramid network (FPN) [19] is the first method to enhance features by fusing features from different levels and constructing feature pyramids, where upper feature maps are responsible for larger object detection, and lower feature maps are responsible for smaller object detection. Despite FPN improves multi-scale detection performance, the heuristic mapping mechanism



(a) Mapping between pyramid level and proposal size in vanilla FPN detectors.

(b) Detection performance of original P_2 and our P_2' on Tsinghua-Tencent 100K.

Fig. 1. The drawback of small object detection in vanilla FPN detectors. (a)*Feature Coupling:* Both small and medium objects are detected on the lowest level (P_2) of FPN. (b)*Poor Performance of Small Objects on* P_2: The detection performance of P_2 varies with scale, and the average precision (AP) and average recall (AR) decline sharply when instances turn small. The extended pyramid level P_2' in our EFPN mitigates this performance drop

between pyramid level and proposal size in FPN detectors may confuse small object detection. As shown in Fig. 1(a), small-sized objects must share the same feature map with medium-sized objects and some large-sized objects, while easy cases like large-sized objects can pick features from a suitable level. Besides, as shown in Fig. 1(b), the detection accuracy and recall of the FPN bottom layer fall dramatically as the object scale decreases. Fig.1 suggests that, feature coupling across scales in vanilla FPN detectors still degenerates the ability of small object detection.

Intuitively, another way of compensating for the information loss of small objects is to increase the feature resolution. Thus some super-resolution (SR) methods are introduced to object detection. Early practices [11,3] directly super-resolve the input image, but the computational cost of feature extraction in the following network would be expensive. Li et al. [14] introduce GAN [10] to lift features of small objects to higher resolution. Noh et al. [25] use high-resolution target features to supervise SR of the whole feature map containing context information. These feature SR methods avoid adding to the burden of the CNN backbone, but they imagine the absent details only on the basis of the low-resolution feature map, and neglect credible details encoded in other features of backbones. Hence, they are inclined to fabricate fake textures and artifacts on CNN features, causing false positives.

In this paper, we propose extended feature pyramid network (EFPN), which employs large-scale SR features with abundant regional details to decouple small and medium object detection. EFPN extends the original FPN with a high-resolution level specialized for small-sized object detection. To avoid expensive computation that would be caused by direct high-resolution image input, the

extended high-resolution feature maps of our method is generated by feature SR embedded FPN-like framework. After construction of the vanilla feature pyramid, the proposed feature texture transfer (FTT) module firstly combines deep semantics from low-resolution features and shallow regional textures from high-resolution feature reference. Then, the subsequent FPN-like lateral connection will further enrich the regional characteristics by tailor-made intermediate CNN feature maps. One advantage of EFPN is that the generation of the high-resolution feature maps depends on original real features produced by CNN and FPN, rather than on unreliable imagination in other similar methods. As shown in Fig. 1(b), the extended pyramid level with credible details in EFPN improves detection performance on small objects significantly.

Moreover, we introduce features which are generated by large-scale input images as supervision to optimize EFPN, and design a foreground-background-balanced loss function. We argue that general reconstruction loss will lead to insufficient learning of positive pixels, as small instances merely cover fractional area on the whole feature map. In light of the importance of foreground-background balance [20], we add loss of object areas to global loss function, drawing attention to the feature quality of positive pixels.

We evaluate our method on challenging small traffic-sign dataset Tsinghua-Tencent 100K and general object detection dataset MS COCO. The results demonstrates that the proposed EFPN outperforms other state-of-the-art methods on both datasets. Besides, compared with multi-scale test, single-scale EFPN achieves similar performance but with fewer computing resources.

For clarity, the main contributions of our work can be summarized as:

(1) We propose extended feature pyramid network (EFPN) which improves the performance of small object detection.
(2) We design a pivotal feature reference-based SR module named feature texture transfer (FTT), to endow the extended feature pyramid with credible details for more accurate small object detection.
(3) We introduce a foreground-background-balanced loss function to draw attention on positive pixels, alleviating area imbalance of foreground and background.
(4) Our efficient approach significantly improves the performance of detectors, and becomes state-of-the-art on Tsinghua-Tencent 100K and small category of MS COCO.

2 Related Work

2.1 Deep Object Detectors

Deep learning based detectors have ruled general object detection due to their high performance. The successful two-stage methods [8,7,28,5] firstly generate Regions of Interest (RoIs), and then refine RoIs with a classifier and a regressor. One-stage detectors [24,27,20], another kind of prevalent detectors, directly conduct classification and localization on CNN feature maps with the help of

pre-defined anchor boxes. Recently, anchor-free frameworks [13,38,31,39] also become increasingly popular. Despite of the development of deep object detectors, small object detection remains an unsolved challenge. Dilated convolution [34] is introduced in [23,17,16] to augment receptive fields for multi-scale detection. However, general detectors tend to focus more on improving the performance of easier large instances, since the metric of general object detection is average precision of all scales. Detectors specialized for small objects still need more exploration.

2.2 Cross-Scale Features

Utilizing cross-scale features is an effective way to alleviate the problem arising from object scale variation. Building image pyramids is a traditional approach to generating cross-scale features. Use of features from different layers of network is another kind of cross-scale practice. SSD [24] and MS-CNN [4] detect objects of different scales on different layers of CNN backbone. FPN [19] constructs feature pyramids by merging features from lower layers and higher layers via a top-down pathway. Following FPN, FPN variants explore more information pathways in feature pyramids. PANet [22] adds an extra down-top pathway to pass shallow localization information up. G-FRNet [1] introduces gate unit on the pathway, which passes crucial information and block ambiguous information. NAS-FPN [6] delves into optimal pathway configuration using AutoML. Though these FPN variants improve the performance of multi-scale object detection, they continue to use the same number of layers as original FPN. But these layers are not suitable for small object detection, which leads to still poor performance of small objects.

2.3 Super-Resolution in Object Detection

Some studies introduce SR to object detection, since small object detection always benefits from large scales. Image-level SR is adopted in some specific situations where extremely small objects exist, such as satellite images [15] and images with crowded tiny faces [2]. But large-scale images are burdensome for subsequent networks. Instead of super-resolving the whole image, SOD-MTGAN [3] only super-resolves the area of RoIs, but large quantities of RoIs still need considerable computation. The other way of SR is to directly super-resolve features. Li et al. [14] use Perceptual GAN to enhance features of small objects with the characteristics of large objects. STDN [37] employs sub-pixel convolution on top layers of DenseNet [12] to detect small objects and meanwhile reduce network parameters. Noh et al. [25] super-resolve the whole feature map and introduce supervision signal to training process. Nevertheless, above-mentioned feature SR methods are all based on restricted information from a single feature map. Recent reference-based SR methods [35,36] have capacity of enhancing SR images with textures or contents from reference images. Enlightened by reference-based SR, we design a novel module to super-resolves features under the reference of

shallow features with credible details, thus generating features more suitable for small object detection.

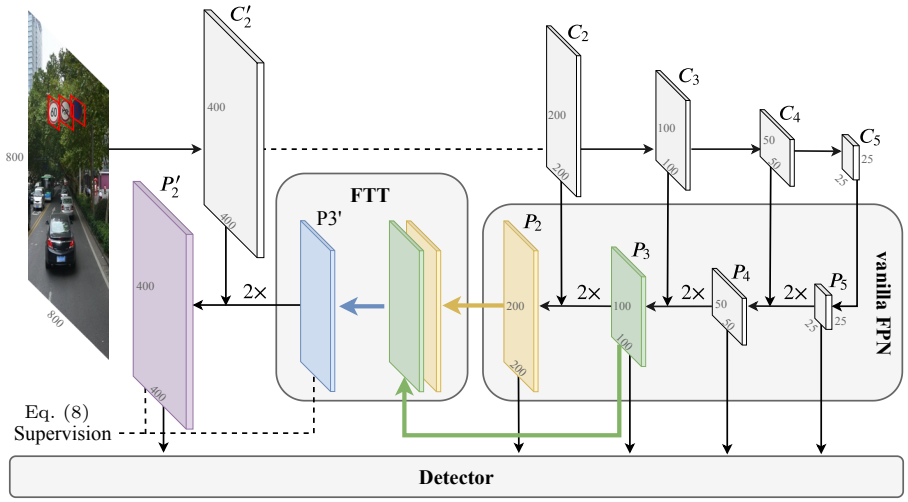


Fig. 2. The framework of extended feature pyramid network (EFPN). Here C_i denotes the feature map from stage i of CNN backbone, and P_i denotes the corresponding pyramid level on EFPN. Top 4 layers of EFPN are vanilla FPN layers. Feature texture transfer (FTT) module integrates semantic contents from P_3 and regional textures from P_2. And then, an FPN-like top-down pathway passes FTT module output down to form the final extended pyramid level P_2'. The extended feature pyramid $(P_2', P_2, P_3, P_4, P_5)$ will be fed to the following detector for further object localization and classification

3 Our Approach

In this section, we will introduce the proposed extended feature pyramid network (EFPN) in detail. First, we construct an extended feature pyramid, which is specialized for small objects with a high-resolution feature map at the bottom. Specifically, we design a novel module named feature texture transfer(FTT), to generate intermediate features for the extended feature pyramid. Moreover, we employ a new foreground-background-balanced loss function to further enforce learning of positive pixels. The pipeline of EFPN network and FTT module is explained in Sec. 3.1 and Sec. 3.2, and Sec. 3.3 elaborates our loss function design.

3.1 Extended Feature Pyramid Network

Vanilla FPN constructs a 4-layer feature pyramid by upsampling high-level CNN feature maps and fusing them with lower features by lateral connections. Al-

Table 1. Generation of C_2' in ResNet/ResNeXt backbones. A new branch without max-pooling in stage2 is added to generate C_2', simulating the semantics and resolution of C_2 from 2× input image. The branches of C_2 and C_2' share the same weights. In EFPN, C_2 and C_2' are generated simultaneously from 1× input

Layer Name	Layer Components	
Input	800×800(1×)	800×800(1×)
Stage1	7×7, 64, stride 2	7×7, 64, stride 2
Stage2	3×3 max pool, stride 2 residual blocks ×3	residual blocks ×3
Output	C_2:(200×200)	C_2':(400×400)

though features on different pyramid levels are responsible for objects of different sizes, small object detection and medium object detection are still coupled on the same bottom layer P_2 of FPN, as shown in Fig. 1. To relieve this issue, we propose EFPN to extend the vanilla feature pyramid with a new level, which accounts for small object detection with more regional details.

We implement the extended feature pyramid by an FPN-like framework embedded with a feature SR module. This pipeline directly generates high-resolution features from low-resolution images to support small object detection, while stays in low computational cost. The overview of EFPN is shown in Fig. 2.

Top 4 pyramid layers are constructed by top-down pathways for medium and large object detection. The bottom extension in EFPN, which contains an FTT module, a top-down pathway and a purple pyramid layer in Fig. 2, aims to capture regional details for small objects. More specifically, in the extension, the 3rd and 4th pyramid layers of EFPN which are denoted by green and yellow layers respectively in Fig. 2, are mixed up in the feature SR module FTT to produce the intermediate feature P_3' with selected regional information, which is denoted by a blue diamond in Fig. 2. And then, the top-down pathway merges P_3' with a tailor-made high-resolution CNN feature map $C2'$, producing the final extended pyramid layer P_2'. We remove a max-pooling layer in ResNet/ResNeXt stage2, and get C_2' as the output of stage2, as shown in in Table 1. C_2' shares the same representation level with original C_2 but contains more regional details due to its higher resolution. And the smaller receptive field in C_2' also helps better locate small objects. Mathematically, operations of the extension in the proposed EFPN can be described as

$$P_2' = P_3' \uparrow_{2\times} + C_2' \tag{1}$$

where $\uparrow_{2\times}$ denotes double upscaling by nearest-neighbor interpolation.

In EFPN detectors, the mapping between proposal size and pyramid level still follows the fashion in [19]:

$$l = \lfloor l_0 + log_2(\sqrt{wh}/224) \rfloor \tag{2}$$

Here l represents pyramid level, w and h are the width and height of a box proposal, 224 is the canonical ImageNet pre-training size, and l_0 is the target

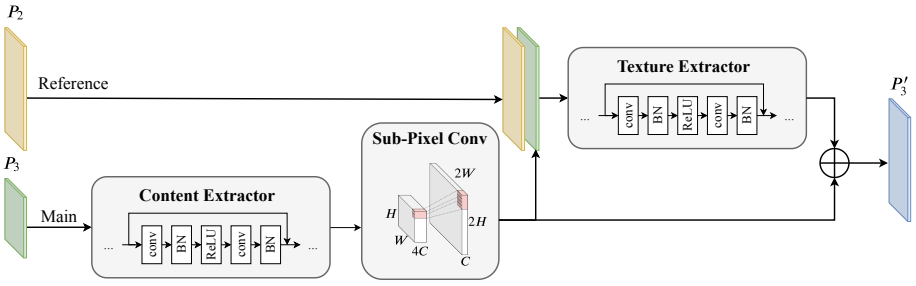


Fig. 3. The framework of FTT module. Main semantic contents of input feature P_3 are firstly extracted by a content extractor. And then we double the resolution of the content features by sub-pixel convolution. The texture extractor selects credible regional textures for small object detection from the wrap of mainstream features and reference features. Finally, residual connection helps fuse the textures with super-resolved content features to produce P_3' for the extended feature pyramid

level on which an box proposal with $w \times h = 224^2$ should be mapped into. Since the detector which follows EFPN fits various receptive fields adaptively, the receptive field drift mentioned in [25] can be ignored.

3.2 Feature Texture Transfer

Enlightened by image reference-based SR [35], we design FTT module to super-resolve features and extract regional textures from reference features simultaneously. Without FTT, noises in the 4th level P_2 of EFPN would directly pass down to the extended pyramid level, and overwhelm meaningful semantics. However, the proposed FTT output synthesizes strong semantics in upper low-resolution features and critical local details in lower high-resolution reference features, but discards disturbing noises in reference.

As shown in Fig. 3, the main input of FTT module is the feature map P_3 from the 3rd layer of EFPN, and the reference is the feature map P_2 from the 4th layer of EFPN. The output P_3' can be defined as

$$P_3' = \mathbf{E_t}(P_2 \parallel \mathbf{E_c}(P_3) \uparrow_{2\times}) + \mathbf{E_c}(P_3) \uparrow_{2\times} \tag{3}$$

where $\mathbf{E_t}(\cdot)$ denotes texture extractor component, $\mathbf{E_c}(\cdot)$ denotes content extractor component, $\uparrow_{2\times}$ here denotes double upscaling by sub-pixel convolution [29], and $\parallel$ denotes feature concatenation. The content extractor and texture extractor are both composed of residual blocks.

In the main stream, we apply sub-pixel convolution to upscale spatial resolution of the content features from the main input P_3 considering its efficiency. Sub-pixel convolution augments pixels on the dimensions of width and height via diverting pixels on the dimension of channel. Denote the feature generated by convolution layers as $F \in \mathbb{R}^{H\times W\times C\cdot r^2}$. The pixel shuffle operator in sub-pixel convolution rearranges the feature to a map of shape $rH \times rW \times C$. This

operation can be mathematically defined as

$$\mathbf{PS}(F)_{x,y,c} = F_{\lfloor x/r \rfloor, \lfloor y/r \rfloor, C \cdot r \cdot mod(y,r) + C \cdot mod(x,r) + c} \tag{4}$$

where $\mathbf{PS}(F)_{x,y,c}$ denotes the output feature pixel on coordinates (x, y, c) after pixel shuffle operation $\mathbf{PS}(\cdot)$, and r denotes the upscaling factor. In our FTT module, we adopt $r = 2$ in order to double the spatial scale.

In the reference stream, the wrap of reference feature P_2 and super-resolved content feature P_3 is fed to texture extractor. Texture extractor aims to pick up credible textures that are for small object detection and block useless noises from the wrap.

The final element-wise addition of textures and contents ensures the output integrates both semantic and regional information from input and reference. Hence, the feature map P_3' possesses selected reliable textures from shallow feature reference P_2, as well as similar semantics from the deeper level P_3.

3.3 Training Loss

Foreground-Background-Balanced Loss. Foreground-background-balanced loss is designed to improve comprehensive quality of EFPN. Common global loss will lead to insufficient learning of small object areas, because small objects only make up fractional part of the whole image. Foreground-background-balanced loss function improves the feature quality of both background and foreground by two parts: 1) global reconstruction loss 2) positive patch loss.

Global construction loss mainly enforces resemblance to the real background features, since background pixels consist most part of an image. Here we adopt l_1 loss that is commonly used in SR as global reconstruction loss L_{glob}:

$$L_{glob}(F, F^t) = ||F^t - F||_1 \tag{5}$$

where F denotes the generated feature map, and F^t denotes the target feature map.

Positive patch loss is used to draw attention to positive pixels, because severe foreground-background imbalance will impede detector performance [20]. We employ l_1 loss on foreground areas as positive patch loss L_{glob}:

$$L_{pos}(F, F^t) = \frac{1}{N} \sum_{(x,y) \in P_{pos}} ||F^t_{x,y} - F_{x,y}||_1 \tag{6}$$

where P_{pos} denotes the patches of ground truth objects, N denotes the total number of positive pixels, and (x, y) denotes the coordinates of pixels on feature maps. Positive patch loss plays the role of a stronger constraint for the areas where objects locate, enforcing learning true representation of these areas.

The foreground-background-balanced loss function L_{fbb} is then defined as

$$L_{fbb}(F, F^t) = L_{glob}(F, F^t) + \lambda L_{pos}(F, F^t) \tag{7}$$

where λ is a weight balancing factor. The balanced loss function mines true positives by improving feature quality of foreground areas, and kills false positives by improving feature quality of background areas.

Total Loss. Feature maps from 2× scale FPN are introduced to supervise the training process of EFPN. Not only the bottom extended pyramid level is under supervision, but the FTT module is under supervision as well. The overall training objective in EFPN is defined as :

$$L = L_{fbb}(P_3', P_3^{2\times}) + L_{fbb}(P_2', P_2^{2\times}) \tag{8}$$

Here $P_2^{2\times}$ is the target P_2 from 2× input FPN, and $P_3^{2\times}$ is the target P_3 from 2× input FPN.

4 Experiments

4.1 Datasets and Evaluation Metrics

Tsinghua-Tencent 100K. Tsinghua-Tencent 100K [40] is a dataset for traffic-sign detection and classification. It contains 100,000 high-resolution (2400×2400) images with 30,000 traffic-sign instances. Importantly, in *test* set, 92% of instances cover an area less than 0.2% of the entire image. The dominant majority of small objects in Tsinghua-Tencent 100K make it an excellent benchmark for small object detection.

Tsinghua-Tencent 100K benchmark divides all instances into three scales: small ($area < 32^2$), medium ($32^2 < area < 96^2$), and large ($area > 96^2$). Following the protocol in [40,25,14], we select 45 classes with more than 100 instances for evaluation, and report accuracy and recall at IoU=0.5 of three scales.

MS COCO. Microsoft COCO (MS COCO) [21] is a widely-used large-scale dataset for general object detection, segmentation and captioning. It consists of three subsets: the *train* subset with 118k images, the *val* subset with 5k images, and the *test-dev* subset with 20k images. Object detection on MS COCO confronts three challenges: (1) small objects: the size of about 65% of instances is less than 6% of the image size. (2) more instances in a single image than other similar datasets (3) different illumination and shapes of objects.

We report average precision (AP) and average recall (AR) of small category ($area < 32^2$) on *test-dev* subset, in order to highlight the detection performance of small objects. In MS COCO, the AP and AR are averaged over 10 IoU thresholds (IoU $= 0.5 : 0.05 : 0.95$), which rewards detectors with better localization.

4.2 Implementation Details

We implement our proposed EFPN with a Faster R-CNN detector, where ResNet-50 and ResNeXt-101 [32] are used as backbones. The original Faster R-CNN with FPN is firstly trained as baseline. Then, we train EFPN with backbones and heads freezed. When EFPN converges, we finetune a new detector head for the extended pyramid level with the help of OHEM [30], because there is always

Table 2. Detection performance comparison with state-of-the-art methods on Tsinghua-Tencent 100K *test* split. The symbol "*" means bells and whistles are used

Method	Small			Medium			Large		
	Acc.	Rec.	F1	Acc.	Rec.	F1	Acc.	Rec.	F1
FRCNN w FPN	80.2	86.9	83.4	94.4	94.4	94.4	92.9	93.0	92.9
Zhu et al. [40]	82.0	87.0	84.4	91.0	94.0	92.5	91.0	88.0	89.5
Li et al. [14]	84.0	89.0	86.4	91.0	96.0	93.4	91.0	89.0	90.0
Liang et al. [18]	84.0	**93.0**	88.3	95.0	97.0	96.0	**96.0**	92.0	94.0
Noh et al. [25]	84.9	92.6	88.6	94.5	**97.5**	96.0	93.3	**97.5**	95.4
EFPN(single-scale)	83.6	87.1	85.3	95.0	95.2	95.1	92.8	93.2	93.0
EFPN*(multi-scale)	**85.7**	92.3	**88.9**	**95.7**	96.7	**96.2**	94.3	97.1	**95.7**

a gap between the extended feature map P_2' and the target map P_2 from $2\times$ input image. During inference, the new detector head outputs small bounding boxes from the extended pyramid level, and the original detector head outputs medium and large bounding boxes from top 4 pyramid levels. In the end, all predicted boxes from different pyramid levels are combined to yield the final detection result.

We employ 2 residual blocks for content extractor and texture extractor in texture transfer module. The weight λ for balancing foreground and background in training loss is set to 1.

In Tsinghua-Tencent 100K experiment, we augment each class to about 1000 instances by cropping and color jitter owing to uneven numbers of different classes. Those labels not included in evaluating 45 classes are also used in training for better generalization. The model is trained on *train* split and tested on *test* split. Single-scale test uses images resized to 1400×1400, and RoIs of size smaller than 56 are assigned to the pyramid level P_2' accordingly.

In MS COCO experiment, we follow the training scheme in Detectron [9], and add data augmentation of scale and color jitter. The model is trained on *train* split, and tested on *test-dev* split. Single-scale test uses images resized to 800 on the shorter side, and RoIs with size smaller than 112 are assigned to the pyramid level P_2' accordingly.

4.3 Performance Comparison

Tsinghua-Tencent 100K We present our model results and comparison with other state-of-the-arts on Tsinghua-Tencent 100K in Table. 2. EFPN demonstrates its competence in locating and classifying small-sized objects more precisely. Compared to Faster R-CNN with ResNeXt-101-FPN, single-scale EFPN improves small object accuracy greatly by 3.4%, and improves small object recall by 0.2%. Accuracy and recall of medium objects also increase modestly by 0.6% and 0.8%, respectively. We infer the reason that some medium objects shrink after image resizing and are allocated to the extended pyramid level P_2' for detection. It's worth noting that, the 1400×1400 single-scale test of our proposed

Table 3. Comparison of single-scale test with state-of-the-art general detection methods on small category of MS COCO *test-dev* set. All results come from images resized to 800 on the shorter side

Method	Backbone	AP_S	AR_S
Noh et al. [25]	ResNet-101	16.2	-
FRCNN w FPN	ResNet-50	21.0	32.2
TridentNet [16]	ResNet-101	23.9	-
SOD-MTGAN [3]	ResNet-101	24.7	-
Libra R-CNN [26]	ResNeXt-101	25.3	-
PANet [22]	ResNeXt-101	25.4	-
FRCNN w FPN	ResNeXt-101	25.5	39.1
FSAF [39]	ResNeXt-101	26.6	-
RPDet [33]	ResNet-101-Deformable	26.6	-
FRCNN w EFPN	ResNet-50	22.0	35.9
FRCNN w EFPN	ResNext-101	**26.7**	**42.3**

EFPN outperforms the state-of-the-art 1600×1600 single-scale test of Noh et al. [25] in terms of accuracy and recall of small objects: 83.6% vs. 82.1%, and 87.1% vs. 86.6%, respectively.

Moreover, we introduce F1 score to evaluate detector's performance comprehensively. Multi-scale evaluation of EFPN yields not only best accuracy on small and medium objects, but also new state-of-the-art comprehensive F1 scores across three scales.

MS COCO. We report single-scale model results of our proposed EFPN and other general detectors on small category of MS COCO *test-dev* split. Although the quantity of small objects is smaller in MS COCO than that in Tsinghua-Tencent 100K, EFPN still enhances the ability of general object detectors dramatically. EFPN suits different backbones, and results in prominent gain on ResNet-50/ResNeXt-101 when compared with FPN. Besides, the performance of EFPN on small object detection exceeds not only other FPN variants like Libra R-CNN [26] and PANet [22], but also similar SR-based methods from Noh et al. [25] and Bai et al. [3]. Specifically, our model outperforms other state-of-the-art multi-scale general detectors on small objects, such as TridentNet [16], FSAF [39] and RPDet [33].

4.4 Ablation Studies

We conduct ablation experiments to validate efficiency of EFPN and the contribution of each network component. The backbone of ResNeXt-101 and the detector head of Faster R-CNN are adopted. All the models are trained on Tsinghua-Tencent 100K *train* split and tested on *test* split. Results are presented in Table. 4 and Table. 5.

Table 4. Efficiency validation of EFPN on Tsinghua-Tencent 100K. Here FPN-1400/FPN-2800/EFPN-1400 denotes FPN/EFPN test with 1400(1×)/2800(2×) input, and FPN-1400 + P_2-2800 means we use training target P_2 from FPN-2800 as the extended pyramid layer to form an extended feature pyramid

Model	$F1_S$	$F1_M$	$F1_L$	Runtime(s)	GPU Memory(MB)
FPN-1400	83.4	94.4	92.9	**0.45**	**2285**
FPN-2800	85.0	94.2	72.1	1.42	6349
FPN-1400 + P_2-2800	85.0	95.0	**93.1**	1.68	7217
EFPN-1400	**85.3**	**95.1**	93.0	1.05	4899

Table 5. Effect of each component in EFPN on Tsinghua-Tencent 100K *test* set

Extended level	Balanced Loss	FTT	Small			Medium			Large		
			Acc.	Rec.	F1	Acc.	Rec.	F1	Acc.	Rec.	F1
			80.2	86.9	83.4	94.4	94.4	94.4	92.9	93.0	92.9
✓			80.6	87.0	83.7	94.0	94.4	94.2	94.4	92.9	93.6
✓	✓		82.8	86.1	84.4	95.6	94.2	94.9	95.0	91.9	93.4
✓	✓	✓	83.6	87.1	85.3	95.0	95.2	95.1	92.8	93.2	93.0

EFPN is efficient on computation and memory. As shown in Table. 4, we compare the performance of EFPN with FPN test of different scales. All the models are tested on a single GTX 1080Ti GPU. Large input scale in FPN-2800 improves the performance of small objects by 1.6%, but sacrifices the performance of large objects sharply by 20.8%. Combining FPN-1400 and P_2 from FPN-2800 achieves multi-scale high performance, but the computational cost of runtime and GPU memory is more expensive than 2× test. Our proposed EFPN realizes the same high precision as FPN-1400 + P_2-2800, but with affordable computational cost between 1× test and 2× test of FPN. Through single forward propagation, EFPN efficiently achieves the precision of multi-scale FPN test.

The extended pyramid level alone is not enough. We test effect of the extended feature pyramid without FTT module and foreground-background-balanced loss, since FPN-1400 + P_2-2800 works in Table 4. ESPCN [29] is an SR method based on single image SR. We replace FTT module with a three-layer ESPCN, which realizes the same function of creating intermediate upstream feature maps and passing them to downstream lateral connection in the extension of EFPN. Supervision of P_2 and P_3 from 2× image input is realized by global l_1 loss. As shown in Table 5, it turns out that the extended pyramid level without FTT module and foreground attention has a limited effect, improving F1 score of small category by only 0.3%. Scarcely any extra missing small objects are called back by the extended pyramid level alone.

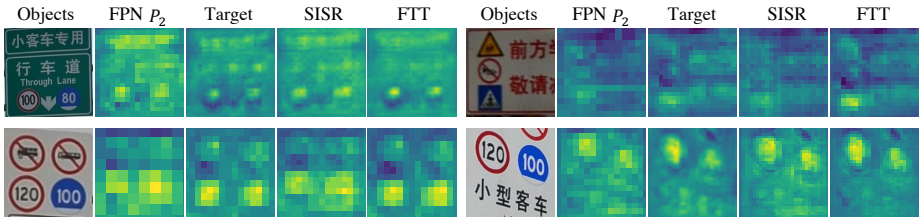


Fig. 4. Visualizing the quality of features for small object detection from different methods. Here *Target* denotes FPN P_2 from $2\times$ input, *SISR* denotes P_2' produced from ESPCN [29], and FTT denotes P_2' produced from FTT module

Foreground-background balanced loss is crucial. Balanced loss function with foreground attention is added to the extended feature pyramid with ESPCN embedded. Table 5 indicates that balanced loss improves accuracy of small category by 2.2%, thus bringing gain of 0.7% on F1 score, which indicates that, foreground-background-balanced loss encourages meaningful change on the positive areas of the extended feature maps.

we further delve into different configuration of the balancing hyper-parameter λ. When λ is set to 0.5/1.0/1.5, we get F1 score of 84.8/85.3/85.1 on small category. Hence we adopt $\lambda = 1.0$ to achieve better balance between accuracy and recall.

FTT module further enhances the quality of EFPN. Finally, we replace ESPCN with our proposed FTT module. In Table 5, it increases accuracy and recall of small category by 0.8% and 1.0%, respectively. Compared to single image SR, FTT module digs out more hard small cases. In the meanwhile, FTT module also ensures fewer false positives by reducing artifacts on the background.

In Fig. 4, we visualize the extended features from ESPCN and FTT module to further demonstrate the superiority of FTT module. We find that the features from FTT module resembles target feature more, and have clearer boundaries between object areas and background areas. More abundant regional details help detectors to distinguish positive and negative examples, thus giving better location and classification.

4.5 Qualitative Results

In Fig. 5, we present detection examples of Tsinghua-Tencent 100K and MS COCO. Compared with FPN baseline, our proposed EFPN recalls tiny and crowded instances better. Despite original ground-truth labels in MS COCO do not include all small objects, our method still detects objects existing but not labeled, which can be regarded as reasonable false positive examples.

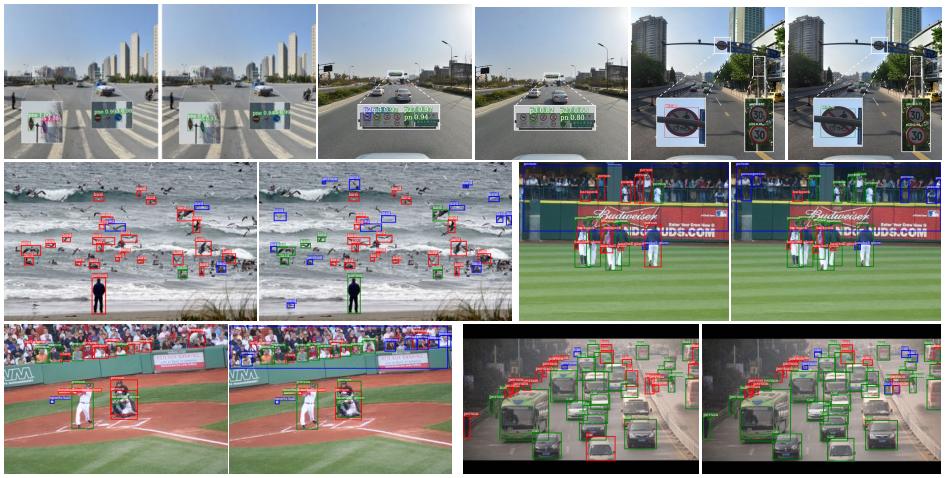


Fig. 5. Qualitative examples comparison between base model FPN and our EFPN on Tsinghua-Tencent 100K (row1) and MS COCO (row2&row3). The right in each pair denotes FPN results, while the left denotes EFPN results. The red boxes represent false negatives, the blue boxes represent false positives, and the green boxes represent true positives. Detectors of traffic-signs and general objects both profit from EFPN on challenging small object detection

5 Conclusion

In this paper, we propose EFPN to remedy the problem of small object detection. A novel FTT module is embedded in the FPN-like framework to efficiently capture more regional details for the extended pyramid level. Additionally, we design a foreground-background-balanced training loss to alleviate area imbalance of foreground and background. State-of-the-art performance on various datasets demonstrate superiority of EFPN in small object detection.

EFPN can be combined with various detectors to strengthen small object detection, which means, EFPN can be transferred to more specific situations of small object detection like face detection or satellite image detection. We would like to further explore applications of EFPN in more fields.

References

1. Amirul Islam, M., Rochan, M., Bruce, N.D., Wang, Y.: Gated feedback refinement network for dense image labeling. In: CVPR (2017)
2. Bai, Y., Zhang, Y., Ding, M., Ghanem, B.: Finding tiny faces in the wild with generative adversarial network. In: CVPR (2018)
3. Bai, Y., Zhang, Y., Ding, M., Ghanem, B.: Sod-mtgan: Small object detection via multi-task generative adversarial network. In: ECCV (2018)
4. Cai, Z., Fan, Q., Feris, R.S., Vasconcelos, N.: A unified multi-scale deep convolutional neural network for fast object detection. In: ECCV (2016)

5. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In: NIPS (2016)
6. Ghiasi, G., Lin, T.Y., Le, Q.V.: Nas-fpn: Learning scalable feature pyramid architecture for object detection. In: CVPR (2019)
7. Girshick, R.: Fast r-cnn. In: ICCV (2015)
8. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR (2014)
9. Girshick, R., Radosavovic, I., Gkioxari, G., Dollár, P., He, K.: Detectron. `https://github.com/facebookresearch/detectron` (2018)
10. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS (2014)
11. Haris, M., Shakhnarovich, G., Ukita, N.: Task-driven super resolution: Object detection in low-resolution images. arXiv preprint arXiv:1803.11316 (2018)
12. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR (2017)
13. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: ECCV (2018)
14. Li, J., Liang, X., Wei, Y., Xu, T., Feng, J., Yan, S.: Perceptual generative adversarial networks for small object detection. In: CVPR (2017)
15. Li, L., Wang, W., Luo, H., Ying, S.: Super-resolution reconstruction of high-resolution satellite zy-3 tlc images. Sensors **17**(5), 1062 (2017)
16. Li, Y., Chen, Y., Wang, N., Zhang, Z.: Scale-aware trident networks for object detection. In: ICCV (2019)
17. Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., Sun, J.: Detnet: Design backbone for object detection. In: ECCV (2018)
18. Liang, Z., Shao, J., Zhang, D., Gao, L.: Small object detection using deep feature pyramid networks. In: Pacific Rim Conference on Multimedia (2018)
19. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017)
20. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV (2017)
21. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
22. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: CVPR (2018)
23. Liu, S., Huang, D., et al.: Receptive field block net for accurate and fast object detection. In: ECCV (2018)
24. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV (2016)
25. Noh, J., Bae, W., Lee, W., Seo, J., Kim, G.: Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection. In: ICCV (2019)
26. Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W., Lin, D.: Libra r-cnn: Towards balanced learning for object detection. In: CVPR (2019)
27. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR (2016)
28. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NIPS (2015)
29. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: CVPR (2016)

30. Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: CVPR (2016)
31. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: ICCV (2019)
32. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: CVPR (2017)
33. Yang, Z., Liu, S., Hu, H., Wang, L., Lin, S.: Reppoints: Point set representation for object detection. In: ICCV (2019)
34. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: ICLR (2016)
35. Zhang, Z., Wang, Z., Lin, Z., Qi, H.: Image super-resolution by neural texture transfer. In: CVPR (2019)
36. Zheng, H., Ji, M., Wang, H., Liu, Y., Fang, L.: Crossnet: An end-to-end reference-based super resolution network using cross-scale warping. In: ECCV (2018)
37. Zhou, P., Ni, B., Geng, C., Hu, J., Xu, Y.: Scale-transferrable object detection. In: CVPR (2018)
38. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)
39. Zhu, C., He, Y., Savvides, M.: Feature selective anchor-free module for single-shot object detection. In: CVPR (2019)
40. Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., Hu, S.: Traffic-sign detection and classification in the wild. In: CVPR (2016)