SOD-MTGAN: Small Object Detection via Multi-Task Generative Adversarial Network

Yancheng Bai[1,2⋆], Yongqiang Zhang[1,3⋆], Mingli Ding[3], and Bernard Ghanem[1]

[1] Visual Computing Center, King Abdullah University of Science and Technology.
baiyancheng20@gmail.com, bernard.ghanem@kaust.edu.sa
[2] Institute of Software, Chinese Academy of Sciences (CAS).
[3] School of Electrical Engineering and Automation, Harbin Institute of Technology.
{zhangyongqiang, dingml}@hit.edu.cn

Abstract. Object detection is a fundamental and important problem in computer vision. Although impressive results have been achieved on large/medium sized objects in large-scale detection benchmarks (*e.g.* the COCO dataset), the performance on small objects is far from satisfactory. The reason is that small objects lack sufficient detailed appearance information, which can distinguish them from the background or similar objects. To deal with the small object detection problem, we propose an end-to-end multi-task generative adversarial network (MTGAN). In the MTGAN, the generator is a super-resolution network, which can up-sample small blurred images into fine-scale ones and recover detailed information for more accurate detection. The discriminator is a multi-task network, which describes each super-resolved image patch with a real/fake score, object category scores, and bounding box regression offsets. Furthermore, to make the generator recover more details for easier detection, the classification and regression losses in the discriminator are back-propagated into the generator during training. Extensive experiments on the challenging COCO dataset demonstrate the effectiveness of the proposed method in restoring a clear super-resolved image from a blurred small one, and show that the detection performance, especially for small sized objects, improves over state-of-the-art methods.

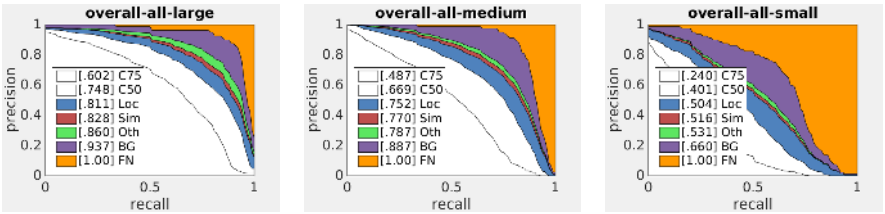
Keywords: Small Object Detection; Super-resolution; Multi-task; Generative Adversarial Network; COCO

1 Introduction

Object detection is a fundamental and important problem in computer vision. It is usually a key step towards many real-world applications, including image retrieval, intelligent surveillance, autonomous driving, etc. Object detection has been extensively studied over the past few decades and huge progress has been made with the emergence of deep convolutional neural networks. Currently, there are two main frameworks for CNN-based object detection: (i) the one-stage

⋆ Equal contribution.

framework, such as YOLO [27] and SSD [24], which applies an object classifier and regressor in a dense manner without objectness pruning; and (ii) the two-stage framework, such as Faster-RCNN [29], RFCN [3] and FPN [22], which extracts object proposals followed by per-proposal classification and regression.



(a) Large (b) Medium (c) Small

Fig. 1. The overall error analysis of the performance of the FPN detector [22] over all categories on the large, medium, and small subsets of the COCO dataset [23], respectively. The plots in each sub-image are a series of precision-recall curves under different evaluation settings defined in [23]. From the comparisons, we can see that there is a large gap between the performance of small and large/medium sized objects.

Object detectors of both frameworks have achieved impressive results on objects of large/medium size in large-scale detection benchmarks (*e.g.* the COCO dataset [23]) as shown in Figure 1(a) and 1(b). However, the performance on small sized objects (defined as in [23]) is far from satisfactory as shown in Figure 1(c). From the comparisons, we can see that there is a large gap between the performance of small and large/medium sized objects. The main difficulty for small object detection (SOD) is that small objects lack appearance information needed to distinguish them from background (or similar categories) and to achieve better localization. To achieve better detection performance on these small objects, SSD [24] exploits the intermediate *conv* feature maps to represent small objects. However, the shallow fine-grained *conv* feature maps are less discriminative, which leads to many false positive results. On the other hand, FPN [22] uses the feature pyramid to represent objects at different scales, in which low-resolution feature maps with strong semantic information are up-sampled and fused with the high-resolution feature maps with weak semantic information. However, up-sampling might generate artifacts, which can degrade detection performance.

To deal with the SOD problem, we propose a unified end-to-end convolutional neural network based on the classical generative adversarial network (GAN) framework, which can be incorporated into any existing detector. Following the structure of the seminal GAN work [9, 21], there are two sub-networks in our model: a generator network and a discriminator network. In the generator, a super-resolution network (SRN) is introduced to up-sample a small object image to a larger scale. Compared to directly resizing the image with bilinear interpolation, SRN can generate images of higher quality and less artifacts at

large up-scaling factors (4× in our current implementation). In the discriminator, we introduce the classification and regression branches for the task of object detection. The real and generated super-resolved images pass through the discriminator network that *jointly* distinguishes whether they are real or generated high-resolution images, determines which classes they belong to, and refines the predicted bounding boxes. More importantly, the classification and regression losses are further back-propagated to the generator, which encourages the generator to produce higher quality images for easier classification and better localization.

Contributions. This paper makes the following three main contributions. **(1)** A novel unified end-to-end multi-task generative adversarial network (MTGAN) for small object detection is proposed, which can be incorporated with any existing detector. **(2)** In the MTGAN, the generator network produces super-resolved images and the multi-task discriminator network is introduced to distinguish the real high-resolution images from fake ones, predict object categories, and refine bounding boxes, simultaneously. More importantly, the classification and regression losses are back-propagated to further guide the generator network to produce super-resolved images for easier classification and better localization. **(3)** Finally, we demonstrate the effectiveness of MTGAN within the object detection pipeline, where detection performance improves a lot over several state-of-the-art baseline detectors, primarily for small objects.

2 Related Work

2.1 General Object Detection

As a classic topic, numerous object detection systems have been proposed during the past decade or so. Traditional object detection methods are based on handcrafted features and the deformable part model (DPM). Due to the limited representation of handcrafted features, traditional object detectors register subpar performance, particularly on small sized objects.

In recent years, superior performance in image classification and scene recognition has been achieved with the resurgence of deep neural networks including CNNs [19, 32, 34]. Similarly, the performance of object detection has been significantly boosted due to richer appearance and spatial representations, which are learned by CNNs [7] from large scale image datasets. Currently, a CNN-based object detector can be simply categorized as belonging to one of two frameworks: the two stage framework and the one stage framework. The region-based CNN (RCNN) [7] can be considered as a milestone of the two stage framework for object detection and it has achieved state-of-the-art detection performance. Each region proposal is processed separately in RCNN [7], which is very time-consuming. After that, ROI-Pooling is introduced in Fast-RCNN [6], which can share the computation between the proposal extraction and classification steps, thus improving the efficiency greatly. By learning both these stages end-to-end, Faster RCNN [29] has registered further improvement in both detection performance and computational efficiency. However, all detectors of this framework

show unsatisfactory performance on small objects in the COCO benchmark, since they do not have any explicit strategy to deal with such objects. To detect small objects better, FPN [22] combines the low-resolution, semantically strong features with high-resolution, semantically weak features via a top-down pathway and lateral connections, in which the learned *conv* feature maps are expected to contain strong semantic information for small objects. Because of this, FPN shows superior performance over Faster RCNN for the task of detecting small objects. However, the low-resolution feature maps in FPN are up-sampled to create the feature pyramid, a process which tends to introduce artifacts into the features and consequently degrades detection performance. Compared to FPN, our proposed method employs the super-resolution network to generate images with high-resolution ($4\times$ up-scaling) from images with low-resolution, thus, avoiding the artifact problem caused by the up-sampling operator in FPN.

In the one stage framework, the detector directly classifies anchors into specific classes and regresses bounding boxes in a dense manner. For example, in SSD [24] (a typical one-stage detector), the low-level intermediate *conv* feature maps of high-resolution are used to detect small objects. However, these *conv* features usually only capture basic visual patterns void of strong semantic information, which may lead to many false positive results. Compared to SSD-like detectors, our discriminator uses deep strong semantic features to better represent small objects, thus, reducing the false positive rate.

2.2 Generative Adversarial Networks

In the seminal work [9], the generative adversarial network (GAN) is introduced to generate realistic-looking images from random noise inputs. GANs have achieved impressive results in image generation [4], image editing [35], representation learning [25], image super-resolution [21] and style transfer [16]. Recently, GANs have been successfully applied to super-resolution (SRGAN) [21], leading to impressive and promising results. Compared to super-resolution on natural images, images of specific objects in the COCO benchmark for example are full of diversity (*e.g.* blur, pose and illumination), thus, making the super-resolution process on these images much more challenging. In fact, the super-resolution images generated by SRGAN are blurred especially for low-resolution small objects, which is not helpful to train an accurate object classifier. To alleviate this problem, we introduce novel losses into the loss function of the generator, *i.e.* the classification and regression losses are back-propagated to the generator network in our proposed MTGAN, which further guides the generator to reconstruct finer super-resolved images for easier classification and better localization.

3 MTGAN for Small Object Detection

In this section, we introduce the proposed method in detail. First, we give a brief description of the classical GAN network to lay the context for describing our proposed Multi-Task GAN (MTGAN) for small object detection. Then, the

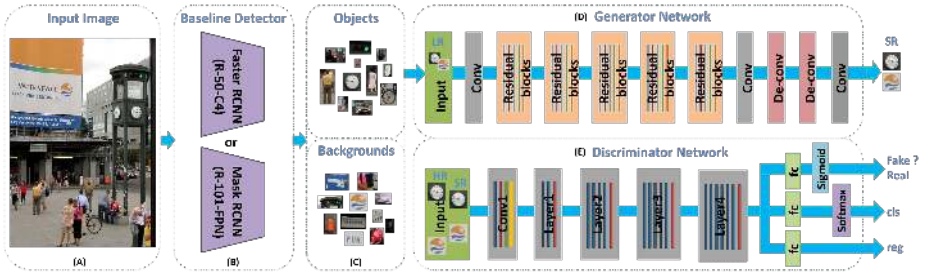


Fig. 2. The pipeline of the proposed small object detection system (SOD-MTGAN). (A) The images are fed into the network. (B) The baseline detector can be any type of detector (*e.g.* Faster RCNN [29], FPN [22], or SSD [24]). It is used to crop positive (*i.e.* objects) and negative (*i.e.* background) examples from input images for training the generator and discriminator networks, or generate regions of interest (ROIs) for testing. (C) The positive and negative examples (or ROIs) are generated by off-the-shelf detectors. (D) The generator sub-network reconstructs a super-resolved version (4× upscaling) of the low-resolution input image. (D) The discriminator network distinguishes the real from the generated high-resolution images, predicts the object categories, and regresses the object locations, simultaneously. The discriminator network can use any typical architecture like AlexNet [20], VGGNet [32] or ResNet [12] as the backbone network. We use ResNet-50 or ResNet-101 in our experiments.

whole architecture of our framework is described (refer to Figure 2 for an illustration). Finally, we present each part of our MTGAN network and define the loss functions for training the generator and discriminator, respectively.

3.1 GAN

GAN [9] learns a generator network G and a discriminator network D simultaneously via an adversarial process. The training process alternately optimizes the generator and discriminator, which are in competition with each other. The generator G is trained to produce samples to fool the discriminator D, and D is trained to distinguish real from fake images produced by G. The GAN loss to be optimized is defined as follows:

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}_{x \sim p_{data}(x)}[\log D_\theta(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D_\theta(G_\omega(z)))], \quad (1)$$

where z is random noise, x denotes the real data, θ and ω denote the parameters of D and G respectively. Here, G tries to minimize the objective function, while D tries to maximize it as Eq (2):

$$\arg\min_G \max_D \mathcal{L}_{GAN}(G, D) \quad (2)$$

Similar to [9, 21], we design a generator network G_w, which is optimized in an alternating manner with discriminator network D_θ seeking to jointly solve the

Table 1. The architecture of the generator and discriminator network. "conv" and "layer*" represent the convolutional layer, "x5" denotes a residual block which has 5 convolutional layers, "de-conv" means a up-sampling convolutional layer, "2x" denotes up-sampling by a factor of 2, and "fc" indicates a fully connected layer. Note that we only post the architecture of the discriminator network with ResNet-50.

	Generator						Discriminator (ResNet-50)					
Layer	conv	conv x5	conv	de-conv	de-conv	conv	conv	layer1	layer2	layer3	layer4	fc
Kernel Num.	64	64	64	256	256	3	64	128	256	512	1024	3
Kernel Size	9	3	3	3	3	9	3	1	1	1	1	-
Stride	1	1	1	2x	2x	1	2	1	2	2	2	-

super-resolution, object classification, and bounding box regression problems for small object detection. Therefore, the overall loss is defined as follows:

$$\begin{aligned} \arg\min_{w} \max_{\theta} \quad & \mathbb{E}_{(I^{HR},u,v)\sim p_{train}(I^{HR},u,v)}[\log D_\theta(I^{HR},u,v)]+ \\ & \mathbb{E}_{(I^{LR},u,v)\sim p_G(I^{LR},u,v)}[\log(1-D_\theta(G_w(I^{LR}),u,v))], \end{aligned} \tag{3}$$

where I^{LR} and I^{HR} denote low-resolution and high-resolution images, respectively. u is the class label and v is the ground-truth bounding-box regression target. Unlike [9], the input of our generator is a low-resolution image rather than random noise. Compared to [21], we have multiple tasks in the discriminator, where we distinguish the generated super-resolved images *vs.* real high-resolution images, classify the object category, and regress the object location jointly. Specifically, the general idea behind Eq (3) is that it allows one to train a generator G with the goal of fooling a differentiable discriminator D that is trained to distinguish super-resolved images from real high-resolution images. Furthermore, our method (SOD-MTGAN) extends classical SRGAN [21] by adding two more parallel branches to classify the categories and regress the bounding boxes of candidate ROI images. Moreover, the classification loss and regression loss in the discriminator are back-propagated to the generator to further promote it to produce super-resolved images that are also suitable for easier classification and better localization. In the following subsection, we introduce the architecture of the MTGAN and the training losses in detail.

3.2 Network Architecture

Our generator takes low-resolution images as input, instead of random noise, and outputs super-resolved images. For the purpose of object detection, the discriminator is designed to distinguish generated super-resolved images from real high-resolution images, classify the object categories, and regress the location jointly.

Generator Network (G_w). As shown in Table 1 and Figure 2, we adopt a deep CNN architecture which has shown effectiveness for image de-blurring in [13] and face detection in [1]. Different from [13], our generator includes up-sampling layers (*i.e.* de-conv in Table 1). There are two up-sampling fractionally-strided *conv* layers, three conv layers, and five residual blocks in the network.

Particularly, in these residual blocks, we use two *conv* layers with 3x3 kernels and 64 feature maps followed by batch-normalization layers [15] and parametric ReLU [11] as the activation function. Each de-convolutional layer consists of learned kernels, which up-samples a low-resolution image to a 2× super-resolved image, which is usually better than re-sizing the same image by an interpolation method [5, 17, 33].

Our generator first up-samples low-resolution small images, which include both object and background candidate ROI images, to 4× super-resolved images via the de-convolutional layers, and then performs convolution to produce corresponding clear images. The outputs of the generator (clear super-resolved images) are easier for the discriminator to classify as fake or real and to perform object detection (*i.e.* object classification and bounding-box regression).

Discriminator Network (D_θ). We employ ResNet-50 or ResNet-101 [12] as our backbone network in the discriminator, and Table 1 shows the architecture of the ResNet-50 network. We add three parallel fc layers behind the last average pooling layer of the backbone network, which play the role of distinguishing the real high-resolution images from the generated super-resolved images, classifying object categories, and regressing bounding boxes, respectively. For this specific task, the first fc layer (called fc_{GAN}) uses a sigmoid loss function [26], while the classification fc layer (called fc_{cls}) and regression fc layer (called fc_{reg}) use the softmax and smooth $L1$ loss [6] functions, respectively.

The input of the discriminator is a high-resolution ROI image, and the output of the fc_{GAN} branch is the probability (p_{GAN}) of the input image being a real image, the output of fc_{cls} branch is the probability ($p_{cls} = (p_0, ..., p_K)$) of the input image being each of $K+1$ object categories, and the output of fc_{reg} branch is the bounding-box regression offsets ($t = (t_x, t_y, t_w, t_h)$) for the ROI candidate.

3.3 Overall Loss Function

We adopt the pixel-wise and adversarial losses from some state-of-the-art GAN approaches [21, 16] to optimize our generator. In contrast to [21], we remove the feature matching loss to decrease the computational complexity without sacrificing much in generation performance. Furthermore, we introduce the classification and regression losses into the generator objective function to drive the generator network to recover fine details from small scale images for easier detection.

Pixel-wise Loss. The input of our generator network is small ROI images instead of random noise [9]. A natural and simple way to enforce the output of the generator (*i.e.* the super-resolved images) to be close to the ground-truth images is by minimizing the pixel-wise MSE loss, and it is computed as Eq (4):

$$L_{MSE}(w) = \frac{1}{N}\sum_{i=1}^{N}\|G_w(I_i^{LR}) - I_i^{HR}\|^2, \tag{4}$$

where I_i^{LR}, $G_w(I_i^{LR})$ and I_i^{HR} denote small low-resolution images, generated super-resolved images, and real high-resolution images, respectively. G represents the generator network, and w denotes its parameters. However, it is known

that the solution to the MSE optimization problem usually lacks high-frequency content, which results in blurred images with overly smooth texture.

Adversarial Loss. To achieve more realistic results, we introduce the adversarial loss [21] to the objective loss, defined as Eq(5):

$$L_{adv} = \frac{1}{N}\sum_{i=1}^{N}\log(1 - D_\theta(G_w(I_i^{LR}))) \tag{5}$$

The adversarial loss encourages the network to generate sharper high-frequency details so as to fool the discriminator D. In Eq (5), $D_\theta(G_w(I_i^{LR}))$ denotes the probability of the resolved image $G_w(I_i^{LR})$ being a real high-resolution image.

Classification Loss. In order to complete the task of object detection and to make the generated images easier to classify, we introduce the classification loss to the overall objective. Let $\{I_i^{LR}, i = 1, 2, \ldots, N\}$ and $\{I_i^{HR}, i = 1, 2, \ldots, N\}$ denote low-resolution images and real high-resolution images respectively, and $\{u_i, i = 1, 2, \ldots, N\}$ represent their corresponding labels, where $u_i \in \{0, ..., K\}$ indicates the object category. As such, we formulate the classification loss as:

$$L_{cls}(p, u) = \frac{1}{N}\sum_{i=1}^{N} -(\log(D_{cls}(G_w(I_i^{LR}))) + \log(D_{cls}(I_i^{HR}))) \tag{6}$$

where $p_{I_i^{LR}} = D_{cls}(G_w(I_i^{LR}))$ and $p_{I_i^{HR}} = D_{cls}(I_i^{HR}))$ denote the probabilities of the generated super-resolved image and the real high-resolution image belonging to the true category u_i, respectively.

In our method, our classification loss plays two roles. First, it guides the discriminator to learn a classifier that classifies high-resolution images, albeit generated super-resolved and real high-resolution images, as real or fake. Second, it promotes the generator to recovery sharper images for easier classification.

Regression Loss. To enable more accurate localization, we also introduce a bounding box regression loss [6] to the objective function, defined in Eq (7):

$$L_{reg}(t, v) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j\in\{x,y,w,h\}}[u_i \geq 1](\mathrm{S}_{L_1}(t_{i,j}^{HR} - v_{i,j}) + \mathrm{S}_{L_1}(t_{i,j}^{SR} - v_{i,j})) \tag{7}$$

in which,

$$\mathrm{S}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{8}$$

where $v_i = (v_{i,x}, v_{i,y}, v_{i,w}, v_{i,h})$ denotes a tuple of the true bounding-box regression target, and $t_i = (t_{i,x}, t_{i,y}, t_{i,w}, t_{i,h})$ denotes the predicted regression tuple. t_i^{HR} and t_i^{SR} denote the tuples for the i-th real high-resolution and generated super-resolved images, respectively. The bracket indicator function $[u_i \geq 1]$ equals to 1 when $u_i \geq 1$ and 0 otherwise. For a more detailed description of the regression loss, we refer the reader to [6].

Similar to the classification loss, our regression loss also has two purposes. First, it encourages the discriminator to regress the location of the object candidates cropped from the baseline detector. Second, it promotes the generator to produce super-resolved images with fine details for more accurate localization.

Objective Function. Based on the above analysis, we combine the adversarial loss in Eq (5), classification loss in Eq (6) and regression loss in Eq (7) with the pixel-wise MSE loss in Eq (4). As such, our GAN network can be trained by optimizing the objective function in Eq (9):

$$\begin{aligned}\max_{\theta}\min_{w}\quad & \frac{1}{N}\sum_{i=1}^{N}\alpha(\log(1-D_\theta(G_w(I_i^{LR})))+\log D_\theta(I_i^{HR}))+\\ & \frac{1}{N}\sum_{i=1}^{N}-\beta(\log(D_{cls}(G_w(I_i^{LR})))+\log(D_{cls}(I_i^{HR})))+\\ & \frac{1}{N}\sum_{i=1}^{N}\gamma\sum_{j\in\{x,y,w,h\}}[u_i\geq 1](\mathrm{S}_{L_1}(t_{i,j}^{HR}-v_{i,j})+\mathrm{S}_{L_1}(t_{i,j}^{SR}-v_{i,j}))+\\ & \frac{1}{N}\sum_{i=1}^{N}\|G_w(I_i^{LR})-I_i^{HR}\|^2\end{aligned} \tag{9}$$

where α, β, and γ are weights trading off the different terms. These weights are cross-validated in our experiments.

Directly optimizing Eq (9) with respect to w for updating generator G makes w diverge to infinity rapidly, since large w always makes the objective attain a large loss. For better behavior, we optimize the objective function in a fixed point optimization manner, as done in previous GAN work [21, 16]. Specifically, we optimize for the parameter w of generator G while keeping the discriminator D fixed and then update its parameter θ keeping the generator fixed. Below are the resulting two sub-problems that are iteratively optimized as:

$$\begin{aligned}\min_{w}\quad & \frac{1}{N}\sum_{i=1}^{N}(\alpha\log(1-D_\theta(G_w(I_i^{LR})))-\beta\log(D_{cls}(G_w(I_i^{LR}))))+\\ & \frac{1}{N}\sum_{i=1}^{N}\gamma\sum_{j\in\{x,y,w,h\}}[u_i\geq 1]\mathrm{S}_{L_1}(t_{i,j}^{SR}-v_{i,j})+\frac{1}{N}\sum_{i=1}^{N}\|G_{w1}(I_i^{LR})-I_i^{HR}\|^2\end{aligned} \tag{10}$$

and

$$\begin{aligned}\min_{\theta}\quad & \frac{1}{N}\sum_{i=1}^{N}-\alpha(\log(1-D_\theta(G_w(I_i^{LR})))+\log D_\theta(I_i^{HR}))+\\ & \frac{1}{N}\sum_{i=1}^{N}-\beta(\log(D_{cls}(G_w(I_i^{LR})))+\log(D_{cls}(I_i^{HR})))+\\ & \frac{1}{N}\sum_{i=1}^{N}\gamma\sum_{j\in\{x,y,w,h\}}[u_i\geq 1](\mathrm{S}_{L_1}(t_{i,j}^{HR}-v_{i,j})+\mathrm{S}_{L_1}(t_{i,j}^{SR}-v_{i,j}))\end{aligned} \tag{11}$$

The loss function of generator G in Eq(10) consists of adversarial loss Eq(5), MSE loss Eq(4), classification loss Eq(6) and regression loss Eq(7), which enforce that the reconstructed images be similar to real, object specific, and localizable high-resolution images with high-frequency details. Compared to the previous GANs, we add the classification and regression losses of generated super-resolved

object images to the generator loss. By introducing these two losses, the super-resolved images recovered from the generator network are more realistic than those optimized by only using the adversarial and MSE losses.

The loss function of discriminator D in Eq (11) introduces the classification loss Eq (6) and the regression loss Eq (7). The function of classification loss is to classify the categories of the real high-resolution and generated super-resolved images, which is parallel to the basic formulation of GAN [9] to distinguish real or generated high-resolution images. In the field of small object detection, as we all know, a few pixel drift may make the predicted bounding-boxes fail to fulfill the evaluation criteria. Therefore, we introduce the regression loss (regression branch) into the discriminator network for better localization.

4 Experiments

In this section, we validate our proposed SOD-MTGAN detector on a challenging public object detection benchmark (*i.e.* COCO dataset [23]), where includes some ablation studies and comparisons against other state-of-the-art detectors.

4.1 Training and Validation Datasets

We use the COCO dataset [23] for all experiments. As stated in [23], there are more small objects than large/medium objects in the dataset, approximately 41% of objects are small ($area < 32^2$). Therefore, we use this dataset for training and validating the proposed method. For the object detection task, there are 125K images taken in natural settings and of everyday life (*i.e.* objects with much diversity). 80K/40K/5K of the data is randomly selected for training, validation, and testing, respectively. Following previous works [2, 22], we use the union of 80k training images and a subset of 35k validation images ($trainval135k$) for training, and report ablation results on the remaining 5k validation images ($minival$).

During evaluation, the COCO dataset is divided into three subsets (small, medium, and large) based on the areas of objects. The medium and large subsets contain objects with an area larger than 32^2 and 96^2 pixels, respectively, while the small subset contains objects with an area less than 32^2 pixels. In this paper, we focus on small object detection using our proposed MTGAN network. We report the final detection performance using the standard COCO metrics, which include AP (averaged over all IoU thresholds, *i.e.* [0.5:0.05:0.95]), AP_{50}, AP_{75} and AP_S, AP_M, AP_L (AP at different scales).

4.2 Implementation Details

In the generator network, we set the trade-off weights $\alpha = 0.001$, $\beta = \gamma = 0.01$. The generator network is trained from scratch and the weights in each layer are initialized with a zero-mean Gaussian distribution with standard deviation 0.02, and the biases are initialized with 0. To avoid undesirable local optima, we first train an MSE-based SR network to initialize the generator network. For the

Table 2. The detection performance (AP) of our proposed method SOD-MTGAN against the baseline methods on the COCO *minival* subset. The AP performance of Faster RCNN [29] and Mask-RCNN [10] are provided by [8]. Obviously, SOD-MTGAN outperforms the baseline methods, especially on the small subset where the AP performance increases more than 1.5%.

Methods	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster-RCNN (Baseline)	ResNet-50-C4	36.5	57.3	39.3	18.4	40.6	50.6
SOD-MTGAN (Ours)	ResNet-50	**37.2**	**57.7**	**40.2**	**19.9**	**41.1**	**51.2**
Mask-RCNN (Baseline)	ResNet-101-FPN	40.9	61.9	44.8	23.5	44.2	53.9
SOD-MTGAN (Ours)	ResNet-101	**41.5**	**62.5**	**45.4**	**25.1**	**44.6**	**54.1**

discriminator network, we employ the ResNet-50 or ResNet-101 [12] model pre-trained on ImageNet as our backbone network and add three parallel *fc* layers as described in Section 3.2. The *fc* layers are initialized by a zero-mean Gaussian distribution with standard deviation 0.1, and biases initialized with 0.

Our baseline detectors are based on Faster RCNN with ResNet50-C4 [12] and FPN with ResNet101 [22]. All hyper-parameters of the baseline detectors are adopted from the setup in [10]. For training our generator and discriminator networks, we crop positive and negative ROI examples from COCO [23] *trainval*135*k* set with our baseline detectors. The corresponding low-resolution images are generated by down-sampling the high-resolution images using bicubic interpolation with a factor 4. During testing, 100 ROIs are cropped by our baseline detector and then fed to our MTGAN network to produce final detection.

During training, we use the Adam optimizer [18] for the generator and the SGD optimizer for the discriminator network. The learning rate for SGD is initially set to 0.01 and then reduced by a factor of 10 after every $40k$ mini-batches. Training is terminated after a maximum of $80k$ iterations. We alternately update the generator and discriminator network as in [9]. Our system is implemented in PyTorch, and the source code will be made publicly available.

4.3 Ablation Studies

We first compare our proposed method with the baseline detectors to prove the effectiveness of the MTGAN for small object detection. Moreover, we verify the positive influence of the regression branch in the discriminator network by comparing the AP performance with/without this branch. Finally, to validate the contribution of each loss (adversarial, classification, and regression) in the loss function of the generator, we also conduct ablation studies by gradually adding each of them to the pixel-wise MSE loss. Unless otherwise stated, all the ablation studies use the ResNet-50 as the backbone network in the discriminator.

Influence of the Multi-task GAN (MTGAN). Table 2 (the 2^{nd} *vs.* 3^{rd} row and the 4^{th} *vs.* 5^{th} row) compares the performance of the baseline detectors against our method on the COCO *minival* subset. From Table 2, we observe that the performance of our MTGAN with ResNet-50 outperforms Faster-RCNN

Fig. 3. Some examples of super-resolved images generated by our MTGAN network from small low-resolution patches. The first column of each image group depicts the original low-resolution image, which is upsampled 4× for visualization. The second column is the ground truth high-resolution image, while the third column is the corresponding super-resolved image generated by our generator network.

(the ResNet-50-C4 detector) by a sizable margin (*i.e.* 1.5% in AP) on the small subset. Similarly, MTGAN with ResNet-101 improves over the FPN detector with ResNet-101 by 1.6% in AP. The reason is that the baseline detectors perform the down-sampling operations (*i.e.* convolution with stride 2) when extracting *conv* feature maps. The small objects themselves contain limited information, and the majority of the detailed information will be lost after down-sampling. For example, if the input is a 16×16 pixel object ROI, the result is a 1×1 C4 feature map and nothing is preserved for the C5 feature map. These limited *conv* feature maps degrade the detection performance for such small objects. In contrast, our method up-samples the low-resolution image to a fine scale, thus, recovering the detailed information and making detection possible. Figure 3 shows some super-resolved images generated by our MTGAN generator.

Influence of the regression branch. As shown in Figure 1, imperfect localization is one of the main sources of detection error. This especially the case for small sized objects, where small shifts in their bounding boxes lead to failed detections when the standard strict evaluation criteria are used. The regression branch in the discriminator can further refine bounding boxes and lead to more accurate localization. From Table 3 (1^{st} and 5^{th} row), we see that the AP performance on the small object subset improves by 0.9% when the regression branch is added, thus, demonstrating its effectiveness on the detection pipeline.

Influence of the adversarial loss. Table 3 (the 2^{nd} and 5^{th} row) shows that the AP on the small subset drops by 0.5% without the adversarial loss. The reason is that the generated images without adversarial loss are over-smooth and lack high frequency information, which is important for object detection. To encourage the generator to produce high-quality images for better detection, we use the adversarial loss to train our generator network.

Influence of the classification loss. From Table 3 (the 3^{rd} and 5^{th} row), we see that the AP performance increases by about 1% on the small subset when

Table 3. Performance of our SOD-MTGAN model trained with and without the regression branch, adversarial loss, classification loss, and regression loss on the COCO *minival* subset. "reg+" indicates the regression branch in the discriminator, "adv" denotes the adversarial loss in Eq (5), "cls" represents the classification loss in Eq (6), and "reg" indicates the regression loss in Eq (7).

Methods	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
w/o reg+ branch	36.7	57.5	39.8	19.0	40.9	49.9
w/o adv loss	37.0	57.6	40.0	19.4	41.0	51.0
w/o cls loss	36.8	57.6	39.9	19.2	41.1	50.3
w/o reg loss	36.7	57.6	39.7	19.1	41.1	50.2
SOD-MTGAN (Ours)	**37.2**	**57.7**	**40.2**	**19.9**	**41.2**	**51.2**

the classification loss is incorporated. Clearly, this validates the claim that the classification loss promotes the generator to recover finer detailed information for better classification. In doing so, the discriminator can exploit the fine details to predict the correct category of the ROI images.

Influence of the regression loss. As shown in Table 3 (the 4^{th} and 5^{th} row), the AP performance increases by nearly 1% on the small subset by using the regression loss to train the generator network. Similar to the classification loss, the regression loss drives the generator to recover some fine details for better localization. The increased AP demonstrates the necessity of the regression loss in the generator loss function.

4.4 State-of-the-Art Comparison

We compare our proposed method (SOD-MTGAN) with several state-of-the-art object detectors [24, 28, 12, 22, 14, 31, 10] on the COCO $test-dev$ subset. Table 4 lists the performance of every detector, from which we conclude that our method surpasses all other state-of-the-art methods on all subsets. More importantly, our SOD-MTGAN achieves the highest performance (24.7%) on the small subset, outperforming the second best object detector by about 3%. This AP improvement is most notable for the small object subset, which clearly demonstrates the effectiveness of our method on small object detection.

4.5 Qualitative Results

Figure 4 shows some detection results generated by the proposed SOD-MTGAN detector. We observe that our method successfully finds almost all the objects, even though some ones are very small. This demonstrates the effectiveness of our detector on the small object detection problem. Figure 4 shows some failure cases including some false negative and positive results, which indicate that there is still room for progress in further improving small object detection performance.

Table 4. The performance (AP) of the proposed SOD-MTGAN detector and other state-of-the-art methods on COCO $test-dev$ subset. "+++" denotes the more complex training/test stages, which includes multi-scale train/test, horizontal flip train/test and OHEM [30] in the Faster RCNN.

Methods	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
SSD512 [24]	VGG16	26.8	46.5	27.8	9.0	28.9	41.9
YOLO9000 [28]	Darknet-19	21.6	44.0	19.2	5.0	22.4	35.5
Faster RCNN+++ [12]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
FPN [22]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
G-RMI [14]	Inception-ResNet-v2	34.7	55.5	36.7	13.5	38.1	52.0
TDM [31]	Inception-ResNet-v2-TDM	36.8	57.7	39.7	16.2	39.8	52.1
Mask RCNN [10]	ResNeXt-101-FPN	39.8	62.3	43.4	22.1	43.2	51.2
SOD-MTGAN (Ours)	ResNet-101	**41.4**	**63.2**	**45.4**	**24.7**	**44.2**	**52.6**

Fig. 4. Qualitative results of the SOD-MTGAN detector. Green and red boxes denote the ground-truths and the results of our method. Best seen in color and zoomed in.

5 Conclusion

In this paper, we propose an end-to-end multi-task GAN (MTGAN) to detect small objects in unconstrained scenarios. In the MTGAN, the generator upsamples the small blurred ROI images to fine-scale clear images, which are passed through the discriminator for classification and bounding box regression. To recover detailed information for better detection, the classification and regression losses in the discriminator are propagated back to the generator. Extensive experiments on the COCO dataset demonstrate that our detector improves state-of-the-art AP performance in general, where the largest improvement is observed for small sized objects.

Acknowledgments. This work was supported mainly by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research and by Natural Science Foundation of China, Grant No. 61603372.

References

1. Bai, Y., Zhang, Y., Ding, M., Ghanem, B.: Finding tiny faces in the wild with generative adversarial network. In: CVPR (June 2018)
2. Bell, S., Zitnick, C.L., Bala, K., Girshick, R.: Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. CVPR (2016)
3. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In: NIPS. pp. 379–387 (2016)
4. Denton, E.L., Chintala, S., szlam, a., Fergus, R.: Deep generative image models using a laplacian pyramid of adversarial networks. In: Advances in Neural Information Processing Systems 28, pp. 1486–1494. Curran Associates, Inc. (2015), http://papers.nips.cc/paper/5773-deep-generative-image-models-using-a-laplacian-pyramid-of-adversarial-networks.pdf
5. Dong, C., Loy, C.C., Tang, X.: Accelerating the Super-Resolution Convolutional Neural Network, pp. 391–407. Springer International Publishing, Cham (2016)
6. Girshick, R.: Fast r-cnn. In: ICCV. pp. 1440–1448. IEEE (2015)
7. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR. pp. 580–587 (2014)
8. Girshick, R., Radosavovic, I., Gkioxari, G., Dollár, P., He, K.: Detectron. https://github.com/facebookresearch/detectron (2018)
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems 27, pp. 2672–2680. Curran Associates, Inc. (2014), http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf
10. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask r-cnn. In: CVPR. pp. 2961–2969 (2017)
11. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: ICCV. pp. 1026–1034 (2015)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (June 2016)
13. Hradi, M., Kotera, J., Zemk, P., roubek, F.: Convolutional neural networks for direct text deblurring. In: Xianghua Xie, M.W.J., Tam, G.K.L. (eds.) BMVC. pp. 6.1–6.13 (2015)
14. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al.: Speed/accuracy trade-offs for modern convolutional object detectors. In: IEEE CVPR (2017)
15. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML. pp. 448–456 (2015)
16. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR. pp. 1125–1134 (2017)
17. Kim, J., Kwon Lee, J., Mu Lee, K.: Accurate image super-resolution using very deep convolutional networks. In: CVPR (June 2016)
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014)
19. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. pp. 1097–1105 (2012)
20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012), http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

21. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR. pp. 4681–4690 (2017)
22. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR. vol. 1, p. 4 (2017)
23. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. pp. 740–755. Springer (2014)
24. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV. pp. 21–37. Springer (2016)
25. Mathieu, M.F., Zhao, J.J., Zhao, J., Ramesh, A., Sprechmann, P., LeCun, Y.: Disentangling factors of variation in deep representation using adversarial training. In: Advances in Neural Information Processing Systems 29, pp. 5040–5048. Curran Associates, Inc. (2016), http://papers.nips.cc/paper/6051-disentangling-factors-of-variation-in-deep-representation-using-adversarial-training.pdf
26. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. CoRR **abs/1511.06434** (2015)
27. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. pp. 779–788 (2016)
28. Redmon, J., Farhadi, A.: Yolo9000: Better, faster, stronger. In: CVPR. pp. 6517–6525. IEEE (2017)
29. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NIPS. pp. 91–99 (2015)
30. Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: CVPR. pp. 761–769 (2016)
31. Shrivastava, A., Sukthankar, R., Malik, J., Gupta, A.: Beyond skip connections: Top-down modulation for object detection. CoRR **abs/1612.06851** (2016)
32. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)
33. Wang, Z., Liu, D., Yang, J., Han, W., Huang, T.: Deep networks for image super-resolution with sparse prior. In: ICCV (December 2015)
34. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: NIPS. pp. 487–495 (2014)
35. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative Visual Manipulation on the Natural Image Manifold, pp. 597–613. Springer International Publishing, Cham (2016)