Extended Feature Pyramid Network for Small Object Detection

Chunfang Deng, Mengmeng Wang, Liang Liu, Yong Liu, and Yunliang Jiang

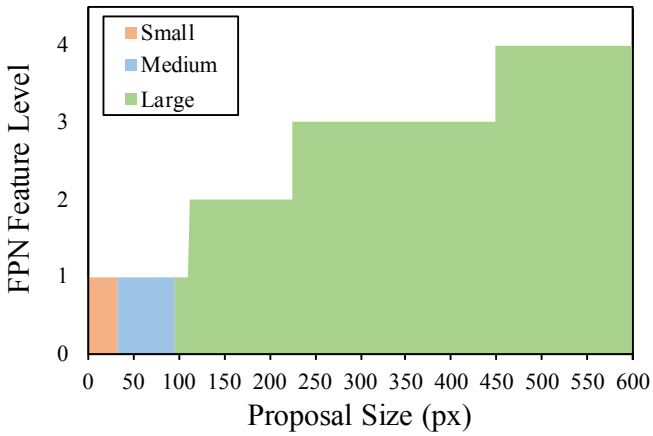
***Abstract*—Small object detection remains an unsolved challenge because it is hard to extract information of small objects with only a few pixels. While scale-level corresponding detection in feature pyramid network alleviates this problem, we find feature coupling of various scales still impairs the performance of small objects. In this paper, we propose an 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 introduce a cross resolution distillation mechanism to transfer the ability of perceiving details across the scales of the network, where a foreground-background-balanced loss function is designed 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.**

I. Introduction

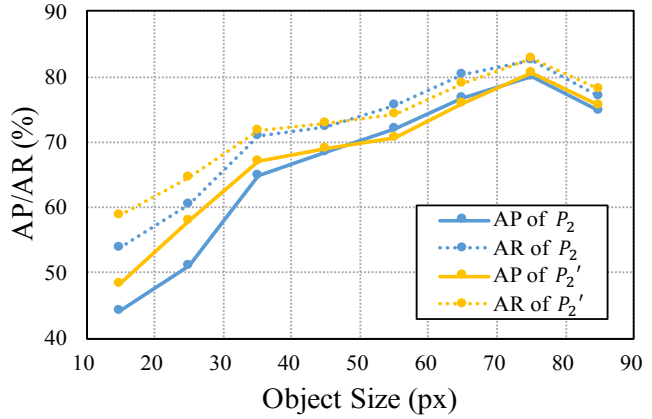
Object detection is a fundamental task of many advanced computer vision problems such as segmentation, image caption and video understanding. Over the past few years, rapid development of deep learning has boosted the popularity of CNN-based detectors, which mainly include two-stage pipelines [1]–[4] and one-stage pipelines [5]–[7]. 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) [8] 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 between pyramid level and proposal size in FPN detectors may confuse small object detection. As shown in Figure 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 Figure 1(b), the detection accuracy and recall of the FPN bottom layer fall dramatically as the object scale decreases. Figure1 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



(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.

Chunfang Deng, Mengmeng Wang, Liang Liu, Yong Liu are with the Laboratory of Advanced Perception on Robotics and Intelligent Learning, College of Control Science and Enginneering, Zhejiang University, Hangzhou 310027, China; E-mail: dengcf@zju.edu.cn, mengmengwang@zju.edu.cn, leonliuz@zju.edu.cn, yongliu@iipc.zju.edu.cn

Yunliang Jiang is with Huzhou University, Huzhou, China. E-mail: jyl@zjhu.edu.cn

detection. Early practices [9], [10] directly super-resolve the input image, but the computational cost of feature extraction in the following network would be expensive. Li et al. [11] introduce GAN [12] to lift features of small objects to higher resolution. Noh et al. [13] 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 Figure 1(b), the extended pyramid level with credible details in EFPN improves detection performance on small objects significantly.

Moreover, we introduce a cross resolution distillation mechanism, where features generated by large-scale input images are used as supervision to optimize EFPN with small-scale inputs. Under the guidance of high-quality features, the network with small-scale inputs is able to learn the knowledge how the large-scale network perceives small object information, and apply the knowledge on inner modules to improve its own performance. Thereinto, we 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 [7], 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 demonstrate 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:

- We propose an extended feature pyramid network (EFPN) which improves the performance of small object detection.
- 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.
- We introduce a cross resolution distillation strategy to learn the ability of perceiving object details from larger-scale network. A foreground-background-balanced loss function is designed in distillation to draw attention on positive pixels, alleviating area imbalance of foreground and background.
- 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.

II. Related Work

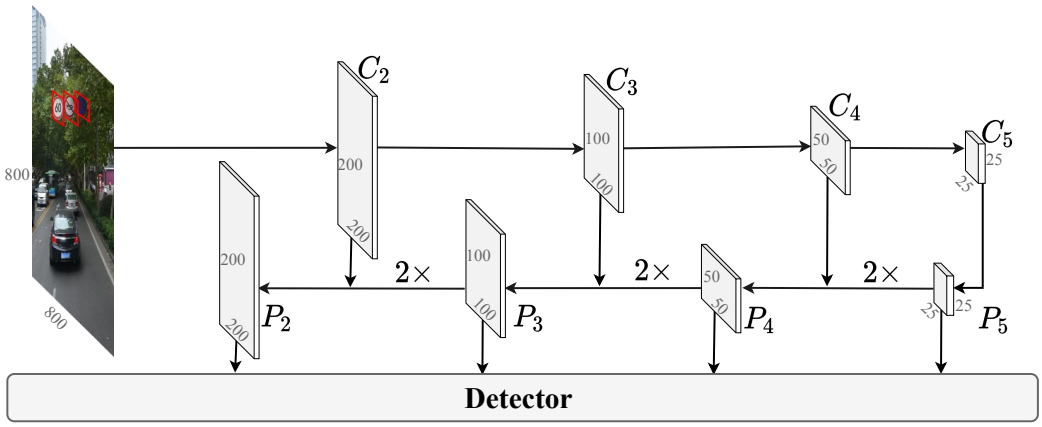
In this section, we firstly introduce deep learning based general object detectors, and then discuss relevant small object detection methods including utilizing cross-scale features and combining with super-resolution.

A. Deep Object Detectors

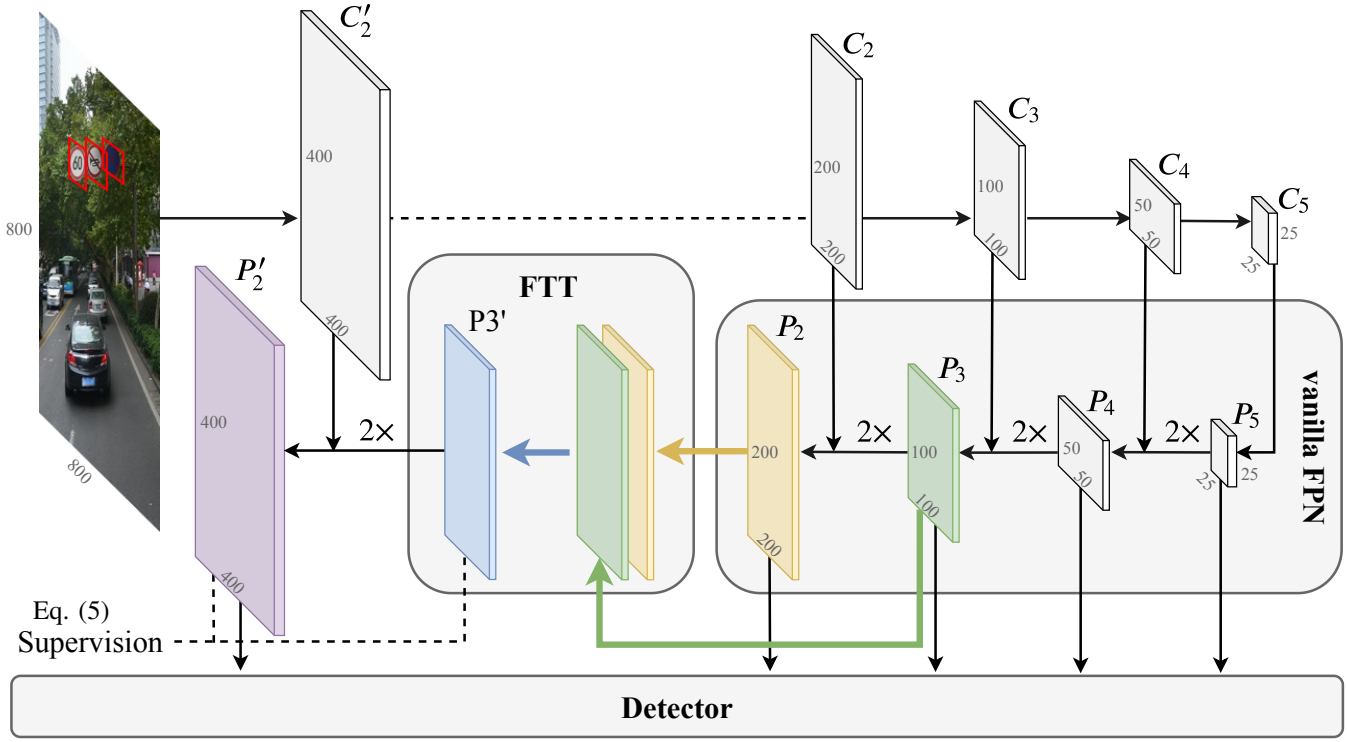
Deep learning based detectors have ruled general object detection due to their high performance. The successful two-stage methods [1]–[4] firstly generate Regions of Interest (RoIs), and then refine RoIs with a classifier and a regressor. One-stage detectors [5]–[7], another kind of prevalent detectors, directly conduct classification and localization on CNN feature maps with the help of pre-defined anchor boxes. Recently, anchor-free frameworks [14]–[17] also become increasingly popular. Despite of the development of deep object detectors, small object detection remains an unsolved challenge. Adaptive convolution module is proposed in [18] to enhance features on the area of concerned small objects. To enrich context information, dilated convolution [19] is introduced in [20]–[22] to augment receptive fields for multi-scale detection. Besides, directly adding context attention [23] is also an effective way to enhance detectors' performance. 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.

B. 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 [5] and MS-CNN [24] detect objects of different scales on different layers of CNN backbone. FPN [8] 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 [25]



(a) The pipeline of vanilla FPN.



(b) The pipeline of EFPN.

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/FPN. The dash line between C_2 and C_2' in (b) means C_2 and C_2' are parallel on the 2nd stage of backbone and share similar semantic information. 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.

adds an extra down-top pathway to pass shallow localization information up. M2Det [26] introduces a U-shape module to enhance multi-scale features. NAS-FPN [27] 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.

C. 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 [28] and images with crowded tiny faces [29]. But large-scale images are burdensome for subsequent networks. Instead of super-resolving the whole image, SOD-MTGAN [10] 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. [11] use Perceptual GAN to enhance features of small objects with the charac-

teristics of large objects. Noh et al. [13] super-resolve the whole feature map and introduce supervision signal to training process. These GAN-based frameworks are hard to train and are not efficiently combined with multi-scale feature pyramids. STDN [30] employs sub-pixel convolution on top layers of DenseNet [31] to detect small objects and meanwhile reduce network parameters, but it is based on restricted information from a single feature map. Single-image super-resolution [32]–[34] tends to fabricate regional details, which harms precise object location. Recent reference-based SR methods [35], [36] have capacity of enhancing SR images with textures or contents from reference images. Enlightened by them, we design a novel module to super-resolves features under reference and extend FPN, thus generating features with credible details more suitable for small object detection.

III. Our Approach

Since the feature coupling of various scales and unsuitable mapping between pyramid level and object size would degenerate the performance of detectors, we propose an extended feature pyramid network (EFPN) to decouple detection of objects with different sizes and allocate a more suitable feature level for small objects.

First, we construct an extended feature pyramid, which is specialized for small objects with a high-resolution feature map at the bottom. Small objects are assigned to this layer owing to its rich regional information. To strengthen the extended layer, we design a novel module named feature texture transfer(FTT), to generate intermediate features for the extended feature pyramid. Moreover, we employ cross resolution distillation where a new foreground-background-balanced loss function is proposed to further enforce learning of positive pixels. The pipeline of EFPN network and FTT module is explained in Sec. III-A and Sec. III-B, and Sec. III-C elaborates our cross resolution distillation design.

A. 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. Although 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 Figure 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 Figure 2(b).

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 Figure 2(b), aims to capture regional details for small objects. In EFPN, we denote the feature maps which share the same semantic level with C_i/P_i from vanilla FPN but with higher resolution as C_i'/P_i'. More specifically, in the extension, the 3rd and 4th pyramid layers of EFPN which are denoted by green and yellow layers respectively in Figure 2(b), 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 Figure 2(b). 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 I. 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

TABLE I
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 $2\times$ 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 $1\times$ input.

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

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

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 [8]:

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

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 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 [13] can be ignored.

B. 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 Figure 3, the main input of FTT module is the feature map P_3 from the 3rd layer of EFPN, and the reference

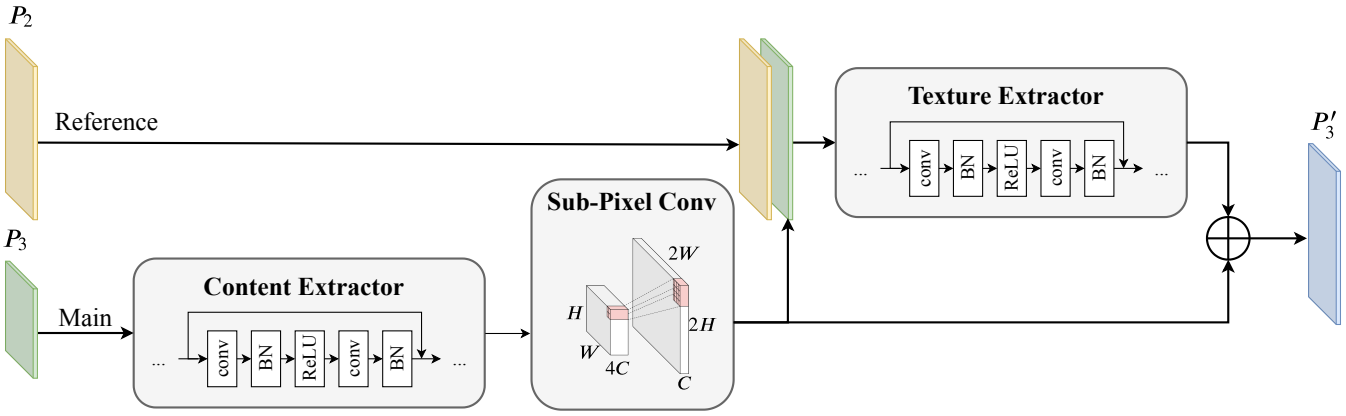


Fig. 3. The framework of FTT module. Main semantic contents of input feature P_3 are firstly extracted by a content extractor. And then we double the resolution of the content features by sub-pixel convolution. The texture extractor selects credible regional textures for small object detection from the 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.

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 [32], 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.

C. Cross Resolution Distillation

Multi-scale training and testing has already been a general trick for object detection, since using inputs of higher resolution is an effective way to improve detection performance on small objects, as shown in Figure 5. However, the detection performance saturates at a certain large scale, and the extra extensive computing resources and runtime brought by multi-scale testing are unaffordable in practical applications. To this end, we propose a mechanism termed cross resolution distillation which introduces features from high-resolution inputs as supervision signals. As shown in Figure 4, middle layers of FPN with $2\times$ scale inputs are used to guide the training of student model EFPN with $1\times$-scale inputs. For purpose of saving GPU memory, teacher model FPN and student model EFPN share the same parameter weights from top 4 layers of EFPN. Furthermore, strong knowledge distillation constraints are enforced on FEFPN, where $P_3^{2\times}$ supervises the learning of FTT module, and $P_2^{2\times}$ supervises the bottom layer of EFPN. Feature-level guidance strengthens the model by distilling knowledge of how larger-scale network deals with regional details, and teach the skill to SR modules in EFPN. During testing, our EFPN method is able to perform well on small objects, which is more efficient using low-resolution inputs than direct multi-scale practice.

The student model EFPN is trained to optimize the following loss function L:

$$L = L_{fbb}(P'_3, P_3^{2\times}) + L_{fbb}(P'_2, P_2^{2\times}) \tag{5}$$

Here $P_2^{2\times}$ is the target P_2 from $2\times$ input FPN, and $P_3^{2\times}$ is the target P_3 from $2\times$ input FPN. L_{fbb} is our proposed foreground-background-balanced loss to address area imbalance between small objects and background, and accordingly 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

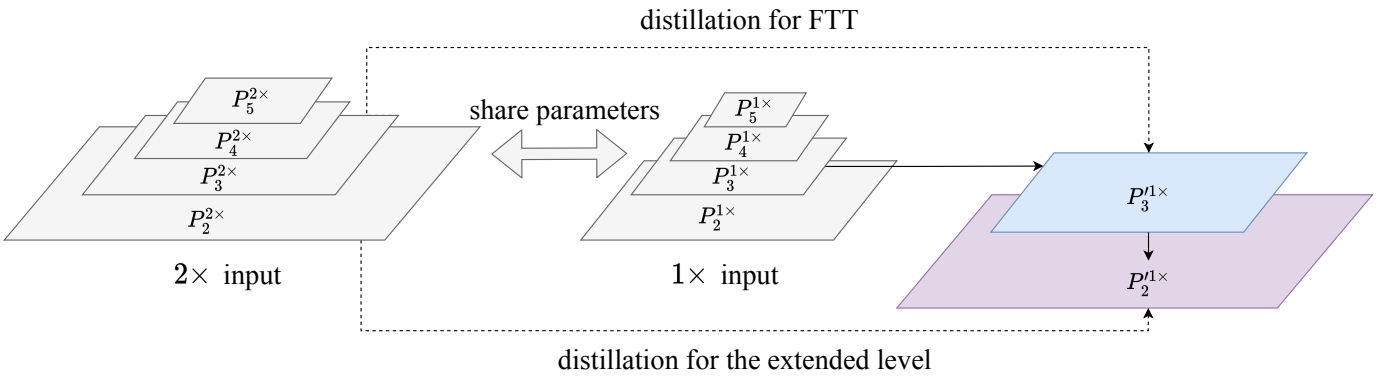


Fig. 4. Cross resolution distillation in EFPN network. The top 4 layers of EFPN would take 2×-scale as input, and generate features as targets for knowledge distillation of the FTT module and the extended pyramid level.

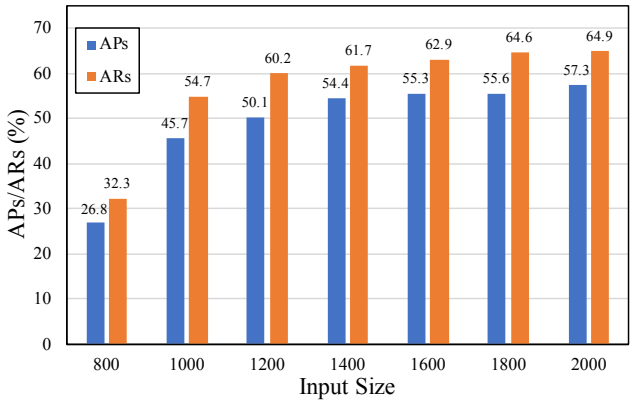


Fig. 5. Larger input scale brings performance gain in small object detection. Here *APs* and *ARs* denote the average precision and average recall of small objects on Tsinghua-Tencent 100K.

part of an image. Here we adopt l_1 loss that is commonly used in SR as global reconstruction loss L_{glob}:

$$L_{glob}(F, F^t) = ||F^t - F||_1 \tag{6}$$

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 [7]. We employ l_1 loss on foreground areas as positive patch loss L_{glob}:

$$L_{pos}(F, F^t) = \frac{1}{N} \sum_{(x,y)\in P_{pos}} ||F^t_{x,y} - F_{x,y}||_1 \tag{7}$$

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

$$L_{fbb}(F, F^t) = L_{glob}(F, F^t) + \lambda L_{pos}(F, F^t) \tag{8}$$

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.

IV. Experiments

A. Experimental Settings

1) Benchmark Datasets: We experiment our method on two benchmarks, including traffic-sign detection scenes specialized for small objects and general detection scenes. We compare our method with baselines and other state-of-the-arts on both scenes.

Tsinghua-Tencent 100K [37] 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.

Microsoft COCO(MS COCO) [38] 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.

2) Evaluation Metrics: In both Tsinghua-Tencent 100K and MS COCO, instances in images are divided into three scales according to their area: small subset with $area < 32^2$, medium subset with $32^2 < area < 96^2$, and large subset with $area > 96^2$.

For Tsinghua-Tencent 100K, following the protocol in [11], [13], [37], we select 45 classes with more than 100 instances for evaluation, and report accuracy/recall at IoU=0.5 of three scales. Moreover, we introduce F1 score to evaluate detector's performance comprehensively.

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{9}$$

TABLE II
EFFECT OF EACH COMPONENT IN EFPN ON TSINGHUA-TENCENT 100K *test* SET.

Extended Pyramid Level	Cross Resolution Distillation	Feature Texture Transfer	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
✓			80.4	86.9	83.5	94.2	94.5	94.4	93.0	93.0	93.0
✓	✓		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

Though straightforward, accuracy/recall metric highly depends on a man-made confidence threshold, because it only evaluates the results with confidence higher than the threshold. Thus we also use average precision (AP) and average recall (AR) metric, with average precision/recall over all confidence threshold, to remove the relevance to confidence.

In MS COCO, despite of confidence, the AP and AR are also averaged over 10 IoU thresholds (IoU = $0.5 : 0.05 : 0.95$), which reward detectors with better localization.

3) Implementation Details: We implement our proposed EFPN with a Faster R-CNN detector, where ResNet-50 and ResNeXt-101 [39] 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 [40], because there is always 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 random crops 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. Images are 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 [41], and add data augmentation of scale and color jitter. The model is trained on *train* split, and tested on *test-dev* split. Images are resized to 800 on the shorter side, and RoIs with size smaller than 112 are assigned to the pyramid level P_2' accordingly.

B. Ablation Studies

We conduct ablation experiments to validate the efficiency of EFPN and the contribution of each network component. Ablation studies are based on Tsinghua-Tencent 100K dataset, and Faster R-CNN with the backbone of ResNeXt-101 are adopted as base model. All the models are trained on Tsinghua-Tencent 100K *train* split and tested on *test* split. Results are presented in Table III and Table II.

TABLE III
EFFICIENCY VALIDATION OF EFPN ON TSINGHUA-TENCENT 100K. HERE FPN-1400/FPN-2800/EFPN-1400 DENOTES FPN/EFPN TEST WITH $1400(1\times)/2800(2\times)$ 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 w/o FTT	83.8	94.8	93.1	0.84	4767
EFPN-1400	**85.3**	**95.1**	93.0	1.05	4899

1) EFPN is efficient on computation and memory: As shown in Table III, 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 F1 score of small objects by 1.6%, but sacrifices the performance of large objects sharply by 20.8% on F1 score. 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 $2\times$ test. Our proposed EFPN realizes the same high precision as FPN-1400 + P_2-2800, but with affordable computational cost between $1\times$ test and $2\times$ test of FPN. Besides, we also replace the FTT module with a nearest-neighbor interpolation to test the model complexity brought by FTT. The test results indicate that the pivotal module FTT significantly improves the detection performance, but adds only a small part of computational resource. On account of feature-level SR and cross resolution distillation design, EFPN efficiently achieves the precision of multi-scale FPN test through single forward propagation.

2) The extended pyramid level alone is not enough: We test effect of the extended feature pyramid without FTT module and cross resolution distillation, since FPN-1400 + P_2-2800 performs well in Table III. ESPCN [32] is an SR method based on single image input, where a sub-pixel convolution layer is embedded as well. 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. In addition, without cross resolution distillation, only detection supervision of RPN and detector head is used. As shown in Table II, it turns out that the extended pyramid level without FTT module and knowledge distillation has a limited effect,

TABLE IV
EFFECT OF EACH COMPONENT IN CROSS RESOLUTION DISTILLATION.

Loss on the Extended Level	Balanced Loss	Loss on Feature Texture Transfer	Small			Medium			Large		
			Acc.	Rec.	F1	Acc.	Rec.	F1	Acc.	Rec.	F1
			80.4	86.9	83.5	94.2	94.5	94.4	93.0	93.0	93.0
✓			80.6	87.0	83.7	94.0	94.4	94.2	93.4	92.6	93.1
✓	✓		81.6	86.8	84.1	94.9	94.5	94.7	94.4	92.2	93.3
✓	✓	✓	82.8	86.1	84.4	95.6	94.2	94.9	95.0	91.9	93.4

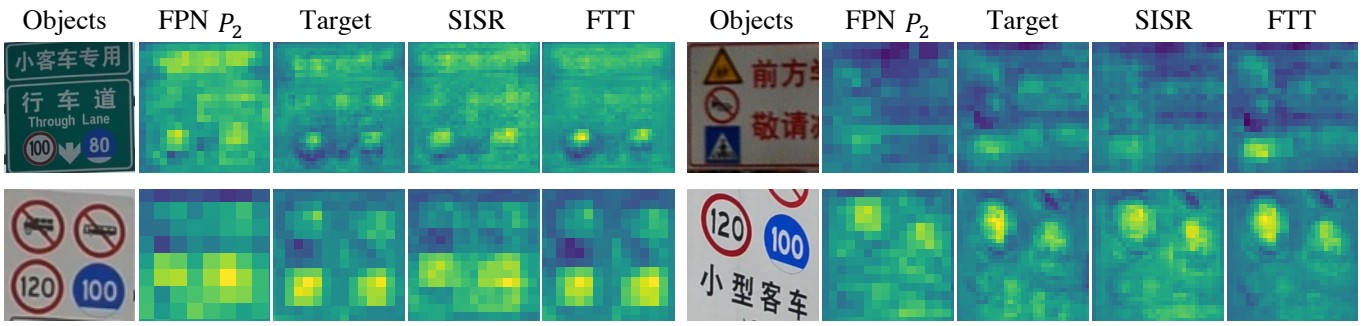


Fig. 6. Visualizing the quality of features for small object detection from different methods. Here *Target* denotes FPN P_2 from $2\times$ input, *SISR* denotes P_2' produced from ESPCN [32], and *FTT* denotes P_2' produced from FTT. Stronger resemblance to target features and clearer object boundaries are achieved by FTT, which is beneficial for preciser detection.

improving F1 score of small category by only 0.1%. Scarcely any extra missing small objects are called back by the extended pyramid level alone.

3) Cross resolution distillation is crucial: Our proposed knowledge distillation mechanism is added to the extended feature pyramid with ESPCN embedded. Under the guidance of large-scale features, the overall performance of the extended pyramid improves. As shown in Table II, the accuracy of small category rises by 2.4, thus bringing gain of 0.9 on F1 score. And the F1 scores of medium and large subsets also rise by 0.5 and 0.4 respectively. We infer the reason may be that some medium objects shrink after image resizing and are allocated to the extended pyramid level P_2' for detection.

To delve into the effect of foreground-background-balanced loss function and dual supervision on extended pyramid level and feature SR module, we conduct an ablation study inside the cross resolution distillation mechanism. We follow the setting of EFPN with ESPCN embedded. As suggested by Table IV, without foreground-background-balanced loss function, global loss on the extended pyramid level $P_2'^{2\times}$ plays a limited role, and improves F1 score of small category by merely 0.2%. The balanced loss function encourages meaningful change on the positive areas of the extended feature maps, which raises the F1 score of small/medium/large by 0.4%/0.5%/0.2%. Besides, dual supervision on both $P_2'^{2\times}$ and $P_3'^{2\times}$ further improves F1 score of small/medium/large by 0.3%/0.2%/0.1%, which suggests that dual supervision on extended pyramid level and feature SR module would force useful regional object details shift from large-scale network.

Moreover, we also attempt 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.

TABLE V
COMPARISON OF THE DETECTION PERFORMANCE ON INPUT/OUTPUT OF FTT MODULE. FTT ENRICHES THE SUPER-RESOLVED OUTPUT P_3' WITH INFORMATION ABOUT SMALL OBJECTS.

Part of FTT	Detection Layer	$F1_S$	$F1_M$	$F1_L$
Input	P_3-1400	0.0	28.8	83.0
Target	P_3-2800	10.3	73.2	71.6
Output	P_3'-1400	7.3	69.2	86.4

4) FTT module further enhances the quality of EFPN: Finally, we replace ESPCN with our proposed FTT module. In Table II, 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 order to give a more intuitive demonstration of the function of FTT module, we carry out a quantitative experiment in Table V, where objects are directly detected on the features from FTT module rather than on the pyramids. In Table V, we find that the feature P_3-1400 before SR process is dramatically poorer at small object detection than cross resolution distillation target. Our proposed FTT narrows the gap, achieving 7.3%/40.4%/3.4% performance gain on the output for small/medium/large subset respectively. Accordingly, FTT module transfers more information of small-sized objects to the SR output, and then passes it down to the extended pyramid level for better detection.

The superiority of FTT module can also be proved by Figure 6, where we visualize the features w/wo FTT module. The features from FTT module resembles target features more, and have clearer boundaries between object areas and background areas. More abundant regional details help detectors to

TABLE VI
DETECTION PERFORMANCE COMPARISON WITH STATE-OF-THE-ART METHODS ON TSINGHUA-TENCENT 100K *test* SPLIT. EFPN SINGLE-SCALE TEST USES INPUT IMAGES WITH SIZE OF 1600, AND MULTI-SCALE TEST USES INPUT IMAGES WITH SIZE FROM 1400 TO 2800.

Method	Test Scale	Small			Medium			Large			Overall		
		Acc.	Rec.	F1	Acc.	Rec.	F1	Acc.	Rec.	F1	Acc.	Rec.	F1
FRCNN w FPN	single	80.2	86.9	83.4	94.4	94.4	94.4	92.9	93.0	92.9	-	-	-
Zhu et al. [37]	single	82.0	87.0	84.4	91.0	94.0	92.5	91.0	88.0	89.5	-	-	-
Li et al. [11]	single	84.0	89.0	86.4	91.0	96.0	93.4	91.0	89.0	90.0	-	-	-
Liang et al. [42]	unknown	84.0	**93.0**	88.3	95.0	97.0	96.0	**96.0**	92.0	94.0	-	-	-
Noh et al. [13]	single	82.1	86.6	84.3	93.7	95.5	94.6	92.7	93.7	93.2	89.1	91.9	90.5
	unknown	84.9	92.6	88.6	94.5	**97.5**	96.0	93.3	**97.5**	95.4	90.6	**95.7**	93.1
EFPN	single	84.8	89.6	87.1	95.3	95.5	95.4	92.5	93.2	92.8	90.9	93.1	92.0
	multi	**85.7**	92.3	**88.9**	**95.7**	96.7	**96.2**	94.3	97.1	**95.7**	**91.6**	95.0	**93.3**

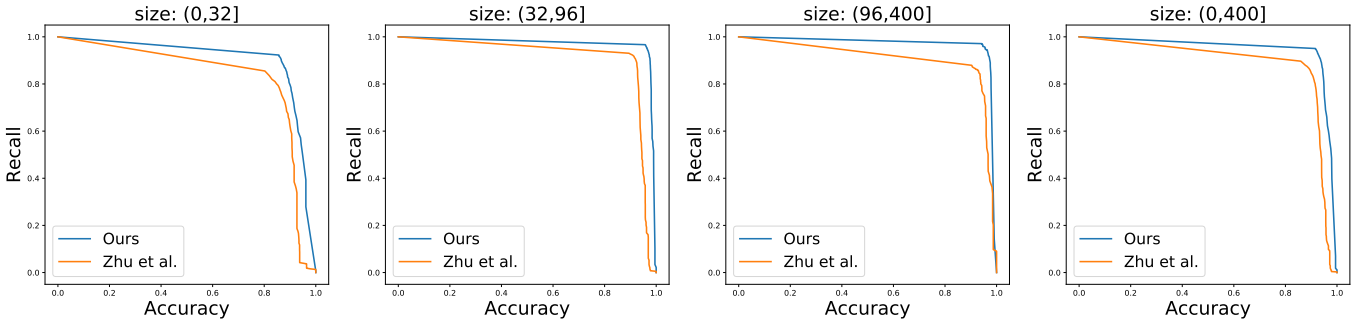


Fig. 7. Comparison of accuracy-recall curves with Zhu et al. [37] on Tsinghua-Tencent 100K *test*, for small (size:$(0, 32]$), medium (size:$(32, 96]$),large (size:$(96, 400]$) and overall (size:$(0, 400]$) category respectively.

TABLE VII
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 Type	Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
SR-Embedded	Noh et al. [13]	ResNet-101	34.2	57.2	36.1	16.2	35.7	48.1
	SOD-MTGAN [10]	ResNet-101	41.4	63.2	45.4	24.7	44.2	52.6
FPN Variants	M2Det [26]	VGG-16	41.0	59.7	45.0	22.1	46.5	53.8
	Libra R-CNN [43]	ResNeXt-101	43.0	64.0	47.0	25.3	45.6	54.6
General Detection	FSAF [17]	ResNeXt-101	42.9	63.8	46.3	26.6	46.2	52.7
	RPDet [44]	ResNet-101-Deformable	42.8	65.0	46.3	24.9	46.2	**54.7**
	Ours	ResNeXt-101	**44.6**	**64.7**	**49.4**	**28.0**	**47.5**	54.2

TABLE VIII
PERFORMANCE COMPARISON WITH FPN BASELINES ON TSINGHUA-TENCENT 100K *test* SUBSET.

Method	Backbone	AP_S	AR_S	AP_M	AP_L	mAP
FRCNN w FPN	ResNet-50	74.5	84.1	79.7	92.8	75.8
FRCNN w EFPN		**75.8**	**85.6**	**80.8**	**93.5**	**77.0**
FRCNN w FPN	ResNeXt-101	79.6	87.2	79.9	90.2	75.5
FRCNN w EFPN		**81.7**	**90.5**	**82.0**	**90.8**	**77.6**

TABLE IX
PERFORMANCE COMPARISON WITH FPN BASELINES ON MS COCO *val* SUBSET.

Method	Backbone	AP_S	AR_S	AP_M	AP_L	mAP
FRCNN w FPN	ResNet-50	20.9	31.5	40.5	48.9	37.3
FRCNN w EFPN		**22.7**	**38.1**	**41.0**	**49.4**	**38.2**
FRCNN w FPN	ResNeXt-101	24.7	36.0	45.2	53.2	41.1
FRCNN w EFPN		**26.8**	**41.5**	**46.1**	**53.8**	**42.3**

distinguish positive and negative examples, thus giving better location and classification.

5) Our method keeps superior under different metrics with different backbones on different situations: Though we observe performance gain under accuracy/recall metric on Tsinghua-Tencent 100K dataset, there are still some fluctuation on accuracy or recall which does not vary synchronously with F1 score in experiments above. This is mostly caused by the man-made confidence threshold setting in accuracy/recall metric, because it only evaluates part of the detection results, as we stated in Sec. IV-A2. Therefore, we replace the metric with AP/AR, to prove concrete effectiveness of our proposed method.

We compare the performance of our methods with the FPN baseline on Tsinghua-Tencent 100K and MS COCO in Table VIII and Table IX. With ResNeXt-101 backbone, our EFPN method ensures over 3% gain on AR of small category, and over 1% gain on AP of small category. In the meanwhile,

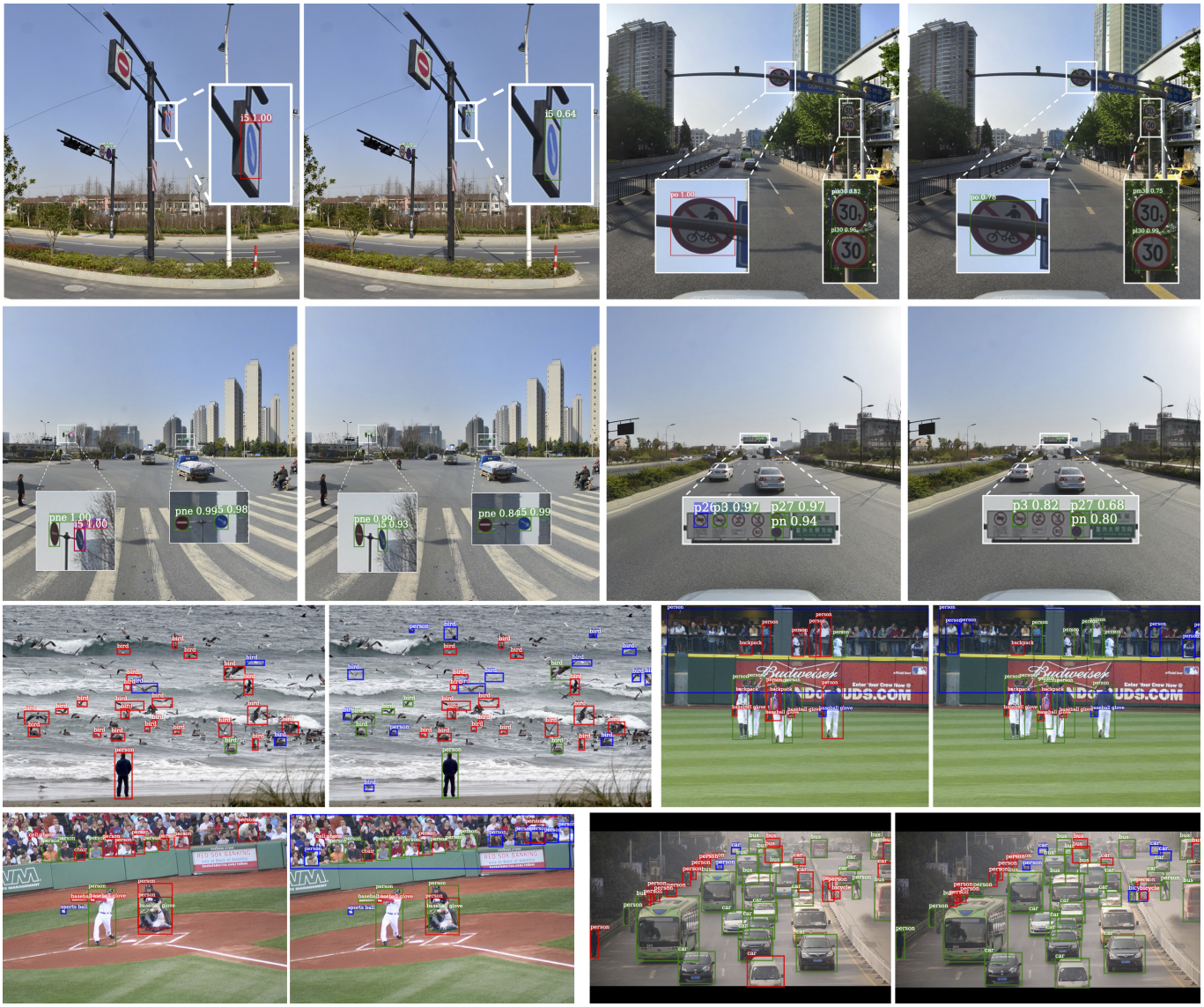


Fig. 8. Qualitative examples comparison between base model FPN and our EFPN on Tsinghua-Tencent 100K (row1&row2) and MS COCO (row3&row). The left in each pair denotes FPN results, while the right denotes EFPN results. The red boxes represent false negatives, the blue boxes represent false positives, and the green boxes represent true positives. Detectors of traffic-signs and general objects both profit from EFPN on challenging small object detection.

overall performance on large-sized objects are also guaranteed, which is shown by higher AP on medium and large subsets from EFPN.

Under the more strict and fair metric of AP/AR, our experiments prove consistent superiority and great generalization of EFPN on datasets of Tsinghua-Tencent 100K and MS COCO with different backbones. Furthermore, we observe more obvious performance gain under AP/AR than under accuracy/recall, which better proves the effectiveness of our method.

C. Comparison with State-of-the-Arts

1) Tsinghua-Tencent 100K: We present our model results and comparison with other state-of-the-arts on Tsinghua-Tencent 100K in Table VI. In addition to the metric of accuracy and recall used by previous studies [11], [13], [37], [42], we also introduce F1 score to evaluate the balance of the model's accuracy and recall.

In Table VI, our proposed method outperforms other methods not only on small scale, but also on all three scales. EFPN outperforms state-of-the-art method in [13], 88.9% *vs.* 88.6% on small subset, 96.2% *vs.* 96.0% on medium subset, 95.7% *vs.* 95.4% on large subset and 93.3% *vs.* 93.1% on overall performance. EFPN particularly demonstrates its competence in locating and classifying small-sized objects more precisely, dramatically improves the accuracy of small objects to 85.7%.

In Fig. 7, we present extended pyramid network (EFPN) results on Tsinghua-Tencent 100K *test* in the form of accuracy-recall curves. Accuracy-recall curves compute accuracy and recall over different confidence thresholds, giving demonstration of the model's comprehensive ability. Compared to Zhu et al. [37], EFPN performs better over all scales, achieving higher average precision comprehensively.

2) MS COCO.: We report single-scale model results of our method 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. Our model outperforms other state-of-the-art methods on small objects, and keeps highly competitive on larger subsets. Good generalization ability makes it robust to fit with different situations.

D. Qualitative Results

In Figure 8, we present detection examples of Tsinghua-Tencent 100K and MS COCO. Compared with FPN baseline, our proposed EFPN recalls tiny and crowded instances better, and particularly classify small objects more precisely. In the examples of MS COCO, despite original ground-truth labels do not include all small objects, our method still detects objects existing but not labeled, which can be regarded as reasonable false positive examples.

V. Conclusion

In this paper, we propose extended pyramid network to remedy the problem of small object detection, where a layer specialized for small objects are generated by the FPN-like framework. A novel feature texture transfer module is embedded in the FPN-like framework to efficiently capture more regional details for the extended pyramid level by way of reference-based feature-level SR. Additionally, we introduce cross resolution distillation mechanism to improve the quality of SR features, where 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, various backbones 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. For future work, we would like to explore practical applications of EFPN in more fields.

References

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[2] R. Girshick, "Fast r-cnn," in *ICCV*, 2015.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.

[4] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *NIPS*, 2016.

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *ECCV*, 2016.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016.

[7] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *ICCV*, 2017.

[8] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017.

[9] M. Haris, G. Shakhnarovich, and N. Ukita, "Task-driven super resolution: Object detection in low-resolution images," *arXiv preprint arXiv:1803.11316*, 2018.

[10] Y. Bai, Y. Zhang, M. Ding, and B. Ghanem, "Sod-mtgan: Small object detection via multi-task generative adversarial network," in *ECCV*, 2018.

[11] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *CVPR*, 2017.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014.

[13] J. Noh, W. Bae, W. Lee, J. Seo, and G. Kim, "Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection," in *ICCV*, 2019.

[14] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *ECCV*, 2018.

[15] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[16] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *ICCV*, 2019.

[17] C. Zhu, Y. He, and M. Savvides, "Feature selective anchor-free module for single-shot object detection," in *CVPR*, 2019.

[18] C. Chen and Q. Ling, "Adaptive convolution for object detection," *IEEE Transactions on Multimedia*, vol. 21, no. 12, pp. 3205–3217, 2019.

[19] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *ICLR*, 2016.

[20] S. Liu, D. Huang *et al.*, "Receptive field block net for accurate and fast object detection," in *ECCV*, 2018.

[21] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "Detnet: Design backbone for object detection," in *ECCV*, 2018.

[22] Y. Li, Y. Chen, N. Wang, and Z. Zhang, "Scale-aware trident networks for object detection," in *ICCV*, 2019.

[23] J. Li, Y. Wei, X. Liang, J. Dong, T. Xu, J. Feng, and S. Yan, "Attentive contexts for object detection," *IEEE Transactions on Multimedia*, vol. 19, no. 5, pp. 944–954, 2017.

[24] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *ECCV*, 2016.

[25] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *CVPR*, 2018.

[26] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, "M2det: A single-shot object detector based on multi-level feature pyramid network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 9259–9266.

[27] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "Nas-fpn: Learning scalable feature pyramid architecture for object detection," in *CVPR*, 2019.

[28] L. Li, W. Wang, H. Luo, and S. Ying, "Super-resolution reconstruction of high-resolution satellite zy-3 tlc images," *Sensors*, vol. 17, no. 5, p. 1062, 2017.

[29] Y. Bai, Y. Zhang, M. Ding, and B. Ghanem, "Finding tiny faces in the wild with generative adversarial network," in *CVPR*, 2018.

[30] P. Zhou, B. Ni, C. Geng, J. Hu, and Y. Xu, "Scale-transferrable object detection," in *CVPR*, 2018.

[31] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017.

[32] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *CVPR*, 2016.

[33] C. Tian, Y. Xu, W. Zuo, B. Zhang, L. Fei, and C. Lin, "Coarse-to-fine cnn for image super-resolution," *IEEE Transactions on Multimedia*, pp. 1–1, 2020.

[34] X. Yang, H. Mei, J. Zhang, K. Xu, B. Yin, Q. Zhang, and X. Wei, "Drfn: Deep recurrent fusion network for single-image super-resolution with large factors," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 328–337, 2019.

[35] Z. Zhang, Z. Wang, Z. Lin, and H. Qi, "Image super-resolution by neural texture transfer," in *CVPR*, 2019.

[36] H. Zheng, M. Ji, H. Wang, Y. Liu, and L. Fang, "Crossnet: An end-to-end reference-based super resolution network using cross-scale warping," in *ECCV*, 2018.

[37] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *CVPR*, 2016.

[38] 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 *ECCV*, 2014.

[39] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *CVPR*, 2017.

[40] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *CVPR*, 2016.

[41] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He, "Detectron," https://github.com/facebookresearch/detectron, 2018.
[42] Z. Liang, J. Shao, D. Zhang, and L. Gao, "Small object detection using deep feature pyramid networks," in *Pacific Rim Conference on Multimedia*, 2018.
[43] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra r-cnn: Towards balanced learning for object detection," in *CVPR*, 2019.
[44] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "Reppoints: Point set representation for object detection," in *ICCV*, 2019.