Self-Supervised Feature Augmentation for Large Image Object Detection

Article *in* IEEE Transactions on Image Processing · May 2020

DOI: 10.1109/TIP.2020.2993403

9 authors, including:

Pan Xingjia
Chinese Academy of Sciences
11 PUBLICATIONS **59** CITATIONS
SEE PROFILE

Fan Tang
Chinese Academy of Sciences
26 PUBLICATIONS **92** CITATIONS
SEE PROFILE

Weiming Dong
Chinese Academy of Sciences
110 PUBLICATIONS **1,317** CITATIONS
SEE PROFILE

Self-Supervised Feature Augmentation for Large Image Object Detection

Xingjia Pan, Fan Tang, Weiming Dong, *Member, IEEE,* Yang Gu, Zhichao Song, Yiping Meng, Pengfei Xu, Oliver Deussen, Changsheng Xu, *Fellow, IEEE,*

***Abstract*—Input scale plays an important role in modern detection frameworks, and an optimal training scale for images exists empirically. However, the optimal one usually cannot be reached in facing extremely large images under the memory constraint. In this study, we explore the scale effect inside the object detection pipeline and find that feature upsampling with the introduction of high-resolution information benefits the detection. Compared with direct input upscaling, feature upsampling trades a small performance loss for a large amount of memory savings. From these observations, we propose a self-supervised feature augmentation network, which takes downsampled images as inputs and aims to generate comparable features with the ones when feeding upscaled images to networks. We present a guided feature upsampling module, which takes downsampled images as inputs, to learn upscaled feature representations with the supervision of real large features acquired from upscaled images. In a self-supervised learning manner, we can introduce detailed information of images to the network. For an efficient feature upsampling, we design a residualized sub-pixel convolution block based on a sub-pixel convolution layer, which involves considerable information in upsampling process. Experiments on Mapillary Vistas Dataset (MVD), Cityscapes, and COCO are conducted to demonstrate the effectiveness of our method. On the MVD and Cityscapes detection benchmarks, in which the images are extremely large, our method surpasses current approaches. On COCO, the proposed method obtains comparable results to existing methods but with higher efficiency.**

***Index Terms*—self-supervise, feature augmentation, object detection, large image.**

I. Introduction

With the help of deep convolutional neural networks (CNNs), the performance of object detection (with/without segmentation masks) has been significantly improved in public challenges such as ImageNet [1] and COCO [2]. In these benchmarks, the typical size of images is 480×640. However, the input images would be at very high resolutions in some application fields, such as self-driving vehicles, where the decision-making component highly depends on visual data analysis and requires reliable, real-time semantic image understanding [3]. In Mapillary Vistas Dataset (MVD) [3], the minimum size of images is fixed at full HD, and the average image resolution is 2500×3450; consequently, the images cannot be directly used as inputs for current state-of-the-art detection/segmentation architectures due to the limitation of the memory and capabilities of GPU.

A common practice to handle the issue described above is to crop the high-resolution input into several subimages to feed deep learning pipelines, as in [4], and then fuse the output of these subimages to generate the final results. This practice, however, is extremely inefficient and time consuming. It also raises another problem on how to crop images to achieve a good tradeoff between performance and speed. Another issue is that merging different instance segmentation predictions is challenging because the pixel-level fusion will be highly influenced by the inconsistencies among different subresults.

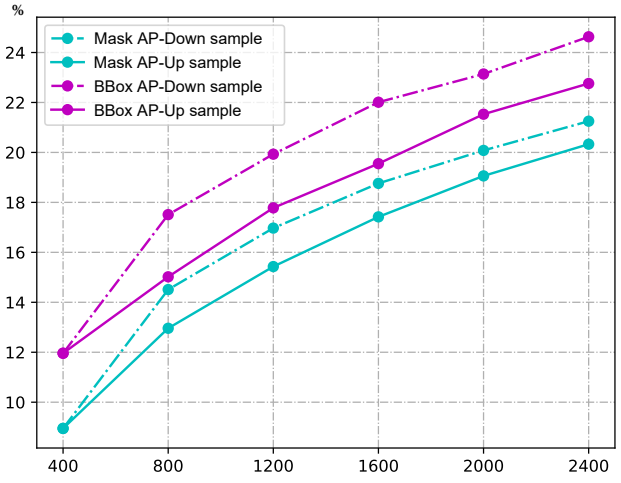


Fig. 1. Comparisons of different input sizes on the MVD detection benchmark. Training samples are obtained by downsampling the original high-resolution images to different scales or upsampling from half-sized images.

A solution is downsampling the input images to medium or small scales to meet the memory limit of the GPU. We design controlled experiments on MVD to determine the effect of the scale of input images. We first downsample the high-resolution images along with the short edge into six different scales (400, 800, 1200, 1600, 2000, and 2400 in our experiments) while keeping the original aspect ratio and then upsample the images from 400 to 800, 1200, 1600, 2000, and 2400. Both the downsampling and upsampling are done with

X. Pan and C. Xu are with NLPR, Institute of Automation, Chinese Academy of Sciences and School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China. E-mail: {panxingjia2015, changsheng.xu}@ia.ac.cn.

F. Tang is with Fosafer, Beijing, China. E-mail: tangfan@fosafer.com.

W. Dong is with NLPR, Institute of Automation, Chinese Academy of Sciences and CASIA-LLVision Joint Laboratory, Beijing, China. E-mail: weiming.dong@ia.ac.cn.

Y. Gu is with Momenta.ai, Suzhou, China. E-mail: guyang@momenta.ai.

Z. Song, Y. Meng and P. Xu are with Didi Chuxing, Beijing, China. Email: {songzhichao, mengyipingkitty, xupengfeipf}@didiglobal.com

O. Deussen is with University of Konstanz, Germany. E-mail: oliver.deussen@uni-konstanz.de

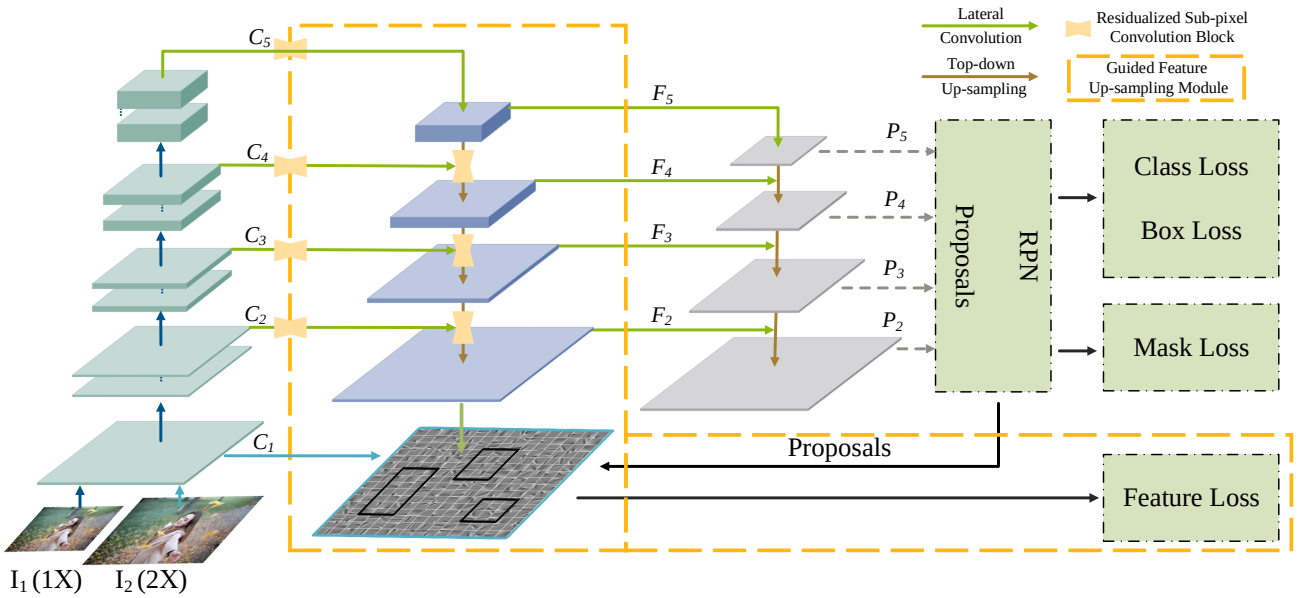


Fig. 2. Overall architecture of our approach. The components in dashed yellow bounding boxes are proposed in this work, whereas the others belong to the FPN-based Mask R-CNN framework. The classification and regression heads of RPN [5] are omitted for clarity. The proposed guided feature upsampling module locates right after the bottom-up pathway of the backbone network and learns the upsampled feature for each stage $\{C_2 - C_5\}$ guided by real larger feature C_1. The learned features ($\{F_2 - F_5\}$) are fed to the second top-down pathway and finally accessed to the detection block.

simple bilinear interpolation. Thus, 11 groups of data sets with different scales and resolutions exist. Except for images of 400-pixel, there are two different resolution images at each scale. We call the images obtained by downsampling as high-resolution ones and those by upsampling as low-resolution ones. Next, we take ResNet-101 [6] network as the backbone and train FPN-based [7] Mask-RCNN [8] using each group of images mentioned above. The scales aforementioned is the input scale to feed to networks and there is not additional resizing operation for images in each experiment. The dash lines in Figure 1 show the comparison of the detection average precision (AP) along with changes in input sizes when taking high-resolution downscaled images as input. When the input images are in small or medium sizes, the detection performance drops considerably. Comparison of the results of smallest scale with the ones of largest scale indicates an approximately 50% decrease in AP with an approximately 80% drop in scale. The solid lines in Figure 1 show the detection results using these low-resolution images as inputs. Consistent with the results indicated by the dash lines, large-sized inputs outperform the low/medium-sized inputs. From these comparisons, we have the following observations:

- With an increase in input size within a certain range, the performance of the detection network is improved accordingly.
- With the same image size, the detectors trained with downsampled (high-resolution) inputs outperform the ones with upsampled (low resolution) inputs due to the information loss in the up-/downsampling process.

In line with such observations, we can roughly conclude that we should consider two aspects, namely, increase the input scale within an appropriate range and improve the resolutions of input images, to improve the detector from the view of input.

However, increasing the input scale directly is memory-consuming and not achievable under the memory limit. Therefore, we attempt to upsample the feature map instead of upsampling the input directly. For demonstrating the validation of feature upsampling, we conduct several experiments to analyze the effect of scale heterogeneity inside the object detection framework and conclude that upsampling features can approximate the effect of directly upsampling input images (Section IV).

In this paper, we propose a self-supervised feature augmentation network (*SFANet*) for extremely large-image object detection and instance segmentation. SFANet takes downsampled images as inputs and aims to learn comparable feature maps to the ones learned from larger-scale images. We present a guided feature upsampling module that takes downsampled images as inputs to learn a high-resolution feature representation with the supervision of real large features acquired from large-scale images. In a self-supervised way, we can introduce detailed information of images to the network. For an efficient upsampling, we propose a residualized sub-pixel convolution block to assist feature augmentation with minimal information loss. Sub-pixel convolution matches the encoding process by adopting considerable information through embedding information into channels (Section V). The main contributions of this paper are as follows:

- We are the first to systematically analyze scale heterogeneity inside the object detection framework.
- We propose a novel deep pipeline called SFANet, which performs the large-image object detection task with the help of self-supervised learning.
- Our method achieves state-of-the-art results on MVD and CityScape benchmarks and acquires comparable results

with higher efficiency on COCO dataset.

II. Related Work

Object detection. From OverFeat [9] and R-CNN [10] to YOLO [11], Mask R-CNN [8] and DetNet [12], the object detection accuracy has been significantly improved. In the research field of object detection, an object detector is divided into one- and two-stage pipelines in accordance with whether the region proposal network is needed. Two-stage object detectors or region-based pipelines [5], [8], [13] generate the region of interest (RoI) first and then perform classification and regression tasks to refine the region proposal and infer the object category. Conversely, one-stage pipelines, which are represented by YOLOs [11], [14] and SSD [15], do not need separate modules to generate region proposal but regress the object location and class directly. Saliency-based methods can also detect and segment salient objects from an image [16]. From PASCAL VOC [17] and ImageNet [1] to COCO [2] and VisualGenome [18], the revolution in visual recognition benchmarks encourages or inversely squeezes the development of the state-of-the-art deep pipelines. Specifically, MVD offers a diverse street-level image dataset with pixel-accurate and instance-specific human annotations for understanding street scenes [3]. However, the images in MVD are all extremely large, which makes current state-of-the-art detectors failed due to computational limitation.

Several works in the visual understanding literature have focused on detecting objects across a large range of scales, which is a fundamental challenge in computer vision and broadly falls into three types of multiscale object detection approaches. On the one hand, detection using hyperfeature representation by simply incorporating skip features into detection, such as UNet [19], Hypercolumns [20], HyperNet [21] and ION [22], does not yield significant improvements due to the high dimensionality. On the other hand, SSD [15], MSCNN [23], RFBNet [24], and DSOD [25] combine predictions from multiple feature maps to handle objects of various sizes. However, simply detecting objects from low layers may result in poor performance because low layers possess limited semantic information. Recent works have proposed detecting objects at multiple layers to combine the best characteristics of both approaches, and the feature of each detection layer is obtained by combining features from different layers [26]–[28].

Self-supervised learning. Self-supervised learning defines an annotation-free pretext task to provide a surrogate supervision signal for feature learning. Many works on self-supervised learning exist. One class of methods removes part of the visual data (*e.g.*, color information) and tasks the network with predicting what has been removed from the rest (*e.g.*, grayscale images) in a discriminative manner [29]–[31]. Doersch et al. proposed to predict the relative position of patches cropped from images [32]. Wang et al. used the similarity of patches obtained from tracking as a self-supervised task [33] and combined the position and similarity predictions of patches [34]. Other tasks include predicting noise [35], clusters [36], [37], count [38], missing patches [39], motion segmentation labels [40], and aesthetic assessment [41]. Doersch et al. [42] proposed to jointly train four different self-supervised tasks and found it to be beneficial. The intuition is that, by solving such tasks, the trained model extracts semantic features that can be useful for other downstream tasks. In our case, we consider a multitask setting in which we train the model using joint supervision from the supervised end-task and an auxiliary self-supervised pretext task. We resize the original image to a smaller one as the network input and introduce the feature of the original image as the auxiliary supervision to help learn feature upsampling effectively.

Feature upsampling. In the hierarchical architecture of CNN, feature maps from different layers are usually at different scales, and this condition makes combining features from different layers challenging. To address this problem, researchers use a deconvolution layer or bilinear interpolation to upsample feature map. The method of upsampling mainly consists of two categories, namely, learnable and parameter free. Deconvolution is the learnable one, which was first proposed in FCN [43] and utilized in later works on semantic segmentation such as [19], [44]–[47]. Cai et al. [23] was the first to apply a deconvolution layer to improve the detection performance; it does not incur additional cost for memory and computation in contrast to input up-sampling. Unpooling and interpolation belong to the latter, which do not need additional parameters. Unpooling has rarely been involved in networks in recent years, whereas the interpolation operation is widely used [48]–[51]. Another parameter-free approach is called "sub-pixel convolution", which is derived from [52], [53] in a superresolution task and is widely broadcast to other tasks such as semantic segmentation [54]–[56] and classification [57]. They all intend to transform raw poor features to one that contains more information than them. We apply sub-pixel convolution to feature upsampling by adopting a residual design to ease the network training.

III. Overview

With the help of deep CNNs, the performance of object detection has been significantly improved. However, most works focus on small images such as the popular benchmark COCO [2], in which the typical size of images is 480 $\times$ 640. The images in some real application fields are much large. We first analyze the scale heterogeneity inside object detection framework and then propose SFANet to learn a upscaled feature representation with minimal information loss for improving the object detection on extremely large images.

Detection at heterogeneous scale. We define $\mathcal{P}_{Feat}$ and $\mathcal{P}_{Det}$ to represent deep feature representation and object detector parts of an object detection pipeline, respectively. We first assign different image scales between training and testing and then assign different scales to $\mathcal{P}_{Feat}$ and $\mathcal{P}_{Det}$. Involving high-resolution information into a heterogeneous-scale structure benefits the detection performance; and the earlier the high-resolution information is involved, the better the performance will be.

Self-supervised feature augmentation. From the observation, we propose a deep object detection framework called

TABLE I
DETECTION RESULTS ON MVD WHEN THE DETECTORS ARE TRAINED AND TESTED AT DIFFERENT SCALES.

Scale			AP
Train		Test	
$\mathcal{P}_{Feat}$	$\mathcal{P}_{Det}$		
500	500	500	10.95
500	500	1000	14.24
500	1000	1000	15.13
1000	1000	1000	16.65
1000	1000	1500	17.21
1000	1500	1500	17.93
1500	1500	1500	18.83

SFANet, to introduce high-resolution information to the framework via a self-supervised manner. We take the FPN-based Mask R-CNN as the baseline and present a guided feature upsampling module to learn a upscaled feature matched to large images effectively. Furthermore, we propose a residualized sub-pixel convolution block to assist the feature upsampling which involves considerable information and promote the learning.

IV. DETECTION AT HETEROGENEOUS SCALE

Recent CNN-based object detectors commonly upsample the input images to generate efficient results. Singh and Davis [58] studied the effect of multiplescale inputs on ImageNet and COCO. They observed that a large-scale input was important for small-object detection. The improvement by training at a large size was marginal when the input scale was over 800×1400. A similar conclusion can be made from Figure 1 on MVD. Within a certain range, the performance is continuously improved with an increment in input scale.

However, the memory is inadequate for extremely large images to reach an optimal scale. Thus, we upsample the feature map instead of the model input because the former is much more space efficient than the latter. We perform a series of experiments to verify whether the feature upsampling is effective and explore where and when to upsample features. Similar to the experiments in Figure 1, all the models involved in this section are trained on MVD by using FPN-based Mask R-CNN and take the ResNet-101 as the backbone network. We divide an object detection pipeline into two parts, namely feature representation ($\mathcal{P}_{Feat}$) and detector heads ($\mathcal{P}_{Det}$). The part $\mathcal{P}_{Feat}$ takes images as input, and outputs the feature representation. The part $\mathcal{P}_{Det}$ consists of classification and regression heads of the network. We prepare three similar groups of datasets for training. The only difference among these datasets is the image scale (500, 1000, and 1500 in our experiments and all images are obtained by downsampling from the original large images).

We train the models on each group of dataset without additional image resizing operation and obtain three models, namely, N_{500} N_{1000}, and N_{1500}. These models are evaluated on the evaluation dataset, and the test scales are the same as the ones when training. The results are listed in the first, fourth, and seventh rows of Table I. The scales of $\mathcal{P}_{Det}$ and $\mathcal{P}_{Feat}$ of each model are the same. We increase the test scale to make the detection scale heterogeneous between the training and testing phases and list the results in the second and fifth rows of Table I. Both models achieve some improvement, and this simple trick is also usually used by researchers to improve detection results. We fix the parameters $\mathcal{P}_{Feat}$ of N_{500} and N_{1000} and fine-tune $\mathcal{P}_{Det}$ by using images at scales 1000 and 1500, respectively. In the fine-tuned networks, $\mathcal{P}_{Feat}$ and $\mathcal{P}_{Det}$ are trained at different scales, which means the detection results are carried at heterogeneous scales within the training phase. The results are listed in the third and sixth rows of Table I. Finally, we fine-tune the entire N_{500} and N_{1000} using the images at scales 1000 and 1500, and the models are actually the same as N_{1000} and N_{1500}.

Comparison of the results in the second and third (fifth and sixth) rows where scale heterogeneous happens within training phase shows that increasing the scale of feature representation can indeed improve the detection performance. The first four rows (or the last four rows) are also compared. We increase the feature scale at different stages in turn, namely, after training, within the training phase ($\mathcal{P}_{Det}$), and at the very beginning of training ($\mathcal{P}_{Feat}$). The earlier upsampling is performed, the better the performance will be. However, the model complexity is also increasing rapidly as features are upsampled prematurely. This condition is a tradeoff between computational complexity and performance.

V. SELF-SUPERVISED FEATURE AUGMENTATION

On the basis of the abovementioned observation, we propose a novel deep object detection framework called SFANet, which increases the resolution of feature representation instead of model input. However, simple upsampling can't introduce the information of high-resolution images as confirmed in the experiments of Figure 1. We introduce a guided feature loss to learn the model in a self-supervised manner for utilizing the information of high-resolution images. We first introduce the network architecture of our method in Section V-A. We then describe the guided feature upsampling module in Section V-B. This module aims to learn a substantial feature representation with minimal information loss when taking downsampled images as inputs. In Section V-C, we introduce the residualized sub-pixel convolution block, which is the basic upsampling operation in the guided feature upsampling module.

A. Network Architecture

We regard the FPN-based Mask R-CNN as the baseline and the overall structure of SFANet is shown in Figure 2. The components in dashed yellow bounding boxes are proposed in this work, whereas the others belong to the FPN-based Mask R-CNN framework. The classification and regression heads of RPN [5] are omitted for clarity. The proposed guided feature upsampling module locates right after the bottom-up pathway of the backbone network and learns the upsampled feature for each stage $\{C_2 - C_5\}$ guided by real large features C_1, which are extracted from the large-scale images. The learned features ($\{F_2 - F_5\}$) are fed to the second top-down pathway and finally accessed to the detection block. Given the input scale in the training phase, we resize the original large images and

feed them ($I_1(1X)$) into the network. We prepare larger-scale images than aforementioned, in which the size is twice as large ($I_2(2X)$) to utilize the information of high-resolution images. We extract the C_1 features corresponding to larger images as the auxiliary supervision to guide the feature learning. In the testing phase, we shield the feature loss branch and do not prepare the larger-scale images because we do not need to calculate the feature loss.

Thus, our overall learning objective can be written as follows:

$$\mathcal{L} = L_{cls} + \lambda L_{reg} + \beta L_{seg} + \gamma L_{gf}, \tag{1}$$

where the first three items are the same as those in Mask R-CNN; and L_{gf} is the guide feature loss in our upsampling module, which will be elucidated below. The RPN losses are omitted here for clarity.

B. Guided Feature Upsampling Module

Feature upsampling was first applied to object detection in [23] by using a deconvolution layer and boosted the detection performance through feature approximation. Without considerably increasing memory and computation cost, feature upsampling provides another way to fit object detection for large scale images under memory limit. However, the benefits of simple feature upsampling without introducing additional supervision are very limited.

Thus, we propose a guided feature upsampling module. U-Net with skipping connection are widely used to generate upsampling images in super-resolution area [59]–[61]. Inspired from these work, we design our the guide feature upsampling module as shown in Figure 2 (marked by a dashed yellow bounding box), to efficiently and effectively improve the feature representation for object detection. We introduce the real large image feature as the auxiliary supervision to learn an effective large image representation via feature upsampling. We regard ResNet [6] as the basic structure and denote C_1 - C_5 as the outputs of conv1-conv5. C_1 is the output immediately after max pooling. The outputs of our upsampling module are defined as F_2 - F_5, corresponding to C_2 - C_5 that are $2\times$ larger in feature width and height. We formulate the guide feature loss, L_{gf}, as follows:

$$L_{gf} = \frac{1}{N_p}|C_1(X) \otimes P - G(F_2(x)) \otimes P|^2, \tag{2}$$

where X, x are the large and small images, respectively. $G(\cdot)$ represents the channel reduction operation which aims to make channel consistent between C_1 and F_2. P indicates a proposal, and $(\cdot) \otimes P$ indicates the region covered by the proposal. N_p is the number of pixels contained in the proposal regions. In our experiments, we adopt a 1×1 convolution layer to reduce the channel dimensions of F_2, which also acts as a buffer to prevent the loss from directly acting on F_2 feature layer. Instead of using a full feature map as supervision, we only backpropagate the loss of a specific region covered by the proposals to reduce the burden of network learning and remove unnecessary noise.

We adopt *lateral upsampling* and *top-down upsampling* simultaneously. The top-down pathway upsamples feature maps from a high pyramid level iteratively to F_2. These features are enhanced by the features from the bottom-up pathway via lateral upsampling. Each lateral connection upsamples feature maps from the bottom-up pathway and fuses them with the features from the top-down pathway. An efficient representation for decoder corresponding to the same level of encoder is generally difficult to learn because of the asymmetry of the depth between the encoder and decoder branches. The network adopted in the detection task is usually deep, which makes the learning of features on the top-down pathway difficult and inefficient. The proposed lateral upsampling not only can enhance the feature on the top-down pathway but also can benefit the information aggregation from each level in the feed-forward process to a high-resolution feature. The costs for memory and computation is limited because we only upsample the last residual block of each stage in our proposed module

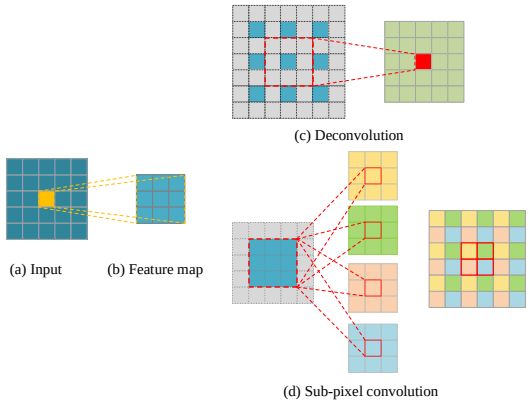
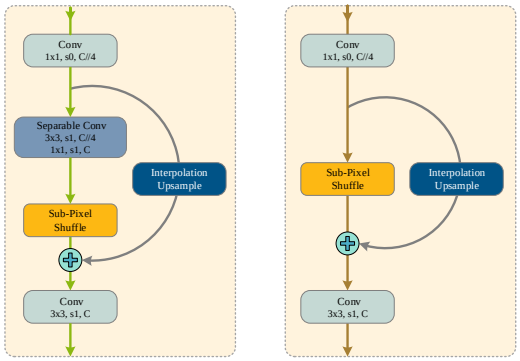


Fig. 3. Comparison between how deconvolution and sub-pixel convolution layers conduct the upsampling progress. (a) Input pixels. (b) Output feature map after 3×3 convolution. (c) Deconvolution and (d) sub-pixel convolution are two different feature upsampling ways. The deconvolution layer first pads zeros and then conducts a standard 3×3 convolution, whereas the sub-pixel embeds information to channels using a $3 \times 3 \times 4$ kernel and then rearranges it periodically.

C. Residualized Sub-pixel Convolution Block

We design a residualized sub-pixel convolution block linking the top-down and lateral connection pathways to upsample features effectively. In contrast to the method in [23] that uses a deconvolution layer for feature upsampling, our method adopts a sub-pixel convolution layer to upscale feature maps. Figure 3 illustrates the details of the difference between the two methods. Figure 3(b) is the result after convolution from Figure 3(a) with a 3×3 kernel. The center element in the original input (filled in yellow) acts on each pixel of the output feature map. When upsampling from the feature map, the deconvolution layer (Figure 3(c)) first pads zeros (marked as gray squares) and then performs a standard convolution. The center element in upsampled feature map (filled in red, corresponding to the yellow pixel in the original input) is obtained from the elements in the dashed red bounding box, that only one element (filled in blue) from the feature map. Compared to the forward convolution, the deconvolution generates upsampled feature map using partial elements, which



(a) Lateral branch (b) Top-down branch

Fig. 4. Residualized sub-pixel convolution block. The two blocks are applied at upsampling operations for (a) lateral and (b)top-down connections in our guided feature upsampling module.

loses information inevitably. In contrast to deconvolution layer, the sub-pixel convolution layer (Figure 3(d)) first embeds information to channels using a $3 \times 3 \times 4$ kernel and then rearranges it periodically to generate an upscaled map. Each element in central region (marked by red rectangle) are calculated using all the elements in Figure 3(b). The sub-pixel convolution involves all the necessary information and is obviously reasonable in the upsampling procedure. i.e., it's the larger receptive field which makes sub-pixel achieve a superior position. As addressed in [6], deep neural networks are difficult to train, whereas using a residual learning framework can ease the training of networks. We explicitly reformulate the layers as learning residual functions with reference to large features by interpolation instead of learning large features directly. For a minimal memory cost, we first use a bottleneck to aggregate information and then adopt depth-wise separable convolution [62] to replace standard convolution.

The proposed block acts slightly different in various pathways. Figure 4 shows the details of the two blocks for the top-down and lateral pathways. They consist of two branches, namely, one embeds high-resolution information into channels using sub-pixel convolution, and the other upsamples the features by interpolation. The two branches are summed element wisely, and the aliasing effect is mitigated using a convolution layer. On the contrary, we adopt a separable depth-wise convolution layer in the lateral connection pathway to reduce the memory cost. Table VI shows the results when using different upsampling methods. Our proposed residualized sub-pixel convolution block outperforms other commonly used methods.

VI. Implementation Details

We adopt the open-source code of mmdetection based on Pytorch[1] to train FPN-based Mask R-CNN. All the pretrained models we use in the experiments are supported by mmdetection. For each image, we sample 512 RoIs with a positive-to-negative ratio of 1:3 as default. Weight decay is set to 0.0001, and momentum is 0.9. Other hyperparameters slightly vary in accordance with datasets, and we detail them in respective experiments.

Training. The backbone network of SFANet is pretrained on ILSVRC dataset [1]. We use the "Xavier" method [63] to randomly initialize the parameters of all added convolution layers with zero bias. The channel number of F_i is set to the number of C_i to reuse the parameters of the FPN model as much as possible. For each minibatch, we process images in accordance with a predefined input scale (e.g., $I_1(1X)$ in Figure 2) and prepare another copy with $2\times$ larger scale (e.g., $I_2(2X)$ in Figure 2). The smaller one is normally used for training, whereas the larger one only passes through the first stage of the bottom-up pathway. We extract the C_1 feature of larger scale images as the auxiliary supervision for the guided feature branch. C_1 feature is a 64-channel vector, and it is $4\times$ smaller than the input. In contrast to the entire network, the memory cost incurred from the extraction of large features is limited. We set $\beta = 0.1$, and other settings are the same as those in the baseline method.

Inference. At the test phase, we shield the guided feature loss branch of the proposed guided feature upsampling module and do not prepare a larger scale of images anymore. The procedure is the same as the baseline method. The proposals are generated on each feature pyramid, P_i, and finally accessed to the detection result via the Fast R-CNN head outputs, followed by Non-maximum Supression (NMS). As default, we retain the top 1000 proposals when the performance of our method is evaluated on all the datasets.

VII. Experiments

Experiments are conducted on three datasets, namely MVD [3], Cityscapes [68] and COCO [2]. The testing tricks used in our experiments are common among existing methods, such as multiscale testing, horizontal flip testing, mask voting and box voting [6], [69]–[73].

A. Experiments on MVD

Dataset and Metrics. MVD is a new street-level image dataset that focuses on high-level, semantic image understanding and feature locations worldwise and is diverse in terms of weather and illumination conditions and capturing sensor characteristics [3]. This dataset contains around $25k$ high-resolution images ($18k$ train, $2k$ val, $5k$ test). The average short side is approximately 2500 pixels, which are much larger than that of COCO ($\sim$400 pixels). MVD is widely used as the basis for visual understanding tasks such as LSUN in CVPR'17 and joint COCO and Mapillary Recognition Challenge in ECCV'18.

We use AP as the main metric which is computed on the basis of instance-level segmentation per object category and averaged over a range of overlaps of 0.5:0.05:0.95 with inclusive start and end. MVD only provides pixel-level annotations. We infer the position of the bounding box in accordance with the principle of minimum closure, and all models are trained end to end.

Main Results. All models are trained on the MVD training set and tested on the val set. We set the learning rate to

[1]https://github.com/open-mmlab/mmdetection

TABLE II
COMPARISON AMONG SFANET, UCENTER [64], MASK R-CNN [8], PANET [65], CASCADE R-CNN [66], IOUNET [67] ON MVD VALIDATION SUBSET IN TERMS OF MASK AP. EACH INPUT SCALE IS THE MAX VALUE CAN BE ACHIEVED IN OUR PLATFORM GIVEN METHODS.

Method	AP	AP^{50}	AP^{75}	AP^S	AP^M	AP^L	Input Scale	Backbone
UCenter-Single	22.8	42.5	-	-	-	-	-	-
UCenter-Ensemble	23.7	43.5	-	-	-	-	-	-
Mask R-CNN	19.91	35.80	19.51	7.51	22.79	39.52	2400	ResNet-101
Mask R-CNN	21.25	38.01	19.98	6.93	23.41	42.17	1800	ResNeXt-101
PANet[test_tricks]	24.9	44.7	-	-	-	-	-	ResNet-50
PANet	21.21	40.78	20.98	9.45	24.01	40.88	2400	ResNet-101
PANet	22.83	42.32	22.53	7.82	25.72	43.40	1800	ResNeXt-101
Cascade R-CNN	20.90	37.79	20.53	8.21	23.58	41.49	2400	ResNet-101
Cascade R-CNN	22.23	40.82	20.98	7.79	24.38	**44.06**	1800	ResNeXt-101
IOUNet	20.83	37.52	20.41	8.13	23.37	41.37	2400	ResNet-101
IOUNet	22.15	40.68	20.87	7.75	24.31	43.97	1800	ResNeXt-101
SFANet	21.49	41.10	21.21	10.03	24.79	39.49	1600	ResNet-101
SFANet	23.35	42.89	23.09	9.89	26.29	41.09	1400	ResNeXt-101
SFANet[test_tricks]	**25.45**	**45.52**	**25.21**	**12.16**	**28.42**	43.21	1400	ResNeXt-101

Fig. 5. Examples of instance segmentation results of SFANet on MVD val subset.

$5e^{-3}$ for the first $90k$ iterations, and decay it to $5e^{-4}$ and $5e^{-5}$ for training another $30k$ and $20k$ iterations, respectively. We compare our methods with the state-of-the-art detectors including Mask R-CNN [8], Cascade R-CNNN [66], and IOUNet [67], and two best approaches, UCenter and PANet, in the past open challenge based on MVD. UCenter [64] reaches the first place in the LSUN 2017 challenge, both single model and the ensemble results are reported. PANet [65] reaches the first place in the COCO 2017 Challenge Instance Segmentation task. Literature [65] only provided the results of AP and AP^{50}, but the scale of input images was not mentioned; for further comparison, we retrain PANet models in accordance with the released code by the author [2]. The

[2]https://github.com/ShuLiu1993/PANet

parameters are set as default accordingly. For PANet, group normalization [74] is used in the released code, whereas batch normalization cross GPU [75] is used in [65]. We do not use either group normalization or batch normalization for fair comparison with Mask R-CNN. Table II shows the results including the original results reported in [65]. For the reimplemented or retrained deep pipelines, we only report the best results with various input scales. The best performance for each pipeline is achieved at the maximum input scale limited by GPU memory. Our method with ResNeXt-101 outperforms the other models with smaller input which is more efficient. We also show examples of instance segmentation results of our method on the validation subset in Figure 5.

TABLE III
MASK AP OF MVD INSTANCE SEGMENTATION CHALLENGE IN DIFFERENT YEARS ON TEST AND VAL SUBSETS.

Methods	AP^{test}	AP^{test}_{50}	AP^{val}	AP^{val}_{50}
LSUN′17	24.8	44.2	23.7	43.5
COCO′17	26.3	45.8	24.9	44.7
SFANet	**26.6**	**46.9**	**27.4**	**47.6**

Joint COCO and Mapillary Recognition Challenge. We apply the proposed SFANet to the instance segmentation task on MVD of the joint COCO and Mapillary Recognition Challenge held on ECCV by adopting additional training and testing tricks and achieve the first place. For training, we adopt multiscale training strategy and fine-tune the models pretrained on COCO. The tricks we adopt here are the same as those described in the main file. In the training phase, we set the short side to [800, 1000, 1300] and [800, 1000, 1400, 1600] for ResNeXt-101 and ResNet-101, respectively. For multiscale testing, we set the short edge to [600, 1000, 1200, 1600], and the maximum size is set to 2000. For ensemble, we use two ResNeXt-101 (32×8d), one ResNet-101, and two DCN-ResNeXt-101 for bounding box and mask generation. The deformable convolution layer is only used in the mask branch. We train these models with similar settings. One ResNeXt-101 (32×8d) is used as the base, with and without balanced sampling, to enhance the diversity among models. Table III shows our results, the winner of LSUN 2017, and the first-place entity of COCO 2017 tested on MVD. Compared with these state-of-the-art results, 3.7%, 2.5% absolute and 15.6%, 10.04% relative improvement are achieved on val subset. On the test dataset, we obtain 1.8%, 0.3% absolute and 8.9%, 3.8% relative improvement.

TABLE IV
RESULTS OF TRAINING AT DIFFERENT SCALES AND TESTING FPS.

Scale	Model	AP	FPS
600	SFANet	15.41	**2.30**
1000	Mask R-CNN	15.98	1.68
1000	PANet	17.02	1.05
1000	Cascade R-CNN	16.79	1.58
1000	IOUNet	16.70	1.61
1000	SFANet	20.39	1.20
1400	SFANet	**23.35**	0.86
1800	Mask R-CNN	21.25	0.82
1800	PANet	22.83	0.44

Input Scale Variety. We train models at different input scales to demonstrate the efficiency of our methods. On the basis of the ResNeXt-101 model, we set the input scale of the FPN-based Mask R-CNN and PANet to 1000 and 1800, whereas our models are trained at scales 600, 1000 and 1400. We record the run time performance of different methods. Table IV shows that our results outperform the other methods at the same input scale. Specifically, our method with input scales 600 and 1400 can achieve comparable results to those of the comparative methods trained on 1000 and 1800, respectively. Our method is also time-efficient with good performance because our small input significantly reduces the amount of calculation in the test phase. In detail, our method is slightly faster than PANet at the same scale, such as 1000. With our additional module, our method is slower than Mask R-CNN reasonably, but our performance is much better. Additional details and discussions are provided in supplemental materials.

TABLE V
ABLATION STUDY WHEN WE SEQUENTIALLY ADD THE FEATURE UPSAMPLING, GUIDED FEATURE LOSS, AND FPN STRUCTURE. THE FEATURE UPSAMPLING AND GUIDED FEATURE LOSS ARE ABBREVIATED AS FU AND GFL, RESPECTIVELY.

Methods	FU	GFL	FPN [7]	AP	AP^{50}	AP^{75}
baseline			✓	19.91	35.80	19.51
Ours	✓			19.29	34.96	18.99
	✓		✓	20.43	36.31	20.29
	✓	✓		17.69	33.57	17.18
	✓	✓	✓	21.49	41.10	21.21

Ablation Study. We provide ablation studies of SFANet by conducting controlled experiments to observe how each component affects the performance on MVD dataset. First, we determine the effects of feature upscaling and guided feature loss. Our module adopts a U-Net structure with skipping connections to restore upscaling feature. These features contain limited semantic information, and cannot handle high level recognition tasks, which require high level semantic features. Therefore, we introduce an additional top-down pathway, namely FPN structure, to promote information aggregation. Table V shows the results when we sequentially add the residualized sub-pixel convolution block and guided feature loss. The baseline method is the FPN-based Mask R-CNN with ResNet-101. Without FPN structure, the performance decreases when adding feature upsampling. The model gets worse when we continue to add guided feature loss, because the supervision, C_1, has limited semantic information and forces the restored upscaling feature to lose high level semantic information. After we add the FPN structure, the feature upsampling and guided loss achieve gains of 0.5 and 1.0, respectively. The results verify the necessity of additional top-down pathway and are consistent with our argument that feature upsampling only provides limited benefits and we must also introduce the information of high-resolution images.

Next, we evaluate the performance by adopting different upsampling techniques in our guided feature upsampling module. We compare our residualized sub-pixel convolution block with linear interpolation, deconvolutional layer and sub-

TABLE VI
ABLATION STUDY WITH DIFFERENT UPSAMPLING METHODS ON RESNET-101.

method	AP	AP^{50}	AP^{75}
Interpolation	19.86	18.23	9.57
Deconvolution	20.15	36.01	19.68
Sub-pixel convolution	20.82	40.51	20.69
Ours	**21.49**	**41.10**	**21.21**

pixel convolution layer. For the deconvolutional layer, we use the kernel of size 3×3. For the sub-pixel convolution layer, we use the same parameters as ours. Table VI shows the results using ResNet-101 as backbone feeding images at scale 1600 as inputs. Compared with a simple linear interpolation, the deconvolution and sub-pixel convolution layers show minimal improvement, whereas our residulized sub-pixel convolution block makes the feature augmentation highly effective.

TABLE VII
ABLATION STUDY WITH DIFFERENT KINDS OF SUPERVISION INFORMATION.

setting	AP	AP^{50}	AP^{75}
baseline	19.91	35.80	19.51
No_Sup	20.43	36.31	20.29
2X image	20.45	37.03	20.31
C_2	20.62	37.75	20.51
C_1	20.99	38.89	20.84
$Guided\_C_1$	**21.49**	**41.10**	**21.21**

We continue to identify the influences of different supervision in our guided feature upsampling module. Table VII shows the results under different supervision sources. The baseline method is the FPN-based Mask R-CNN with ResNet-101 without the guided feature upsampling module. No_Sup indicates training our structure without additional supervision and L_{sfa} loss. C_1 and C_2 represent using the outputs C_1 and C_2 as additional supervision in the bottom-up pathway, respectively. We add a $1 \times 1 \times 64$ convolution layer after F_2 when using the feature extracted from C_1 to keep shape consistent. The $2\times$ image refers to using the image at a large scale as supervision. We add two residualized sub-pixel convolution blocks and a 3-dimension convolution layer after F_2 to generate a superresolution image. $Guided\_C_1$ indicates that we only backpropagate the loss inside the proposals' bounding boxes from RPN. The results are as follows: 1) our pipeline promotes the detection preference; 2) additional supervision benefits the feature augmentation to achieve good detection performance; 3) guided C_1 is the best. In our opinion, the additional bottleneck convolution layer preventing the loss from directly acting on F_2 makes C_1 more robust than C_2. Compared with utilizing a $2\times$ image, using guided C_1 involves less noise in the feature space, which makes the Euclidean distance more credible. Only propagating the features covered by bounding boxes of the proposals reduces the burden of network learning and the interference of unnecessary information.

Parameter Study. We explore the effect of the loss weight β of the guided feature loss branch. The loss weight of the guided feature loss branch should not be set to 1.0 directly because the position of the proposed module is closer to the network input than detection-related losses. We select five values to evaluate the effect of the branch on the final result. In our experiments, we train the models with ResNet-101 in an end-to-end fashion at input scale 1000. Table VIII shows the results by setting different weights for the loss of our guided feature upsampling module. We acquire the best result when the weight is 0.1, but the improvement is marginal compared with the others.

TABLE VIII
PARAMETER STUDY ON LOSS WEIGHTS OF L_{sf}.

β	0.001	0.01	0.1	1.0	5.0
bbox AP	20.73	20.73	21.09	21.18	20.58
mask AP	18.63	19.12	19.31	19.20	18.79

TABLE IX
COMPARISON OF OUR METHOD WITH STATE-OF-THE-ART WORKS ON VALIDATION DATASET IN TERMS OF MASK AP AND AP^{50}. "$fine$" AND "$+COCO$" INDICATE THAT WE TRAIN MODELS ONLY ON FINE ANNOTATED DATA IN CITYSCAPES OR PRETRAINED ON COCO RESPECTIVELY.

method	AP	AP^{50}	training data
Mask R-CNN	31.5	56.8	$fine$
PANet	32.4	58.1	
Cascade R-CNN	32.6	58.9	
IOUNet	32.4	58.3	
SFANet	**33.6**	**60.3**	
Mask R-CNN	36.4	63.1	$+COCO$
PANet	37.6	64.1	
Cascade R-CNN	37.5	64.3	
IOUNet	37.2	63.9	
SFANet	**38.3**	**65.8**	

B. Experiments on Cityscapes

Dataset and Metrics. Cityscapes [68] is another popular dataset containing street scenes captured by car-mounted cameras and the images are 1024×2048. A total of $2,975$ training images, 500 validation images and $1,525$ testing images with fine annotations are available. Another 20k images have coarse annotations, which are excluded for training. We report our results on val subset. There are 8 semantic categories and they are annotated with instance masks. We evaluate the results based on AP in terms of mask and bounding box.

Results and Performance. All methods take the ResNet-FPN-50 as the backbone network. We train with an image scale (shorter side) randomly sampled from [800, 1024]. The inference is on a single scale of 1024 pixels, which is the same as [8]. We report the performance of our SFANet on the validation dataset for comparison in terms of mask AP and AP^{50}. As shown in Table IX, our method surpasses all other methods. We combine our proposed module with other FPN-based methods to further verify the effectiveness of our method. Since the proposed guided feature up-sampling module is with pyramid structure, we just need to insert the proposed module between the bottom-up pathway of ResNet and the top-down pathway of FPN. In these experiments, we half the image scale, i.e., we train models with image scales

TABLE X
WE COMBINE GUIDED FEATURE UPSAMPLING MODULE WITH STATE-OF-THE-ART WORKS AND DISPLAY THE RESULTS ON VALIDATION DATASET IN TERMS OF MASK AP AND AP^{50}. "$fine$" AND "$+COCO$" INDICATE THAT WE TRAIN MODELS ONLY ON FINE ANNOTATED DATA IN CITYSCAPES OR PRETRAINED ON COCO RESPECTIVELY. WE TRAIN MODEL WITH IMAGE A SCALE, WHICH IS RANDOMLY SAMPLED FROM [400, 512].

method	SFA	AP	AP^{50}	training data
Mask R-CNN	$--$	29.1	52.9	
Mask R-CNN	✓	30.4	54.9	
PANet	$--$	30.2	54.3	
PANet	✓	31.3	56.2	
Cascade R-CNN	$--$	30.3	55.1	$fine$
Cascade R-CNN	✓	31.5	57.3	
IOUNet	$--$	30.1	54.9	
IOUNet	✓	31.2	56.3	
Mask R-CNN	$--$	34.1	59.2	
Mask R-CNN	✓	35.6	61.3	
PANet	$--$	35.3	60.2	
PANet	✓	35.5	62.6	
Cascade R-CNN	$--$	35.2	60.1	$+COCO$
Cascade R-CNN	✓	36.4	62.5	
IOUNet	$--$	35.1	59.7	
IOUNet	✓	36.3	62.0	

randomly sampled from [400, 512]. The results of Table X illustrate that our method can roughly achieve $\sim 1.2\%$ mAP improvements and narrow the gap between lower and upper bounds.

C. Experiments on COCO

Dataset and Metrics. COCO [2] dataset is one of the most challenging and popular ones for instance segmentation and object detection. It contains $115k$ images for training and $5k$ images for validation (new split of 2017). COCO mainly focuses on recognition in natural scenes. A total of $20k$ images are used in test-dev and $20k$ images are used as test-challenge. The ground-truth labels of test-dev and test-challenge are not publicly available. There are 80 classes with pixel-level instance mask annotation. We train our models on *train-2017* subset and report the results on *val-2017* subset. We follow the standard evaluation metrics and report mask AP and box ap AP^{bb}.

TABLE XI
COMPARISON OF OUR SFANET WITH STATE-OF-THE-ART METHODS ON COCO IN TERMS OF mAP AND SPEED FPS.

method	AP	AP^{bb}	FPS
RetinaNet	–	37.2	2.67
Mask R-CNN	35.7	39.0	2.52
PANet	36.4	40.1	1.58
Cascade R-CNN	36.9	42.3	2.10
IOUNet	36.8	42.2	2.21
SFANet	**37.5**	**43.9**	**4.26**

Results and Performance. We report the performance of our SFANet on 2017val for comparison in terms of mask AP and bounding box AP^{bb}. All methods take ResNet-101 as the backbone network. The original image scale is small in COCO dataset in contrast to those in MVD and Cityscapes. The best scale can be achieved with GPU memory. According to [8], [65], the best input scale on COCO is 800, which means no further information will be involved when using larger input scales. Therefore, we set the input scale to 800 for all the other methods and use a scale randomly sampled from $[400, 800]$ for our approach to validate the proposed self-supervised feature augmentation. Table XI shows that our method can achieve comparable results with high efficiency. We provide examples of instance segmentation results of our method on the validation subset in Figure 6. For further validation, we report the results of these contrast models at input scale 400 in Table XII. In line with the discussions in Section IV, the results in Table XII act as the lower bound, whereas those in Table XI are the upper bounds. We successfully narrow the performance in the upper bounds by adopting a self-supervised feature augmentation.

TABLE XII
COMPARISON WITH FPN-BASED DETECTORS. SFA MEANS COMBINING PROPOSED GUIDED FEATURE UPSAMPLING MODULE WITH THESE ARCHITECTURES. OUR METHOD OBVIOUSLY NARROWS THE GAP BETWEEN THE UPPER AND LOWER BOUNDS. THE INPUT SCALE IS 400, WHICH IS HALF OF THE NORMAL.

method	SFA	AP	AP^{bb}	Input Scale
RetinaNet	$--$	–	33.1	400
RetinaNet	✓	–	35.2	400
Mask R-CNN	$--$	33.6	36.8	400
Mask R-CNN	✓	34.5	37.9	400
PANet	$--$	34.5	38.2	400
PANet	✓	35.6	39.0	400
Cascade R-CNN	$--$	34.8	39.8	400
Cascade R-CNN	✓	35.8	40.7	400
IOUNet	$--$	34.8	39.9	400
IOUNet	✓	35.5	41.0	400

The proposed self-supervised feature augmentation is an object detection framework that can be applied to other popular detection pipelines. We merge the proposed guided feature upsampling module with other FPN-based detectors. Table XII shows the comparisons between those with/without SFA. The guided feature upsampling module can boost the performance of these state-of-the-art architectures.

VIII. CONCLUSION

In this paper, we explore the scale effect on object detection and propose SFANet, which is mainly oriented to extremely large images. From the perspectives of upsampling features and introducing the information of high-resolution images, we design guided feature upsampling module. This module upscales features by using the proposed residualized sub-pixel convolution block and introduces high-resolution information by adding a guided feature loss branch. The feature upsampling module aims to learn substantial features matched to large images with small network inputs under the supervision of large features. We focus on the detection of extremely large images and conduct several experiments on MVD and Cityscapes to demonstrate the effectiveness of the proposed pipeline. We also merge our guided feature upsampling module with FPN-based detectors. The results on COCO and CityScapes also demonstrate the validity of the findings.

To avoid the mixture between feature upsampling and detector, we introduce an additional top-down pathway, which

Fig. 6. Examples of instance segmentation results of SFANet on COCO 2017-val subset.

makes the network a little complicated. In the future, we will explore how to combine these two pathway without arising semantic aliasing. At present, our method can only upscale $2\times$ feature. We will further study how to perform upsampling at arbitrary multiple efficiently.

ACKNOWLEDGMENT

We thank the anonymous reviewers for valuable comments. This work was supported by National Key R&D Program of China under no. 2018YFC0807500, and by National Natural Science Foundation of China under nos. 61832016 and 61672520 and 61720106006, and by CASIA-LLVision joint laboratory.

REFERENCES

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[2] 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 *Proceedings of the European Conference on Computer Vision*. Cham: Springer International Publishing, 2014, pp. 740–755.

[3] G. Neuhold, T. Ollmann, S. R. Bul, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[4] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning roi transformer for oriented object detection in aerial images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2849–2858.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[7] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[9] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *International Conference on Learning Representations (ICLR)*, April 2014.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.

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

[12] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "Detnet: A backbone network for object detection," *arXiv preprint arXiv:1804.06215*, 2018.

[13] R. Girshick, "Fast R-CNN," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[14] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 6517–6525.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *European Conference on Computer Vision*. Springer, 2016, pp. 21–37.

[16] A. Borji, M.-M. Cheng, Q. Hou, H. Jiang, and J. Li, "Salient object detection: A survey," *Computational Visual Media*, vol. 5, no. 2, pp. 117–150, 2019.

[17] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[18] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[20] B. Hariharan, P. Arbelez, R. Girshick, and J. Malik, "Object instance segmentation and fine-grained localization using hypercolumns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 627–639, April 2017.

[21] T. Kong, A. Yao, Y. Chen, and F. Sun, "Hypernet: Towards accurate region proposal generation and joint object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 845–853.

[22] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2874–2883.

[23] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *European Conference on Computer Vision*. Springer, 2016, pp. 354–370.

[24] S. Liu, D. Huang *et al.*, "Receptive field block net for accurate and fast object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 385–400.

[25] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, and X. Xue, "Dsod: Learning deeply supervised object detectors from scratch," in *IEEE International Conference on Computer Vision (ICCV)*, vol. 3, no. 6, 2017, p. 7.

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

[27] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen, "Ron: Reverse connection with objectness prior networks for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2017, p. 2.

[28] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 936–944.

[29] G. Larsson, M. Maire, and G. Shakhnarovich, "Learning representations for automatic colorization," in *European Conference on Computer Vision*. Springer, 2016, pp. 577–593.

[30] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European conference on computer vision*. Springer, 2016, pp. 649–666.

[31] ——, "Split-brain autoencoders: Unsupervised learning by cross-channel prediction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1058–1067.

[32] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1422–1430.

[33] X. Wang and A. Gupta, "Unsupervised learning of visual representations using videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2794–2802.

[34] X. Wang, K. He, and A. Gupta, "Transitive invariance for self-supervised visual representation learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1329–1338.

[35] P. Bojanowski and A. Joulin, "Unsupervised learning by predicting noise," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 517–526.

[36] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 132–149.

[37] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3733–3742.

[38] M. Noroozi, H. Pirsiavash, and P. Favaro, "Representation learning by learning to count," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5898–5906.

[39] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2536–2544.

[40] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan, "Learning features by watching objects move," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2701–2710.

[41] K. Sheng, W. Dong, M. Chai, G. Wang, P. Zhou, F. Huang, B.-G. Hu, R. Ji, and C. Ma, "Revisiting image aesthetic assessment via self-supervised feature learning," in *Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[42] C. Doersch and A. Zisserman, "Multi-task self-supervised visual learning," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2051–2060.

[43] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[44] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[45] M. A. Islam, M. Rochan, N. D. Bruce, and Y. Wang, "Gated feedback refinement network for dense image labeling," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4877–4885.

[46] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 5168–5177.

[47] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matter-simprove semantic segmentation by global convolutional network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1743–1751.

[48] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[49] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, April 2018.

[50] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[51] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2881–2890.

[52] A. Aitken, C. Ledig, L. Theis, J. Caballero, Z. Wang, and W. Shi, "Checkerboard artifact free sub-pixel convolution: A note on sub-pixel convolution, resize convolution and convolution resize," *arXiv preprint arXiv:1707.02937*, 2017.

[53] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1874–1883.

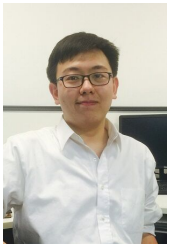
[54] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2018, pp. 1451–1460.

[55] Z. Zhang, X. Zhang, C. Peng, X. Xue, and J. Sun, "ExFuse: Enhancing feature fusion for semantic segmentation," in *European Conference on Computer Vision (ECCV)*. Springer International Publishing, 2018, pp. 273–288.

[56] T. G. Debelee, F. Schwenker, S. Rahimeto, and D. Yohannes, "Evaluation of modified adaptive k-means segmentation algorithm," *Computational Visual Media*, vol. 5, no. 4, pp. 347–361, 2019.

[57] W. Tan, B. Yan, and B. Bare, "Feature super-resolution: Make machine see more clearly," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3994–4002.
[58] B. Singh and L. S. Davis, "An analysis of scale invariance in object detection–snip," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3578–3587.
[59] X. Xu, Y. Ma, and W. Sun, "Towards real scene super-resolution with raw images," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1723–1731.
[60] X. He, Z. Mo, P. Wang, Y. Liu, M. Yang, and J. Cheng, "Ode-inspired network design for single image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1732–1741.
[61] M. S. Rad, B. Bozorgtabar, U.-V. Marti, M. Basler, H. K. Ekenel, and J.-P. Thiran, "SROBB: Targeted perceptual loss for single image super-resolution," in *IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2710–2719.
[62] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.
[63] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256.
[64] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Lsun17: insatnce segmentation task, ucenter winner team."
[65] ——, "Path aggregation network for instance segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8759–8768.
[66] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6154–6162.
[67] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 784–799.
[68] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.
[69] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3150–3158.
[70] S. Gidaris and N. Komodakis, "Object detection via a multi-region and semantic segmentation-aware cnn model," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1134–1142.
[71] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2359–2367.
[72] S. Liu, C. Lu, and J. Jia, "Box aggregation for proposal decimation: Last mile of object detection," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2569–2577.
[73] S. Zagoruyko, A. Lerer, T.-Y. Lin, P. O. Pinheiro, S. Gross, S. Chintala, and P. Dollár, "A multipath network for object detection," *arXiv preprint arXiv:1604.02135*, 2016.
[74] Y. Wu and K. He, "Group normalization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.
[75] C. Peng, T. Xiao, Z. Li, Y. Jiang, X. Zhang, K. Jia, G. Yu, and J. Sun, "MegDet: A large mini-batch object detector," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6181–6189.

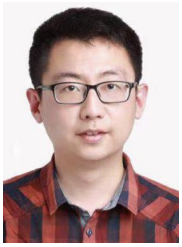
Xingjia Pan received the BSc degree in automation and finance from Nankai University in 2015. He is currently working toward the PhD degree in National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include computer graphics, computer vision and machine learning.



Fan Tang received the BSc degree in computer science from North China Electric Power University in 2013. He is currently working toward the PhD degree in National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include computer graphics, computer vision and machine learning.

Weiming Dong is a Professor in the Sino-European Lab in Computer Science, Automation and Applied Mathematics (LIAMA) and National Laboratory of Pattern Recognition (NLPR) at Institute of Automation, Chinese Academy of Sciences. He received his BSc and MSc degrees in Computer Science in 2001 and 2004, both from Tsinghua University, China. He received his PhD in Computer Science from the University of Lorraine, France, in 2007. His research interests include image synthesis and image recognition. Weiming Dong is a member of the ACM and IEEE.

Yang Gu received the BS degree and ME degree in Mechanical and Electronic Engineering from University of Electronic Science and Technology of China in 2017. His research interests include computer vision, object detection, model acceleration.



Zhichao Song received the BS degree and ME degree in Electronic Engineering from Shanghai Jiaotong University of China in 2018. His research interests include computer vision, object detection and efficient model design.

Yiping Meng received the BSc degree in Software Engineering from University of Electronic Science and Technology of China in 2013. In 2017, she received the MEng degree in National Laboratory of Pattern Recognition, Institute of Automation, Chinese of Sciences. Her research interests include computer vision, image processing, and information visualization.

Pengfei Xu is senior staff engineer at Didichuxing. He received the BS, ME and Ph. D degree in computer science from Harbin Institute of Technology, China. His research interests include computer vision and 3D construction.

Oliver Deussen graduated at Karlsruhe Institute of Technology in 1996 and worked as a postdoctoral researcher at University of Magdeburg on Non-Photorealistic Rendering. In 2000 he was appointed as a professor for Computer Graphics and Media Design by Dresden University of Technology, since 2003 he professor for Computer Graphics and Media Informatics at University of Konstanz. He is interested in a number of areas in computer graphics and information visualization. He has published several books and over 100 refereed publications, he is member of ACM Siggraph, Eurographics and Gesellschaft fuer Informatik, he is associate editor of Computer Graphics Forum and Informatik Spektrum (German Journal for Computer Science). He organized several conferences, was papers co-chair of Eurovis 2004, Eurogrpahics Symposium of Rendering 2005, NPAR 2007, Computational Aesthetics 2009 and papers Co-chair of Eurographics 2011.

Changsheng Xu is a Professor in National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences and Executive Director of China-Singapore Institute of Digital Media. His research interests include multimedia content analysis/indexing/retrieval, pattern recognition and computer vision. He has hold 30 granted/pending patents and published over 200 refereed research papers in these areas. Dr. Xu is an Associate Editor of ACM Trans. on Multimedia Computing, Communications and Applications and ACM/Springer Multimedia Systems Journal. He received the Best Associate Editor Award of ACM Trans. on Multimedia Computing, Communications and Applications in 2012 and the Best Editorial Member Award of ACM/Springer Multimedia Systems Journal in 2008. He served as Program Chair of ACM Multimedia 2009. He has served as associate editor, guest editor, general chair, program chair, area/track chair, special session organizer, session chair and TPC member for over 20 IEEE and ACM prestigious multimedia journals, conferences and workshops. He is IEEE Fellow, IAPR Fellow and ACM Distinguished Scientist.