Better to Follow, Follow to Be Better: Towards Precise Supervision of Feature Super-Resolution for Small Object Detection

Junhyug Noh[1] Wonho Bae[2] Wonhee Lee[1] Jinhwan Seo[1] Gunhee Kim[1]
[1]Seoul National University [2]University of Massachusetts Amherst
[1]{jh.noh, wonhee, jinhwanseo}@vision.snu.ac.kr, gunhee@snu.ac.kr [2]wbae@umass.edu
http://vision.snu.ac.kr/projects/better-to-follow

Abstract

In spite of recent success of proposal-based CNN models for object detection, it is still difficult to detect small objects due to the limited and distorted information that small region of interests (RoI) contain. One way to alleviate this issue is to enhance the features of small RoIs using a super-resolution (SR) technique. We investigate how to improve feature-level super-resolution especially for small object detection, and discover its performance can be significantly improved by (i) utilizing proper high-resolution target features as supervision signals for training of a SR model and (ii) matching the relative receptive fields of training pairs of input low-resolution features and target high-resolution features. We propose a novel feature-level super-resolution approach that not only correctly addresses these two desiderata but also is integrable with any proposal-based detectors with feature pooling. In our experiments, our approach significantly improves the performance of Faster R-CNN on three benchmarks of Tsinghua-Tencent 100K, PASCAL VOC and MS COCO. The improvement for small objects is remarkably large, and encouragingly, those for medium and large objects are nontrivial too. As a result, we achieve new state-of-the-art performance on Tsinghua-Tencent 100K and highly competitive results on both PASCAL VOC and MS COCO.

1. Introduction

Since the emergence of deep convolutional neural networks (CNN), the performance of object detection methods has rapidly improved. There have been two dominant approaches: two-stage proposal-based models [11, 10, 31, 5] with an advantage of accuracy and single-stage proposal-free models [29, 27, 30, 9] with an edge of speed. Despite of the recent dramatic advances in object detection, however, it is still difficult to detect objects in certain conditions, such as small, occluded or truncated. In this work, we focus on improving small object detection in the proposal-based detection framework such as Faster R-CNN [31].

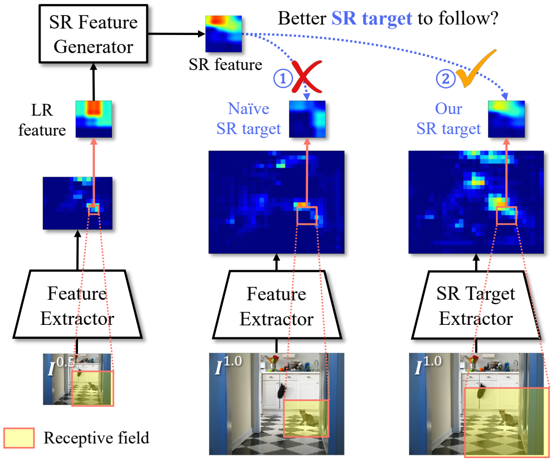


Figure 1: For feature-level super-resolution (SR), it is crucial to have direct supervision from high-resolution target features. However, if we extract them from the same feature extractor as low-resolution (LR) features, the relative receptive fields of two features are mismatched (①), which can significantly misguide the SR feature generator. We introduce SR target extractor that provides proper high-resolution features while keeping the relative receptive fields the same (②).

The proposal-based detectors fundamentally suffer from the issue that the region proposals for small objects are too small to identify. For instance, Huang *et al*. [21] show that mean Average Precision (mAP) scores of small objects are roughly 10 times lower than those of large objects. For small proposals, the region of interest (RoI) pooling layer often extracts replicated feature vectors as inputs to a box predictor, which eventually makes a prediction without enough detail information for small objects. Moreover, it is likely that the position of a RoI pooled feature and its actual position in the image are mismatched [20]. Such distortion of RoI pooling can be partly alleviated by some advanced pooling techniques such as RoI align [15] and PrRoI pooling [22]. However, they do not provide additional information a box predictor can use to better detect small objects.

To enrich the information in small proposals, some previous studies exploit image super-resolution [8, 32, 14]. Due to the serious inefficiency of super-resolving the whole image, Bai *et al.* [1] propose to super-resolve image pixels of the small proposals to be similar to those of large proposals. However, its RoI super-resolution cannot take the context information into account since it focuses only on the RoIs. This drawback can be partly resolved by the feature-level super-resolution which utilizes the context information as the features of proposals are extracted with large receptive fields of consecutive convolution operations. Particularly, Perceptual GAN [23] exploits Generative Adversarial Networks (GAN) [12] to super-resolve the features of proposals and improves the detection accuracy on small objects.

However, existing feature-level super-resolution models for small object detection have one significant limitation: *lack of direct supervision*. That is, their super-resolution models are trained without explicit target features, which results in training instability and restricted quality of super-resolution features. For the image retrieval task, Tan *et al.* [34] show that the feature-wise content loss between the pairs of low-resolution and its high-resolution features leads to better super-resolution features with faster convergence.

Not only that it is important for better training to construct proper high-resolution features as targets, our analysis also reveals that it is critical to match the relative receptive fields between the pairs, especially for small RoIs (Figure 1). That is, in the image retrieval task of [34] where only features of overall images are considered, the relative receptive fields are not much different between the pairs of high and low-resolution features. On the other hand, the difference is extremely large for small RoIs that are common in the object detection tasks, and it leads to poor quality of super-resolution of small proposals.

With this context, the contributions of this work are three-fold:

(1) We thoroughly inspect existing feature-level super-resolution methods for small object detection and discover the performance is significantly improved by (i) utilizing high-resolution target features as supervision signals and (ii) matching the relative receptive fields of input and target features.

(2) We propose a novel feature-level super-resolution approach that is orthogonally applicable on top of any proposal-based detectors with feature pooling. It fully takes advantage of direct supervision of the high-resolution target features that are created by our new target extractor, which exploits atrous convolution with requiring no additional parameters as it shares parameters with CNN backbone of the base detector. Moreover, we propose an iterative refining generator as a novel way to super-resolve features.

(3) Our approach significantly improves the performance of Faster R-CNN for small object detection on three benchmark datasets of Tsinghua-Tencent 100K [38], PASCAL VOC [6] and MS COCO [26] with various CNN backbones such as ResNet-50, ResNet-101 [16] and MobileNet [17]. The improvement for small objects is remarkably large, and encouragingly, those for medium and large objects are nontrivial too. As a result, we achieve new state-of-the-art performance on Tsinghua-Tencent 100K and highly competitive results on both PASCAL VOC and MS COCO.

2. Related Work

We review three dominant research directions for small object detection.

High-resolution images. One straightforward approach to small object detection is to generate high-resolution images as inputs to the detection model. Hu *et al.* [19] apply bilinear interpolation to obtain two times upsampled input images and Fookes *et al.* [8] use traditional super-resolution techniques to better recognize human faces. However, there are two potential problems of image-level super-resolution. First, super-resolution and detection models are often trained independently; the super-resolution model is trained to generate high-resolution images even for the parts that are not important for detection due to their independence. Second, the overall architecture can be too heavy as it takes enlarged super-resolved images as inputs, which may considerably increase inference time. Although Haris *et al.* [14] propose an end-to-end model that jointly trains super-resolution and detection models, it is still inefficient to perform super-resolution on large parts of images that are irrelevant to the detection task. Instead of super-resolving the whole images, SOD-MTGAN [1] pools RoIs first and then train the super-resolution model using those pooled RoIs. Although their work resolves both problems by focusing only on RoIs, it still does not take the context information of RoIs into account.

High-resolution features. One notable feature-level super-resolution approach for small object detection is Perceptual GAN [23]. Since it focuses on only the features of RoIs, it does not suffer from the two problems of image-level super-resolution. Moreover, since the features are extracted by the convolution with large receptive fields, the problem of SOD-MTGAN [1] is alleviated too. However, its super-resolution training can be unstable since it lacks direct supervision; there is no training pairs of low-resolution RoI features and their corresponding high-resolution features. Instead, it implicitly leverages the classification, localization and adversarial loss. For the image retrieval task, Tan *et al.* [34] add the feature-wise L_2 loss to train feature-level super-resolution model. They report that adding such stronger constraint helps the generative network produce

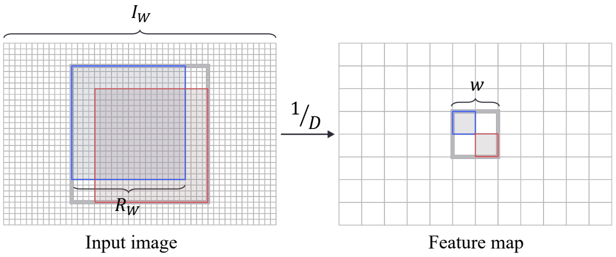


Figure 2: Suppose an input image with width of I_W and its corresponding feature map resized at ratio of $1/D$. An RoI with width of w (grey box) on the feature map has the receptive field surrounded by the grey box on the image. Meanwhile, a single feature cell on the feature map (*i.e.* blue box) has the receptive field with width of R_W on the image. The receptive fields of nearby feature cells are highly overlapped on the image space as described with shared colors.

better features with faster convergence. However, we observe that such direct supervision in [34] is not sufficient for object detection, since it may mislead the super-resolution process due to mismatch of the relative receptive fields between high and low-resolution features. In section 3, we elaborate this problem further.

Context information. Many studies have empirically proved that the context information also helps detect small objects. As demonstrated in [27], the features from the top layers in CNNs are adequate to capture large objects but too coarse to detect small objects, while the features from the bottom layers contain too specific local information which is not useful for detecting large objects but useful for small objects. Thus, many methods [2, 33, 25, 9, 35] employ additional layers to build context features from multiple layers. Another simple way to use context is to consider nearby regions too while RoI pooling. Hu *et al.* [19] extract surrounding regions along with RoIs to detect human faces since knowing the existence of human bodies in the nearby region is helpful. Relational information between objects has been also studied to enhance the detection model [18, 7, 4]. Lastly, several studies [3, 36, 37, 13] propose to use a mixture of convolution and atrous convolution layers to better segment small objects since atrous convolution layer covers larger receptive fields without losing resolution. Because of this trait, we also employ atrous convolution layers to match the relative receptive fields between high and low-resolution features. More detailed explanation is provided in section 3.

3. Mismatch of Relative Receptive Fields

In this section, we discuss why matching relative receptive fields is important to obtain adequate pairs of low-resolution input features and high-resolution target features. Based on this discussion, in the following section, we propose our novel super-resolution target extractor.

One straightforward way to obtain the pairs is to take a large RoI from the original image and its smaller version from the downsampled image [34]. Unfortunately, the features of these pairs do not exactly match up in terms of relative receptive fields. In order to clearly see why such discrepancy occurs, we present an intuitive example in Figure 2 with notations. Considering only one horizontal axis for easiness of discussion, the absolute receptive field (ARF) for the feature of an RoI with width of w is

$$ARF(w) = R_W + (w-1) \times D. \tag{1}$$

The relative receptive field (RRF), defined as ARF relative to the size of an image I_W, is

$$RRF(w, I_W) = \left(R_W + (w-1) \times D\right)/I_W. \tag{2}$$

Let us discuss how RRF differs as the input image resizes. In $\times 0.5$ downsampled input image, the width of the image is $I_W/2$ and that of the RoI on the feature map is $w/2$. We define the discrepancy in RRF (DRRF) of the RoIs between the original and downsampled images as

$$DRRF_{1/2}(w, I_W) = \frac{RRF(w/2, I_W/2)}{RRF(w, I_W)} = 2 - \frac{w}{c+w} \tag{3}$$

where $c = R_W/D - 1$ is a constant. Eq.(3) is easily derivable from Eq.(2).

According to Eq.(3), as w approaches to 0, DRRF converges to 2, while it goes to 1 as w increases. That is, for a small RoI, the relative receptive field (RRF) of the same RoI can be as $\times 2$ as different between the original and downsampled images. On the other hand, the RRFs become similar if the size of a proposal is sufficiently large. For example, for an RoI with $w = 4$ from the input image with $I_W = 1600$, if we use Faster R-CNN with ResNet-50 backbone where $R_W = 291$ and $D = 16$, then $DRRF_{1/2}(4, 1600)$ is close to 1.8. That is, the RRF of the RoI from the downsampled image is around 1.8 times larger than that from the original image. Tan *et al.* [34] deal with the image retrieval task where the entire image features are super-resolved and thus the discrepancy in RRF is not significant. On the contrary, for the super-resolution of small RoIs for detection as in our work, the discrepancy in RRF is critically large and it can seriously misguide the super-resolution model.

4. Our Approach

We propose a novel method that enhances the quality of feature super-resolution for small object detection, based on two key ideas: (i) direct supervision for the super-resolution generator and (ii) the receptive field matching via atrous convolution. We introduce four additional components on top of the base detector model: SR feature generator and

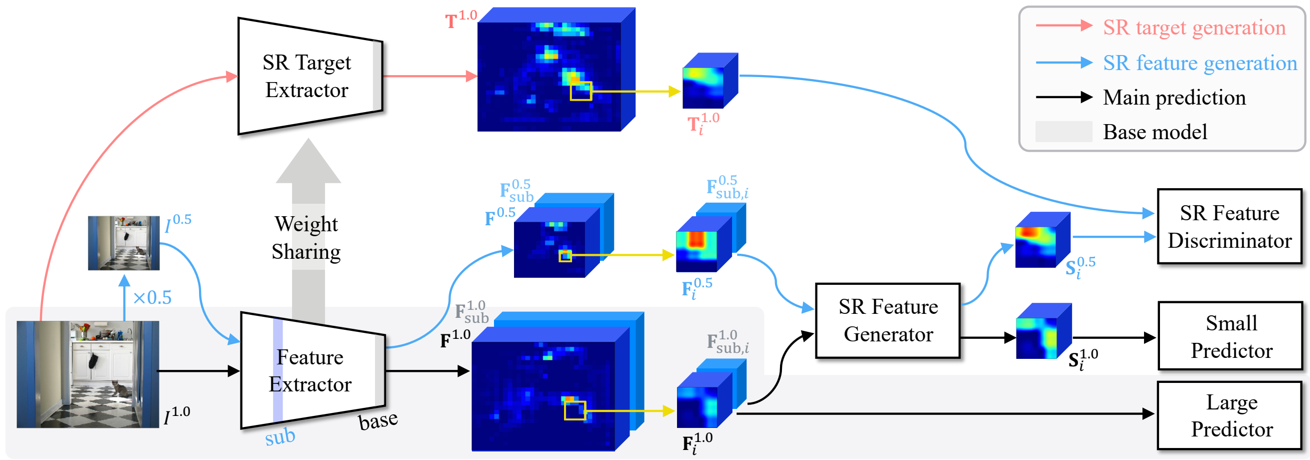
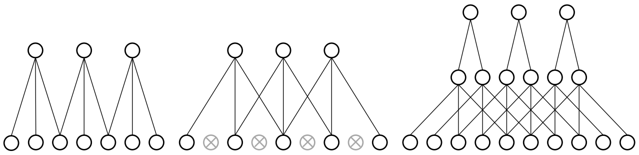


Figure 3: Overall model architecture. Four new components are proposed on top of the base detector model: SR target extractor (section 4.1), SR feature generator and discriminator (section 4.2), and small predictor. As a GAN-based model, the SR feature generator learns to create high-resolution features under the guidance of the SR feature discriminator using the features from the SR target extractor as targets (section 4.3). At inference (specified as *main prediction* arrows), a large proposal is directly passed to the large predictor for classification and localization, while a small proposal is first super-resolved by the SR feature generator and then passed to the small predictor (section 4.4).

discriminator, SR target extractor and small predictor. As a GAN-based model, the SR feature generator produces high-resolution features under the guidance of the SR feature discriminator using the features from the SR target extractor as targets. Additionally, the small predictor is a replica of the predictor in the base detector, which we call as the large predictor. The large predictor computes the confidence of classification and localization for large proposals as done in normal detectors, whereas the small predictor carries out the same task for small proposals that are enhanced first by the SR feature generator. We set the thresholds for the small proposals as (32×32) for Tsinghua-Tencent and (96×96) for VOC and COCO datasets. Figure 3 shows the overall architecture of our model. We explain the model based on Faster R-CNN [31], although our approach is integrable with any proposal-based detector with feature pooling[1].

4.1. Super-resolution Target Extractor

We denote the original input image by $I^{1.0}$ and its $\times 0.5$ downsampled image by $I^{0.5}$. We use $\mathbf{F}_i^{1.0}$ to denote the feature for the i-th RoI from the original image. In section 3, we reveal that it is not a good idea to use $\mathbf{F}_i^{1.0}$ as a super-resolution target for $\mathbf{F}_i^{0.5}$. Instead, we need to extract proper high-resolution target feature denoted by $\mathbf{T}_i^{1.0}$ that has similar RRF with low-resolution feature $\mathbf{F}_i^{0.5}$. To this end, we introduce an additional CNN feature extractor named *super-resolution target extractor* to generate $\mathbf{T}_i^{1.0}$ as in Figure 3. We let the SR target extractor share the same parameters with the CNN backbone (*i.e*. the normal feature extractor in the base detector), because they should not produce different features by channel for the same input.



(a) CONV (b) Atrous CONV (c) Atrous CONV + POOL

Figure 4: Connections between input and output nodes. (a) One convolution layer with filter size of 3 and stride of 2. (b) One atrous convolution layer with filter size of 3, stride of 2 and rate of 2. (c) The same atrous convolution layer as (b) with stride of 1, followed by one pooling layer with filter size of 2 and stride of 2.

One important requirement for the SR target extractor is to adequately address RRF at every layer where the receptive fields are expanded. In regular CNNs, the receptive fields are expanded whenever applying convolution or pooling layers whose filter sizes are greater than 1. Thus, our SR target extractor should be designed to cover the same expanded receptive fields whenever either of those layers are used in the CNN backbone. For parameter-free pooling layers, it can be easily achieved by increasing the filter size. However, for convolution layers, increasing the filter size is not valid as it makes the parameters different from those of the CNN backbone. Therefore, we employ atrous (dilated) convolution layer [3], which involves the same number of parameters as a regular convolution layer while its receptive fields are controlled by a dilation rate. We apply atrous

[1] Most two-stage proposal-based detectors use *feature pooling*, while a few models exploit *score pooling* such as RFCN [5].

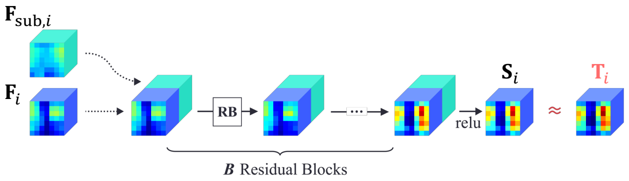


Figure 5: The *super-resolution feature generator*. It transforms the low-resolution input feature $\mathbf{F}_i$ into a super-resolution feature $\mathbf{S}_i$, with additional input $\mathbf{F}_{\text{sub},i}$. It iteratively refines the features via B residual blocks, each of which is the element-wise sum of the input feature and residual with two CONV layers as filters. At the end, only $\mathbf{F}_i$ part is sliced to be $\mathbf{S}_i$.

convolution layers with dilation rate of 2 at every convolution layer with the filter size greater than 1 on the CNN backbone.

One additional treatment is for the stride. As shown in Figure 4(a), if the stride of convolution layer in the CNN backbone is not 1 (*e.g.* 2), it is not valid to simply use the same stride size for atrous convolution because it skips every other pixel as shown in Figure 4(b). This problem can be solved by applying atrous convolution with stride of 1 and then max pooling with 2 as in Figure 4(c).

In summary, the SR target extractor consists of atrous convolution and pooling layers arranged to keep the same RRF as the CNN backbone while sharing the same parameters. The feature $\mathbf{T}_i^{1.0}$ from the SR target extractor is a better target to train the super-resolution model than $\mathbf{F}_i^{1.0}$ from the CNN backbone. Furthermore, $\mathbf{T}_i^{1.0}$ covers larger receptive fields than $\mathbf{F}_i^{1.0}$; they contain more context information that can be useful for better small object detection.

4.2. Super-resolution Feature Generator

Our feature-level super-resolution model is based on Generative Adversarial Networks (GAN) [12]. Its ultimate goal is to transform the pooled features $\mathbf{F}_i^{1.0}$ of small proposals to super-resolved features $\mathbf{S}_i^{1.0}$. In order to make a pair of low-resolution and high-resolution target features, we first downsample the original image at ×0.5, obtain $\mathbf{F}_i^{0.5}$ for i-th proposal and pair it with $\mathbf{T}_i^{1.0}$ generated from the SR target extractor. That is, the super-resolution feature generator in Figure 5 is learned to iteratively refine $\mathbf{F}_i^{0.5}$ into the super-resolution features $\mathbf{S}_i^{0.5}$ so that $\mathbf{S}_i^{0.5}$ is as similar to $\mathbf{T}_i^{1.0}$ as possible. For this objective, we design the feature-wise content ℓ_2 loss as

$$\mathcal{L}_{cont} = \sum_{i=1}^{N} \|\mathbf{T}_i^{1.0} - \mathbf{S}_i^{0.5}\|_2^2. \tag{4}$$

During this process, as input to the generator, we use both the features from the former layer $\mathbf{F}_{\text{sub},i}^{0.5}$ (**sub** layer) and the latter layer $\mathbf{F}_i^{0.5}$ (**base** layer). Since $\mathbf{F}_i^{0.5}$ only contains coarse and low-frequency information for a small RoI, we supplement its fine and high-frequency information $\mathbf{F}_{\text{sub},i}^{0.5}$ from the former layer.

For the SR feature discriminator, we use a multi-layer perceptron (MLP) with three layers. The discriminator is trained to be able to distinguish between $\mathbf{T}_i^{1.0}$ and $\mathbf{S}_i^{0.5}$, while the generator is trained to transform $\mathbf{F}_i^{0.5}$ into $\mathbf{S}_i^{0.5}$ indistinguishable from $\mathbf{T}_i^{1.0}$. Hence, the generator and discriminator respectively minimize

$$\mathcal{L}_{gen} = -\sum_{i=1}^{N} \log D(\mathbf{S}_i^{0.5}) \tag{5}$$

$$\mathcal{L}_{dis} = -\sum_{i=1}^{N} \left(\log D(\mathbf{T}_i^{1.0}) + \log\left(1 - D(\mathbf{S}_i^{0.5})\right)\right). \tag{6}$$

One final remark is when we construct low-resolution input and high-resolution target features for different losses, we use thresholding. Although different thresholds are used for different losses, we apply the following general rule; we discard the high-resolution features if they are too small to be used as targets, and discard the low-resolution features if they are large enough to have no need of super-resolution. We apply different thresholds for different datasets as specified in the overview of section 4. The more detailed explanation on thresholding is provided in supplementary material.

So far, we have discussed how the generator refines the low-resolution feature $\mathbf{F}_i^{0.5}$ to be similar to the target feature $\mathbf{T}_i^{1.0}$. However, our ultimate goal is to better detect small objects; thus, we need to train the generator to super-resolve features in a way that they indeed help detect small objects well. To this end, we further train the generator as follows. After the generator produces the super-resolved features $\mathbf{S}_i^{1.0}$ from $\mathbf{F}_i^{1.0}$, we input it to the small box predictor. Then, we compute the classification loss ($\mathcal{L}_{cls}$) and localization loss ($\mathcal{L}_{loc}$) of the box predictor as in [31], and flow the gradient signals to the generator for fine-tuning.

4.3. Training

We first train the base detector model, which consists of the feature extractor, region proposal network (RPN) and the large predictor. Then, the generator and discriminator are alternatively trained using the features ($\mathbf{F}_i^{1.0}$, $\mathbf{F}_i^{0.5}$ and $\mathbf{T}_i^{1.0}$) while freezing the feature extractors and RPN. The generator is trained under the guidance of the weighted sum of generator, content, classification and localization losses while the discriminator is trained only from the discriminator loss. Along with the GAN structure, the small predictor is simultaneously trained using the super-resolved features $\mathbf{S}_i^{1.0}$ from the classification and localization losses. Notice that we initialize the SR target extractor and small predictor using the weights of the feature extractor and the large predictor of the base detector, respectively.

Model	Small			Medium			Large			Overall		
	Rec.	Acc.	F1	Rec.	Acc.	F1	Rec.	Acc.	F1	Rec.	Acc.	F1
MobileNet [17]	56.1	72.9	63.4	85.1	**84.3**	84.7	90.9	**83.6**	**87.1**	74.7	80.7	77.5
+ Ours	**62.7**	**81.7**	**71.0**	**87.6**	84.0	**85.7**	**91.5**	82.1	86.5	**78.5**	**83.1**	**80.7**
ResNet-50 [16]	68.8	81.9	74.9	90.8	93.1	91.9	91.6	92.3	91.9	82.5	89.2	85.7
+ Ours	**78.2**	**86.5**	**82.2**	**94.7**	**93.8**	**94.3**	**93.6**	**93.0**	**93.3**	**88.4**	**91.1**	**89.7**
ResNet-101 [16]	69.8	81.5	75.2	90.9	93.5	92.2	92.4	92.0	92.2	83.1	**89.2**	86.0
+ Ours	**86.6**	**82.1**	**84.3**	**95.5**	**93.7**	**94.6**	**93.7**	**92.7**	**93.2**	**91.9**	89.1	**90.5**

Table 1: Overall performance on Tsinghua-Tencent 100K *test* dataset. Our proposed model achieves consistent improvement over the base models regardless of the backbone structures.

Model	Small			Medium			Large			Overall		
	Rec.	Acc.	F1	Rec.	Acc.	F1	Rec.	Acc.	F1	Rec.	Acc.	F1
Zhu *et al.* [38]	87.0	82.0	84.4	94.0	91.0	92.5	88.0	91.0	89.5	–	–	–
Perceptual GAN [23]	89.0	84.0	86.4	96.0	91.0	93.4	89.0	91.0	89.9	–	–	–
Liang *et al.* [24]	**93.0**	84.0	88.3	97.0	**95.0**	95.9	92.0	**96.0**	93.9	–	–	–
SOS-CNN [28]	–	–	–	–	–	–	–	–	–	93.0	90.0	91.5
FRCNN [31] + ResNet-101 [16]	80.3	81.6	80.9	94.5	94.8	94.7	94.3	92.6	93.5	89.1	89.7	89.4
+ Ours	92.6	**84.9**	**88.6**	**97.5**	94.5	**96.0**	**97.5**	93.3	**95.4**	**95.7**	**90.6**	**93.1**

Table 2: Performance comparison with the state-of-the-art models on Tsinghua-Tencent 100K *test* dataset.

Once both generator and discriminator converge, we further fine-tune the small and large predictors while freezing all the others. Fine-tuning is useful for the small predictor because it is trained only on super-resolved features which may not be perfectly identical to the target features. It also helps further boost up the performance by focusing solely on classification and localization losses. The large predictor is fine-tuned only with large proposals since the features of the small proposals are no longer passed into it.

4.4. Inference

Once training is done, the inference is much simpler. We only use the SR feature generator and the small predictor on top of the base model, which corresponds to the *main prediction* part in Figure 3. Given an input image $I^{1.0}$, we obtain the features from the CNN backbone $\mathbf{F}^{1.0}$. If the feature proposal is large, the large predictor takes it to make prediction on its class and location. On the other hand, if the feature proposal is small, it is super-resolved first using the SR feature generator and passed into the small predictor.

5. Experiments

We evaluate the performance of our approach on Faster R-CNN [31] as the base network with various backbones (ResNet-50, ResNet-101 [16], and MobileNet [17]) on three benchmark datasets of Tsinghua-Tencent 100K [38], PASCAL VOC [6] and MS COCO [26]. We present more experimental results and analysis in the supplementary file.

5.1. Results on Tsinghua-Tencent 100K

Tsinghua-Tencent 100K [38] is a large benchmark about traffic signs with severe illuminance changes caused by weathers and complex backgrounds. It provides a traffic sign dataset in real world where the sizes of target objects are very small compared to the image size (2048×2048). The dataset has 6K train images and 3K test images. It divides the data in terms of size in the same way as MS COCO [26], which is categorized as small ($area \leq 32 \times 32$), medium ($32 \times 32 < area \leq 96 \times 96$) and large ($area > 96 \times 96$) objects. The portions of small, median and large objects are $(42, 50, 8)\%$, respectively. Due to such dominant presence of small objects, Tsinghua-Tencent 100K is one of the best benchmarks to verify the performance of small object detection.

Evaluation measures. Following the protocol of [38], we evaluate for 45 classes that include more than 100 instances among 182 classes. While only recall and accuracy in terms of sizes are reported in [38], we additionally report F1 scores since they can balance the two metrics. The detection is counted as correct if IoU with the groundtruth is greater than or equal to 0.5.

Quantitative results. We compare the performance of our model to the base models with three backbones as previously specified. We set the threshold for the size of small proposals to 32×32; only the proposals whose area is less than the threshold are treated as inputs to the super-resolution model.

Table 1 summarizes the performance on the Tsinghua-Tencent 100K *test* dataset. We resize the input images from 2048 to 1600 to make learning and inference faster as in [23]. The performance improvement by our approach is significant in the order of small (75.2→84.3 in F1 scores with ResNet-101), medium (92.2→94.6) and large objects (92.2→93.2). The large improvement on small objects are consistent for different CNN backbones such as 63.4→71.0 with MobileNet and 74.9→82.2 with ResNet-50.

Model	PASCAL VOC				MS COCO					
	AP-.5	AP-S	AP-M	AP-L	AP-.5:.95	AP-.5	AP-.75	AP-S	AP-M	AP-L
MobileNet [17]	73.2	5.1	39.3	**76.9**	19.3	38.7	16.9	5.4	20.6	**29.2**
+ Ours	**77.0**	**10.1**	**47.2**	**76.9**	**21.9**	**41.0**	**21.0**	**10.9**	**23.8**	29.0
ResNet-50 [16]	77.1	6.8	42.9	81.1	29.5	52.0	29.8	10.2	31.5	**44.7**
+ Ours	**79.1**	**10.5**	**47.9**	**81.4**	**31.2**	**54.2**	**32.4**	**14.3**	**32.4**	**44.7**
ResNet-101 [16]	78.8	5.9	46.2	82.3	32.0	54.7	32.8	11.3	34.3	**48.1**
+ Ours	**80.6**	**11.1**	**48.9**	**82.7**	**34.2**	**57.2**	**36.1**	**16.2**	**35.7**	**48.1**

Table 3: Overall performance on VOC 2007 *test* and COCO 2017 *test-dev* datasets.

One remark is that although we only super-resolve the small proposals, we obtain the performance gain for medium and large objects as well. It may be because the large predictor is fine-tuned without considering small proposals, which is helpful to focus its modeling power on the medium and large objects. Another reason for improvement in the medium subset is that some proposals that eventually fall in the medium subsets are predicted using the small predictor, due to the offsets added to the proposals in the final step. Given the fact that about 14% of the total objects are in between 32 × 32 and 40 × 40, it may be a valid reason that explains the performance gain for the medium subset.

Comparison with the state-of-the-art methods. Table 2 shows that our proposed model achieves new state-of-the-art performance on Tsinghua-Tencent 100K dataset. In these experiments, we train our model using ResNet-101 as a backbone on the images with their original size. Throughout all the subsets, ours outperform all the previous state-of-the-art models especially in terms of F1 scores.

5.2. Results on PASCAL VOC and MS COCO

We also evaluate our model on PASCAL VOC [6] and MS COCO [26], although the ratio of small objects in these benchmarks are much less than Tsinghua-Tencent 100K. PASCAL VOC consists of 20 object categories with 5K *trainval* and 5K *test* images in 2007 and 11K *trainval* images in 2012. We use 2007 *trainval* + 2012 *trainval* for training and 2007 *test* set for test. MS COCO 2017 consists of 80 object categories with 115K *train*, 5K *val* and 20K *test-dev* images. We use the *train* set for training, and the *val* and *test-dev* set for test. We additionally present the results on the *val* set in the supplementary material.

Evaluation measures. For PASCAL VOC, we use the mAP@.5 metric, which is the averaged AP over all classes when the matching IoU threshold with the groundtruth is greater than or equal to 0.5. For MS COCO, we use the mAP@.5:.95, which is the averaged mAP over different matching IoU thresholds from 0.5 to 0.95. We also divide the results on PASCAL VOC into three different categories according to object sizes; small (AP-S), medium (AP-M) and large (AP-L), as with MS COCO. We set the threshold to 96 × 96 for small proposals since the object sizes are much larger than those of Tsinghua-Tencent 100K.

Model	Small	Medium	Large	Overall
Base model	74.9	91.9	91.9	85.7
+ SR (w.o. supervision)	76.8	93.6	**93.3**	87.5
+ SR (Naïve supervision)	74.4	91.8	92.3	85.3
+ SR (Ours)	**82.2**	**94.3**	**93.3**	**89.7**

Table 4: Comparison of F1 scores between super-resolution methods with ResNet-50 on Tsinghua-Tencent 100K.

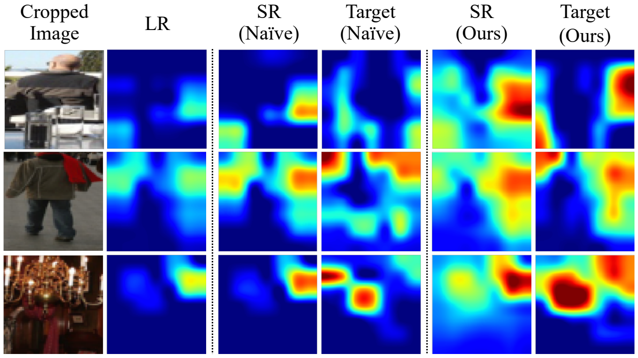


Figure 6: The qualitative results for how RoI features differ between SR with naïve supervision method and ours. The low-resolution features (LR) extracted from the cropped images are super-resolved to be SR (Naïve) and SR (Ours) using SR with naïve supervision and ours, respectively. While SR (Naïve) are not much improved compared to LR, SR (Ours) look very close to Target (Ours).

Quantitative results. Table 3 compares the performance of our model to the baselines on VOC 2007 *test* and COCO 2017 *test-dev*. We observe the similar trend as in Tsinghua-Tencent 100K that the detection enhancement is more significant in the order of small, medium and large objects.

5.3. Comparison of Super-resolution Methods

In this section, we perform an ablation study to analyze different super-resolution methods both quantitatively and qualitatively. We use ResNet-50 as the CNN backbone. We compare our super-resolution approach with two inferior variants; (1) SR without supervision: the model without the content loss ($\mathcal{L}_{cont}$) and (2) SR with naïve supervision: the model trained using the target features from the base feature extractor instead of our SR target extractor.

Table 4 compares F1 scores of different super-resolution

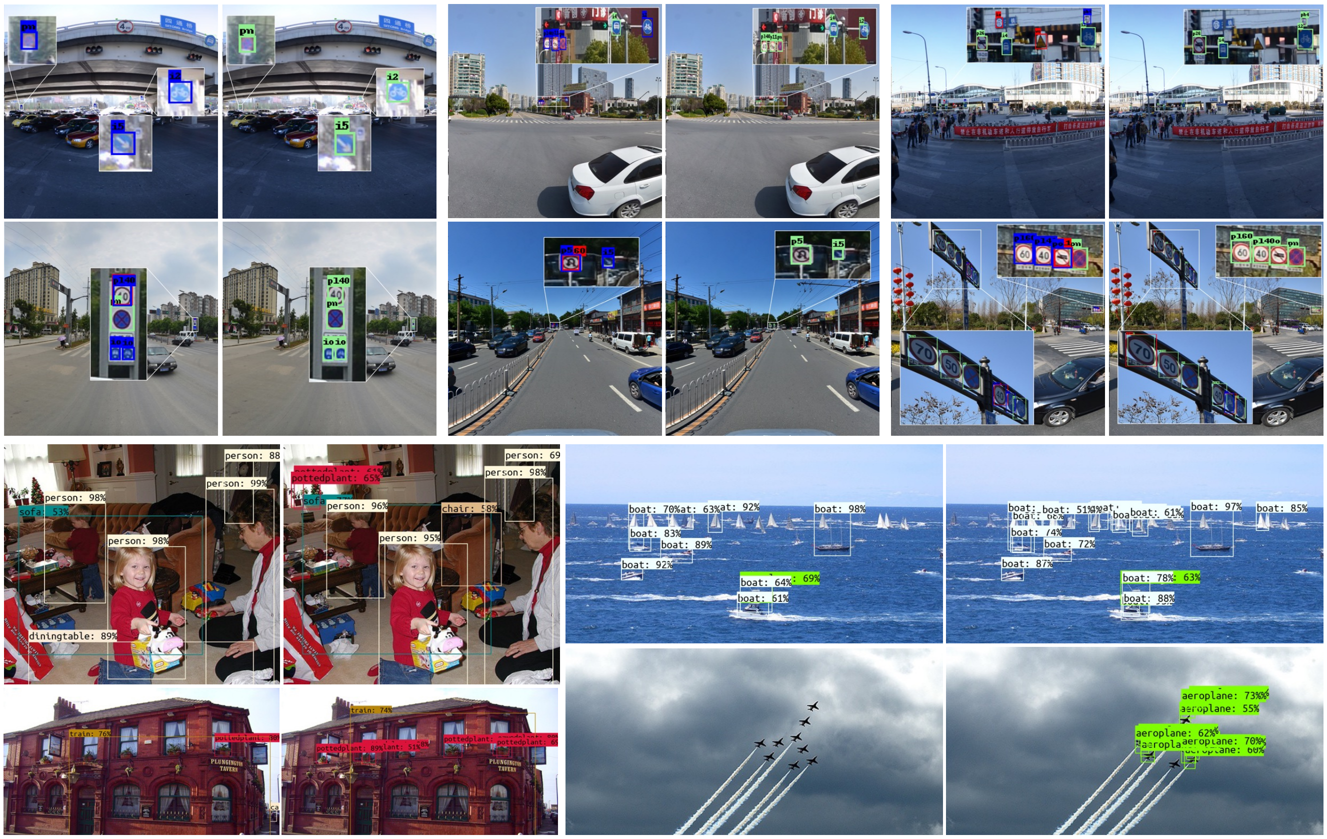


Figure 7: Detection results on Tsinghua-Tencent 100K (the upper half) and PASCAL VOC 2007 (the lower half) datasets. For Tsinghua-Tencent 100K, green, red and blue rectangles represent true positives, false positives and false negatives, respectively. Each pair indicates the results from the base model (left) and our model (right).

models on Tsinghua-Tencent 100K. The other two SR variants obtain only limited performance gains compared to the base model. On the other hand, our SR model achieves significant performance gains, especially for the small subsets. One remark here is SR without supervision performs better than SR with naïve supervision, which implies the improper supervision due to the mismatch of RRF can degrade the performance. Figure 6 qualitatively visualizes the superiority of our model for feature-level super-resolution over SR with naïve supervision method.

5.4. Qualitative Results

Figure 7 illustrates some selected results of detection. For each pair, we show the results of the base detector (left) and our approach (right). Compared to the base model, our approach can detect small objects better with higher confidence. We present more qualitative results including near-miss failure cases in the supplementary file.

6. Conclusion

We proposed a novel feature-level super-resolution approach to improve small object detection for the proposal-based detection framework. Our method is applicable on top of any proposal-based detectors with feature pooling. The experiments on Tsinghua-Tencent 100K, PASCAL VOC and MS COCO benchmarks validated our super-resolution approach was indeed effective to detect small objects. In particular, our work proved that it is important to provide direct supervision using proper high-resolution target features that share the same relative receptive field with the low-resolution input features.

As future work, our model can be enhanced further in a couple of ways. First, we may update the SR feature generator by adopting the state-of-the-art models developed in the image super-resolution task. Second, the super-resolution ratio can be adaptively selected. Although we used only a fixed ratio of 2 in this work, the optimal ratio may depend on the characteristics of RoIs.

Acknowledgements. We appreciate Byeongchang Kim for helpful comments. This work was supported by Samsung Research Funding Center of Samsung Electronics (No. SRFC-TC1603-01) and IITP grant funded by the Korea government (MSIT) (No. 2017-0-01772). Gunhee Kim is the corresponding author.

References

[1] Yancheng Bai, Yongqiang Zhang, Mingli Ding, and Bernard Ghanem. SOD-MTGAN: Small Object Detection via Multi-Task Generative Adversarial Network. In *ECCV*, 2018.

[2] Sean Bell, C Lawrence Zitnick, Kavita Bala, and Ross Girshick. Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks. In *CVPR*, 2016.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. *arXiv:1412.7062*, 2014.

[4] Myung Jin Choi, Antonio Torralba, and Alan S Willsky. A Tree-based Context Model for Object Recognition. *IEEE TPAMI*, 2012.

[5] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In *NIPS*, 2016.

[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, 2010.

[7] Yuan Fang, Kingsley Kuan, Jie Lin, Cheston Tan, and Vijay Chandrasekhar. Object Detection Meets Knowledge Graphs. In *IJCAI*, 2017.

[8] Clinton Fookes, Frank Lin, Vinod Chandran, and Sridha Sridharan. Evaluation of Image Resolution and Super-Resolution on Face Recognition Performance. *Journal of Visual Communication and Image Representation*, 2012.

[9] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. DSSD: Deconvolutional Single Shot Detector. *arXiv:1701.06659*, 2017.

[10] Ross Girshick. Fast R-CNN. In *ICCV*, 2015.

[11] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*, 2014.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *NIPS*, 2014.

[13] Ryuhei Hamaguchi, Aito Fujita, Keisuke Nemoto, Tomoyuki Imaizumi, and Shuhei Hikosaka. Effective Use of Dilated Convolutions for Segmenting Small Object Instances in Remote Sensing Imagery. In *WACV*, 2018.

[14] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Task-Driven Super Resolution: Object Detection in Low-resolution Images. *arXiv:1803.11316*, 2018.

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[17] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv:1704.04861*, 2017.

[18] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation Networks for Object Detection. In *CVPR*, 2018.

[19] Peiyun Hu and Deva Ramanan. Finding Tiny Faces. In *CVPR*, 2017.

[20] Xiaowei Hu, Xuemiao Xu, Yongjie Xiao, Hao Chen, Shengfeng He, Jing Qin, and Pheng-Ann Heng. SINet: A Scale-insensitive Convolutional Neural Network for Fast Vehicle Detection. *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[21] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/Accuracy Trade-offs for Modern Convolutional Object Detectors. In *CVPR*, 2017.

[22] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of Localization Confidence for Accurate Object Detection. In *ECCV*, 2018.

[23] Jianan Li, Xiaodan Liang, Yunchao Wei, Tingfa Xu, Jiashi Feng, and Shuicheng Yan. Perceptual Generative Adversarial Networks for Small Object Detection. In *CVPR*, 2017.

[24] Zhenwen Liang, Jie Shao, Dongyang Zhang, and Lianli Gao. Small Object Detection Using Deep Feature Pyramid Networks. In *Pacific Rim Conference on Multimedia*, 2018.

[25] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *CVPR*, 2017.

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014.

[27] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single Shot MultiBox Detector. In *ECCV*, 2016.

[28] Zibo Meng, Xiaochuan Fan, Xin Chen, Min Chen, and Yan Tong. Detecting Small Signs from Large Images. In *IEEE International Conference on Information Reuse and Integration (IRI)*, 2017.

[29] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *CVPR*, 2016.

[30] Joseph Redmon and Ali Farhadi. YOLO9000: Better, Faster, Stronger. In *CVPR*, 2017.

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NIPS*, 2015.

[32] Mehdi SM Sajjadi, Bernhard Schölkopf, and Michael Hirsch. EnhanceNet: Single Image Super-Resolution through Automated Texture Synthesis. In *ICCV*, 2017.

[33] Abhinav Shrivastava, Rahul Sukthankar, Jitendra Malik, and Abhinav Gupta. Beyond Skip Connections: Top-Down Modulation for Object Detection. *arXiv:1612.06851*, 2016.

[34] Weimin Tan, Bo Yan, and Bahetiyaer Bare. Feature Super-Resolution: Make Machine See More Clearly. In *CVPR*, 2018.

[35] Wei Xiang, Dong-Qing Zhang, Heather Yu, and Vassilis Athitsos. Context-aware Single-Shot Detector. In *WACV*, 2018.

[36] Fisher Yu and Vladlen Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. *arXiv:1511.07122*, 2015.

[37] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid Scene Parsing Network. In *CVPR*, 2017.

[38] Zhe Zhu, Dun Liang, Songhai Zhang, Xiaolei Huang, Baoli Li, and Shimin Hu. Traffic-Sign Detection and Classification in the Wild. In *CVPR*, 2016.