Integration of gradient guidance and edge enhancement into super-resolution for small object detection in aerial images

Jinzhen Mu\textsuperscript{1,2,3} \mid Shuang Li\textsuperscript{1} \mid Zongming Liu\textsuperscript{2,3} \mid Yan Zhou\textsuperscript{4}

\textsuperscript{1} College of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing, China
\textsuperscript{2} Shanghai Aerospace Control Technology Institute, Shanghai, China
\textsuperscript{3} Shanghai Key Laboratory of Aerospace Intelligent Control Technology, Shanghai, China
\textsuperscript{4} School of Automation and Electronic Information, Xiangtan University, Xiangtan, China

Abstract
Detecting small objects are difficult because of their poor-quality appearance and small size, and such issues are especially pronounced for aerial images of great importance. To address the small object detection (SOD) problem, a united architecture that tries to upsample small objects into super-resolved versions, achieving characteristics similar to those large objects and thus resulting in more discriminative detection is used. For this purpose, a new end-to-end multi-task generative adversarial network (GAN) is proposed. In the architecture, the generator is a super-resolution (SR) network, and the discriminator is a multi-task network. In the generator, a gradient guide and an edge-enhancement strategy are introduced to alleviate structural distortions. In the discriminator, a faster region-based convolutional neural network (FRCNN) is incorporated for the task of object detection. Specifically, the discriminator outputs a distribution scalar to measure the realness. Then, each super-resolved image passes through the discriminator with a realness distribution, classification scores, and bounding box regression offsets. Furthermore, the losses of the detection task are backpropagated into the generator during training rather than being optimized independently. Extensive experiments on the challenging cars overhead with context dataset (COWC), detection in optical remote sensing images (DIOR), vision meets drones (VisDrone), and dataset for object detection in aerial images (DOTA) demonstrate the effectiveness of the proposed method in reconstructing structures while generating natural super-resolved images and show the superiority of the proposed method in detecting small objects over state-of-the-art detectors.

1 INTRODUCTION

Object detection in aerial images has numerous valuable applications, such as scene interpretation, resource classification, and activity monitoring [1,2]. Traditional object detectors such as the one-stage frameworks (e.g., you only look once (YOLO) [3], fully convolutional one-stage object detector (FCOS) [4], single shot detection (SSD) [5]) and two-stage frameworks (e.g., faster region-based convolutional neural network (FRCNN) [6], CornerNet [7], spatial pyramid pooling network (SPPNet) [8]) indeed work well for medium or large objects. However, they usually fail to detect small objects. These detectors show a significant performance gap between small objects and normal objects in the benchmark. Taking SSD as an example, it achieves 44.1 mAP and 51.2 mAP on objects with medium and large sizes, but only 20.7 mAP on small objects of the Microsoft Common Objects in Context (COCO) [9] test-dev set. The main reason is that small objects have limited pixels. Therefore, small objects tend to lack rich representations for modern detectors to separate them from similar categories. Moreover, a complex background further reduces the detection performance. However, small objects are very common in many real-world applications. Small object detection (SOD) is much more challenging than normal object detection, and good solutions are still rare so far.

To achieve better performance for SOD, researchers have tailored various architectures [10–29]. Common small object solutions include data augmentation [16], multi-scale feature fusion [12–15, 17, 20, 22], tailored sampling strategies [19, 21, 28, 29], generative adversarial networks (GANs) [23–27], and multi-scale training [18]. However, data augmentation and multi-scale feature fusion often result in substantial time
consumption for training and testing, tailored sampling strategies are specific to particular tasks, GANs usually have undesired structural distortions, and multi-scale training is difficult to train. Motivated by structure-preserving SR (SPSR) [20] and small object detection via multi-task generative adversarial network (SOD-MTGAN) [25], we argue that a preferable way to effectively detect small objects is to upscale poor-quality images into super-resolved ones to reconstruct more useful details for more accurate detection.

Therefore, we propose a new GAN-like architecture to detect small objects. In the generator, a super-resolution (SR) network is introduced to upscale a small object image to a larger scale. In the discriminator, an FRCNN is incorporated for the task of object detection. The SR network includes a basic network, a gradient guidance network (GGN), and an edge-enhancement network (EEN). The GGN provides structural prior to guiding the basic network pays attention to particular regions with sharpness and structures. The EEN further reduces the dirty edges of the SR image. Subsequently, the discriminator distinguishes the real/fake images and detects objects simultaneously. Different from standard GANs [30–39, 24–27], where the realism of a super-resolved sample is estimated by a single scalar (such as 0 (fake) or 1 (real)), our discriminator outputs a distribution as the measure of realness. The single scalar could be viewed as an abstract or a summarization of multiple measures, which together reflect the overall realism of a super-resolved image. Compared to the single scalar, we naturally deduce a super-resolved image based on multiple criteria from more than one angle. More importantly, the losses of detection are backpropagated into the generator for an overall optimization such that the SR network will pay more attention to target regions.

In summary, this work makes the following contributions: (1) proposes a novel end-to-end jointly trainable GAN-like model, which offers a multi-tasking paradigm by handling both SR and detection for aerial imagery; (2) introduces two novel strategies to capture clear edge information by constructing different subnetworks and optimizing each branch with special objectives; (3) introduces a new discriminator that outputs a distribution as the measure of realness, which provides more insights for the generator instead of barely differentiating fake and real; (4) achieves state-of-the-art performance on representative aerial image datasets.

The remainder of this paper is organized as follows. Section 2 reviews GAN-based SR, SOD, and simultaneous GAN-based SR with object detection. Section 3 provides details of our proposed framework. Section 4 presents the experimental results. Section 5 conducts ablation studies, and Section 6 concludes our work.

## 2 RELATED WORK

Our primary focus is on how to enhance the detection accuracy of small objects by producing the SR versions of input images. Furthermore, notable related past works will be discussed in this section.

### 2.1 GAN-based SR

Super-resolution generative adversarial network (SRGAN) [30] is a landmark study in perceptual-driven SR that applies a GAN-based framework to traditional SR. Its generator is composed of residual blocks (RBs). Enhanced SRGAN (ESRGAN) [31] introduces a residual-in-residual dense block (RRDB) to replace the original RB. Further improving ESRGAN (ESRGAN+) [32] uses residual-in-residual dense block residual (RRDBR) to further improve the generative capacity of ESRGAN. Generative adversarial network based edge-enhancement network (EEGAN) [33] develops an additional edge-enhancement subnetwork to the GAN-based SR framework to guide the generator. Structure-preserving SR (SPSR) [34] applies gradient maps (GMs) to the GAN-based SR model as structural guidance to reduce geometric distortions. Spatial feature transformation GAN (SFTGAN) [35] utilizes prior category information to solve the unreal problem. SR generative adversarial networks with ranker (RankSRGAN) [36] introduces a rank-content loss to improve the perceptual quality. SR residual convolutional generative adversarial network (SRResCGAN) [37] proposes a downsampling GAN to follow real-world degradation settings. Single natural image generative adversarial network (SinGAN) [38] introduces a pyramid of fully convolutional GANs to solve the joint deblurring and SR task. Progressive perception-oriented network (PPON) [39] adopts a progressive upsampling method to progressively create high-quality SR results.

### 2.2 SOD

Small objects occupy only a small part of a large area. ‘Small’ here has two meanings: small size and low resolution. In general, small objects contain the least information, resulting in the least features that can be extracted from them.

Common small object solutions include data augmentation [16], multi-scale feature fusion [12–15, 17, 20, 22], and multi-scale training [18]. Recently, researchers have proposed several tailored sampling strategies for detecting small objects [19, 21, 28, 29]. Furthermore, many other meaningful works have been proposed to detect small objects in aerial images [21–23]. However, these detectors still do not meet the requirements for SOD.

We show that a GAN-based SR method is an effective means to improve the detection performance for small objects.

### 2.3 Simultaneous GAN-based SR with object detection

Since GAN-based SR is beneficial for SOD, there are some efforts [24–27] to apply it for improving object detection. Edge-enhanced SR GAN with object detector network (EESRGAN) [24] exploits super-resolved images directly to detect objects at low scales. SOD-MTGAN [25] designs a multi-task loss with an SR loss for object proposals. Perceptual GAN [26] proposes a perceptual GAN to narrow the representation difference of small objects from the large ones. Joint SR and vehicle detection
network (Joint-SRVDNet) [27] proposes a multi-scale GAN to create super-resolved versions of the original images. These methods mentioned above improve the performance of SOD to some extent. Compared with these methods, our method has three differences: one is the distribution outputs, the other is the structure-preserving strategies, and the third is the overall optimization.

3 | METHOD

Previous studies usually use a pre-trained SR network to synthesize a super-resolved version of small objects for conducting SOD. Such methods perform one task without utilizing the advantages of the other task. In addition, SR results often contain structural distortions and blurry edges. To address these problems, we propose a new network to simultaneously generate super-resolved aerial images and locate objects in the super-resolved versions. Our proposed framework is composed of a GAN-based SR network and an FRCNN-based detector, as shown in Figure 1. In our architecture, the generator is an SR network, and the discriminator is a multi-task network. The generator includes an SR branch, a gradient branch, and an edge branch. The discriminator includes an FRCNN-based detector and a discriminator branch. More importantly, the discriminator outputs a distribution rather than a single scalar as the measure of realness and the classification and regression losses are backpropagated to further guide the generator network to produce super-resolved images for easier classification and better localization.

3.1 | Distribution GAN

In standard GAN, the learning process is

$$\min_G \max_D V(G, D) = E_{p_D} \left[ \log(D(x)) - 0 \right] + E_{p_G} \left[ \log(1 - D(x)) \right],$$

where $V(G, D)$ is a value function, $p_D$ is the real data distribution, $p_G$ is the generator’s distribution, and 0 and 1 are two single scalars. The single scalar could be viewed as a summarization of multiple measures, which together reflect the overall realness of a sample [40]. Such a measurement may guide the generator from only one perspective.

Here, we utilize a distribution $p_{\text{realness}}$ to replace the single scalar in Equation (1). Given an input sample $x$, we have $D(x) = \{p_{\text{realness}}(x, u); x, u \in \Omega\}$, where $\Omega$ represents the set of outcomes of $p_{\text{realness}}$, $u$ represents a potential realness. Similarly, we introduce $A_1$ (real) and $A_0$ (fake), which are defined on as virtual ground-truth distributions to represent the realness distributions of real and fake images. Accordingly, the difference between the two scalars is replaced with the Kullback-Leibler (KL) divergence. The learning process thus becomes

$$V(D, G) = E_{p_D(x)} [DKL(A_1 || D(G(x)))] + E_{p_G(x)} [DKL(A_0 || D(G(x)))] \text{,}$$

where $DKL(\cdot)$ is the KL divergence.

In our implementation, we define a discrete distribution over $N$ outcomes $\Omega = \{u_0, u_1, ..., u_{N-1}\}$ to characterize the $p_{\text{realness}}$. Given an input sample $x$, the distribution $p_{\text{realness}}$ is obtained as

$$p_{\text{realness}}(x, u) = \frac{\phi(x)}{\sum_j \phi_j},$$

where $\phi = (\psi_0, \psi_1, ..., \psi_{N-1})$ are the parameters of the discriminator. Similarly, $A_1$ (real) and $A_0$ (fake) are obtained on $\Omega$.

3.2 | Gradient branch

Previous SR networks ignore the importance of structural information. In this section, we try to exploit it. Since the gradient map (GM) reveals the sharpness of each object region in an image, we introduce a gradient guidance strategy to obtain
For this purpose, we design a gradient branch to reconstruct SR GM. The SR gradients can be integrated into the SR branch to provide a structure prior to SR. In addition, the gradients can highlight the regions where sharpness and structures should be given more attention to guide the high-quality generation explicitly. We utilize the difference between two diagonal pixels to calculate the input LR image gradient:

\[
I_x(v) = I(x+1,y+1) - I(x-1,y-1)
\]

\[
I_y(v) = I(x-1,y+1) - I(x+1,y-1),
\]

where \(GM(\cdot)\) represents the extraction function to obtain the GMs and \(v = (x,y)\) represents the pixel coordinates in the GM. In practice, the operation to obtain the gradients can be easily achieved by a convolution layer with a fixed kernel [34].

As shown in Figure 2, the gradient branch incorporates several intermediate-level representations from the SR branch. The reason is that an SR branch carries structural information that is pivotal to the recovery of GM. Hence, we utilize the feature maps from the first blocks and the last blocks of the SR branch as a strong prior to accelerate the recovery of SR GM. A connecting block is used to fuse the features. Then, we feed the syncretic feature maps to the gradient block to extract higher level features. Once the SR GMs are obtained, we integrate the features of the GM produced by the next-to-last layer of gradient branch to the SR branch to guide the SR reconstruction by a matrix multiplication operation. Meanwhile, we utilize a 1x1 convolution layer to generate the output GM, as shown in Figure 2. The magnitude of the GM can implicitly reflect whether a recovered region is sharp or smooth.

We formulate the gradient loss by minimizing the distance between the GM extracted from the SR image and the map from the corresponding HR image. With supervision from gradient domains, the generator creates high-quality images without geometric distortions. Subsequently, we use a pixelwise loss to achieve the above goal.

\[
L_{\text{pix}}^{\text{GM}} = E\left\| GM(G(I_{LR})) - GM(I_{HR}) \right\|_1,
\]

where \(G(I_{LR})\) represents a super-resolved image and \(I_{HR}\) represents a ground-truth image.

### 3.3 SR branch

In this sub-section, we use the SR branch to obtain the intermediate SR (ISR) results. The SR branch provides a fundamental SR operation. This branch constitutes two parts. The first part is a regular SR network which is similar to the generator of ESRGAN. Compared to ESRGAN, we replace the RRDB with the proposed residual-in-residual residual dense block (RRRDB). The RRRDB has a residual-in-residual structure with dense blocks in the main path, as presented in Figure 3(a). We add an additional level of residual learning inside the dense blocks, as presented in Figure 3(b), to augment the network capacity without increasing its complexity. Then, RRRDB combines multi-level residual network and dense connections as depicted in Figure 3(c). This new architecture benefits from both feature exploitation and exploration, resulting in
images of superior perceptual quality. Since we use 23 RRRDB blocks in the SR branch, we use the feature from the 5th, 10th, 15th, and 20th blocks into the gradient branch to enhance the GM. The second part of the SR branch fuses the feature of GM. We fuse the structure information by matrix multiplication. Finally, we use two convolutional layers to reconstruct the final ISR features, as shown in Figure 2.

The generator loss consists of a perceptual loss ($L_{\text{percep}}$) and a content loss ($L_{1}$). The content loss to evaluate the 1-norm distance between the super-resolved image $G(I_{LR})$ and the ground-truth image $I_{HR}$ is:

$$L_{1} = E \left\| \frac{G(I_{LR}) - I_{HR}}{1} \right\|,$$  

where $\varphi_i$ denotes the $i$-th layer of the visual geometry group (VGG) model.

3.4 Edge-enhanced branch

The ISR achieves satisfactory results, but the edges are not detailed enough. Edge information is a critical factor to reconstruct the final SR images. The edge branch aims to further enhance the edges of the ISR image. In the beginning, a Laplacian operator [33, 42] is used to extract edges from the ISR. After the edge information is extracted, it is passed through the project block, RRRDBs, and up-sampling blocks. Finally, the enhanced edges are added to the input image, where the edges extracted by the Laplacian operator are subtracted.

To enhance the consistency between ISR and HR, we use two terms of loss, which include a consistency loss for the images ($L_{\text{img_cst}}$) and a consistency loss for the edges ($L_{\text{edge_cst}}$).

The consistency loss for the images ($L_{\text{img_cst}}$) utilizes the Charbonnier penalty function between the ISR and HR to enhance the consistency of image contents, and $L_{\text{edge_cst}}$ uses the Charbonnier penalty function between the SR’s edges and HR’s edges to enhance the consistency of image edges.

$$L_{\text{img_cst}} = \frac{1}{r^2 \cdot w^2} \sum_{x=1}^{rw} \sum_{y=1}^{rh} \left\| \varphi_i(I_{HR})(x, y) - ED(I_{ISR})(x, y) \right\|,$$

$$L_{\text{edge_cst}} = E \left[ \varphi_i(I_{edge_{HR}} - I_{edge_{SR}}) \right].$$
where $\rho(\cdot)$ is the Charbonnier penalty function, $I^{HR}$ is the ground-truth image, $I^{SR}$ is the ISR image, $GM$ is the edge enhancement operation, $r$ is the scaling factor, $w$ and $h$ represent the width and height of the image, $I_{edge,SR}$ is the edge of the ground-truth image, $I_{edge,HR}$ is edge of the SR image, respectively.

### 3.5 Discriminator

Since the discriminator provides valuable feedback information to the generator, we use it to justify whether the SR result is useful for improving SOD accuracy. Taking a super-resolved image as input, the discriminator passes it into two branches, that is, the adversarial branch and the detection branch. Such that the discriminator outputs a realsness distribution, classification scores, and bounding box regression offsets.

The adversarial branch is similar to the discriminator of ESRGAN, which uses the VGG-19 [42] architecture. To achieve more realistic results, we design a new discriminator loss based on distribution GAN that is given as

$$L_{dis,D} = -E[DKL(A_1||D(I^{SR}))]$$

where $A_1$ is a 2D channel and $D$ is the discriminator. We perform all experiments using a 4x upsampling factor between low- and high-resolution images. To obtain LR images, a bicubic kernel is used to downscale the HR images with a scale factor of 4. We train our architecture jointly in an end-to-end fashion. During training, we use the Adam optimizer for the generator and the SGD optimizer for the discriminator network. The learning rate for Adam is initially set to 0.0001 and then reduced by a factor of 10 after every 25k iterations with batch size 16. The learning rate for SGD is initially set to 0.01 and then reduced by a factor of 10 after every 40k iterations with batch size 16. Training is terminated after a maximum of 80k iterations. Our system is implemented in the PyTorch framework and trained over three NVIDIA 1080ti, and the source code will be made publicly available. Moreover, the hyperparameters will be discussed in the ablation study.

### 4 EXPERIMENTAL RESULTS ANALYSIS

#### 4.1 Implementation strategies and training parameters

We use the generator and the discriminator of ESRGAN as our SR branch and adversarial branch, respectively. In our network, we design RRRDB to replace the one used by the original ESRGAN. Here, we use 23 RRRDB blocks for the SR branch, 3 RRRDB blocks for the edge-enhanced branch, and 4 RRRDB blocks for the gradient branch.

We perform all experiments using a 4x factor for low- and high-resolution images. To obtain LR images, a bicubic kernel is used to downscale the HR images with a scale factor of 4. We train our architecture jointly in an end-to-end fashion. During training, we use the Adam optimizer for the generator and the SGD optimizer for the discriminator network. The learning rate for Adam is initially set to 0.0001 and then reduced by a factor of 10 after every 25k iterations with batch size 16. The learning rate for SGD is initially set to 0.01 and then reduced by a factor of 10 after every 40k iterations with batch size 16. Training is terminated after a maximum of 80k iterations. Our system is implemented in the PyTorch framework and trained over three NVIDIA 1080ti, and the source code will be made publicly available. Moreover, the hyperparameters will be discussed in the ablation study.

#### 4.2 Training dataset

We utilize four publicly available benchmark datasets to evaluate our method [43, 44, 28, 29, 45], which include cars overhead with context dataset (COWC), detection in optical remote sensing images (DIOR), vision meets drones (VisDrone), and dataset for object detection in aerial images (DOTA).

The COWC dataset is collected by Lawrence Livermore National Laboratory and includes 58,247 annotated images. The size of the labeled car in each image ranges from 10 x 24 to 20 x 48. Therefore, the car in the obtained LR images can be viewed as a small object.

The DIOR dataset is collected by Northwestern Polytechnical University and contains 23,463 images covered by 20 object classes. The DIOR dataset is a large-scale detection benchmark for remote sensing image processing. The size of the collected...
image is 800 × 800, and the labeled objects contain a few pixels compared to the background, displaying diverse orientations.

The VisDrone dataset is collected by Tianjin University and consists of 10,209 images acquired from various bird’s-eye views of the street. Moreover, the images captured in a wide range of situations contain different numbers of targets.

The DOTA is collected by Wuhan University and contains 2806 images. In the design of this dataset, the researcher’s initial idea is that for the same object with different sizes and diverse directions, the effectiveness of the detector will be effectively checked. Therefore, in contrast to other datasets, the DOTA dataset contains many oriented objects. The number of listed categories is 15, which includes various pixel sizes ranging from 30 × 30 to 1200 × 1200. We select images with small objects as our testing images.

## 4.3 Validation of SR

We compare our method with other GAN-based SR methods, including ESRGAN [31], EEGAN [33], SFTGAN [35], and PPON [39]. To illustrate the impartiality, we utilize the same datasets to retrain these models. The results of the perceptual index (PI), peak signal-to-noise ratio (PSNR), and structural similarity index (SSIM) scores are presented in Table 1. Specifically, a lower PI score indicates generated images with high quality, and higher PSNR and SSIM scores indicate that the generated data distribution better approaches the real data distribution. In Table 1, we can see that the PI of SFTGAN on DOTA and Visdrone are better than our model but with worse performance on PSNR and SSIM. The main reason is that SFTGAN utilizes semantic segmentation maps as categorical prior to guide SR while our work uses GMs as structure priors to guide SR. Thus, SFTGAN generates more visually pleasing textures while our model generates more sharp edges.

We also conduct a visual comparison, as shown in Figure 4. This comparison shows that the primary bicubic interpolation method is unable to recover detailed textures. PPON obtains more natural textures than the bicubic interpolation method, but with dirty content. ESRGAN and SFTGAN can produce less blur but more ambiguous edges. EEGAN produces sharp edges but undesired structural distortions. As expected, our approach significantly improves the SR performance with pleasing visual quality. As shown in Figure 5, by applying the gradient guidance and edge-enhancement strategy in our method, the distortions and artefacts are well mitigated. As shown in Table 2, the performance of our SR model training with and without gradient branches or edge branches is tested. The results show that the gradient guide provides more meaningful information for SR and that the SR is significantly improved by combining the two branches.

### 4.4 Detection with SR

We run the FRCNN to document the object detection performance on LR, HR, other GAN-based SR, and our SR images. Table 3 provides the comparison results in terms of AP50 and F1. The results clearly show that our method achieves the best performance on all datasets. Specifically, our method achieves on average 27.9% and 38.6% higher AP50 and F1-score than the LR images on the COWC dataset, respectively.

We also compare the detection performance by using the LR images, HR images, and SR images with different detectors, as listed in Table 4. The results indicate that there is a considerable performance gap among those detectors. In addition, in terms of AP50, our model achieves an average result that is 3.25% higher than EESRGAN and 1.1% higher than Joint-SRVDNet, as presented in Table 5.

To validate that the detection loss back-propagates into the SR network can focus the SR on generating details that are beneficial for detection purposes, we compare the detection performance between the SR optimization with detection loss and the SR optimization without detection loss, as listed in Table 6. From Table 6, we see that the AP50 performance increases by approximately 1.9% on COWC when the detection loss is incorporated. Clearly, this validates the claim that the detection loss promotes the generator to recover finer detailed information for better detection.
FIGURE 4  Visual comparison of our SR performance with other GAN-based SR methods

FIGURE 5  Visual representation of the different stages in our method
TABLE 2  Comparison with the three branches on datasets. The results show that both branches can enhance the performance of SR and that the gradient guide provides more meaningful information

| Model                      | DOTA     | Visdrone  | COWC     | DiOR     |
|----------------------------|----------|-----------|----------|----------|
|                            | PI↓ PSNR↑ SSIM↑ | PI↓ PSNR↑ SSIM↑ | PI↓ PSNR↑ SSIM↑ | PI↓ PSNR↑ SSIM↑ |
| SR                        | 3.951 28.550 0.794 | 4.103 25.932 0.685 | 4.552 22.224 0.731 | 3.951 25.964 0.767 |
| SR + Gradient             | 3.712 32.692 0.843 | 3.722 28.987 0.773 | 3.742 24.287 0.817 | 3.437 28.101 0.811 |
| SR + Edge-enhanced        | 3.742 30.001 0.829 | 3.842 26.262 0.733 | 4.142 23.366 0.804 | 3.642 26.363 0.791 |
| SR + Gradient + Edge      | 3.604 33.452 0.872 | 3.523 29.956 0.795 | 3.412 26.96 0.835 | 3.257 29.345 0.824 |

TABLE 3  Comparative detection performance of our proposed method and other GAN-based SR with the same detector. The third bold row represents the best performance score when fed actual HR imagery to FRCNN and the fourth bold row indicates the performance obtained by our proposed network

| Model                      | COWC | Dior | DOTA | Visdrone |
|----------------------------|------|------|------|----------|
|                            | AP50 | F1   | AP50 | F1       |
| LR + FRCNN                 | 51.2% | 0.312 | 54.5% | 0.453 |
| bicubic + FRCNN            | 55.5% | 0.431 | 62.4% | 0.566 |
| HR + FRCNN                 | 81.4% | 0.721 | 90.8% | 0.885 |
| PPON + FRCNN               | 72.2% | 0.632 | 70.4% | 0.623 |
| ESRGAN + FRCNN             | 77.2% | 0.654 | 84.9% | 0.702 |
| SFTGAN + FRCNN             | 78.1% | 0.662 | 85.3% | 0.731 |
| EEGAN + FRCNN              | 78.3% | 0.675 | 86.4% | 0.769 |
| Our + FRCNN                | 79.1% | 0.698 | 87.3% | 0.817 |

TABLE 4  Comparative detection performance between LR with poplar detector and our method

| Model                      | COWC     | Dior     | DOTA     | Visdrone  |
|----------------------------|----------|----------|----------|-----------|
|                            | AP50     | F1       | AP50     | F1        |
| LR + YOLO [3]              | 53.7%    | 0.338    | 60.2%    | 0.480     |
| LR + SSD [5]               | 47.9%    | 0.299    | 58.5%    | 0.469     |
| LR + YOLO [6]              | 51.2%    | 0.312    | 58.4%    | 0.466     |
| HR + YOLOv3 [3]            | 79.2%    | 0.699    | 85.9%    | 0.729     |
| HR + SSD [5]               | 76.8%    | 0.649    | 85.3%    | 0.731     |
| HR + FRCNN [6]             | 81.4%    | 0.721    | 90.8%    | 0.885     |
| SR + YOLO [3]              | 76.5%    | 0.648    | 86.6%    | 0.771     |
| SR + SSD [5]               | 77.7%    | 0.660    | 86.4%    | 0.769     |
| Our                       | 79.1%    | 0.698    | 87.3%    | 0.817     |

TABLE 5  Comparative detection performance between EESRGAN and Joint-SRVDNet

| Model                      | COWC | Dior | DOTA | Visdrone |
|----------------------------|------|------|------|----------|
|                            | AP50 | F1   | AP50 | F1       |
| EESRGAN [24]               | 76.8% | 0.714 | 86.4% | 0.769 |
| Joint-SRVDNet [27]         | 77.3% | 0.725 | 90.0% | 0.893 |
| Ours                       | 79.1% | 0.861 | 87.3% | 0.817 |

TABLE 6  Comparative detection performance between SR optimization with detection loss and SR optimization without detection loss

| Model                      | COWC     | Dior     | DOTA     | Visdrone  |
|----------------------------|----------|----------|----------|-----------|
|                            | AP50     | F1       | AP50     | F1        |
| Without detection loss     | 77.2%    | 0.620    | 86.0%    | 0.764     |
| With detection loss        | 79.1%    | 0.698    | 87.3%    | 0.817     |
Furthermore, several examples of the detection results for small objects are visualized in Figure 6 to Figure 9. Our method successfully finds almost all the objects, even though some objects are very small. This demonstrates the effectiveness of our detector on the SOD problem. Figure 9 shows that our model is limited to rotated objects. In the following research, we will research on detecting the small and rotated objects.

## 5 | ABLATION STUDY

To achieve the best version of our proposed model, we perform several experiments to see the impact of the number of outcomes, the balanced parameters of loss functions, basic component RRRDB, and the optimization strategy on the original version of our work. All experiments are performed on the COWC dataset.

### 5.1 | Distribution GAN analysis

We further study the effect of adjusting the number of outcomes in the realness distribution $p_{\text{realness}}$. As shown in Table 7, it can be observed that SR model produces high-quality images as the number of outcomes grows. However, increasing the number of input images will result in substantial time consumption for training. We choose the number of outcomes as 30.

### 5.2 | Hyperparameter analysis

In this section, we analyse the impact of the balanced parameters $\eta, \beta, \xi, \psi, y, k, \varepsilon$ in the network. Since there is no rule for choosing the optimum parameters $\eta, \beta, \xi, \psi, y, k, \varepsilon$ for the

| Number of outcomes | PSNR↑ | SSIM↑ | Training time (min) |
|--------------------|-------|-------|---------------------|
| 2                  | 24.542| 0.7964| 72                  |
| 5                  | 24.901| 0.8045| 74                  |
| 10                 | 25.468| 0.8111| 85                  |
| 20                 | 25.477| 0.8252| 91                  |
| 30                 | 26.960| 0.8354| 95                  |
| 40                 | 27.042| 0.8543| 118                 |
FIGURE 7 Detection results on the DOTA. The first row represents the detection results on the low-resolution image, and the second row represents the detection results on an SR image.

TABLE 8 SR results of our proposed model using different hyperparameter settings for upscale factor 4×. The bold row represents the performance of our SR model with the default hyperparameter settings.

| Hyperparameter settings | PSNR↑ | SSIM↑ |
|-------------------------|-------|-------|
| \( \eta = \beta = 0.02 \) \( \xi = \psi = \gamma = 0.04 \) \( k = \varepsilon = 0.01 \) | 3.6020 | 26.001 |
| \( \eta = \beta = 0.03 \) \( \xi = \psi = \gamma = 0.05 \) \( k = \varepsilon = 0.02 \) | 3.4728 | 26.577 |
| \( \eta = \beta = 0.01 \) \( \xi = \psi = \gamma = 0.05 \) \( k = \varepsilon = 0.02 \) | 3.5475 | 25.977 |
| \( \eta = \beta = 0.02 \) \( \xi = \psi = \gamma = 0.04 \) \( k = \varepsilon = 0.01 \) | 3.6233 | 25.401 |
| \( \eta = \beta = 0.03 \) \( \xi = \psi = \gamma = 0.05 \) \( k = \varepsilon = 0.01 \) | 3.4122 | 26.9620 |

TABLE 9 SOD results in terms of AP\(_{50}\) and F1-score of our proposed model using different hyperparameter settings. The bold row represents the performance of our SR model with the default hyperparameter settings.

| Hyperparameter settings | AP\(_{50}\) | F1 |
|-------------------------|--------|----|
| \( \eta = \beta = 0.02 \) \( \xi = \psi = \gamma = 0.04 \) \( k = \varepsilon = 0.01 \) | 78.6% | 0.669 |
| \( \eta = \beta = 0.03 \) \( \xi = \psi = \gamma = 0.05 \) \( k = \varepsilon = 0.02 \) | 78.9% | 0.687 |
| \( \eta = \beta = 0.01 \) \( \xi = \psi = \gamma = 0.05 \) \( k = \varepsilon = 0.02 \) | 78.8% | 0.681 |
| \( \eta = \beta = 0.02 \) \( \xi = \psi = \gamma = 0.04 \) \( k = \varepsilon = 0.01 \) | 78.2% | 0.679 |
| \( \eta = \beta = 0.03 \) \( \xi = \psi = \gamma = 0.05 \) \( k = \varepsilon = 0.01 \) | 79.1% | 0.698 |

network, we conduct a series of experiments to determine the optimal parameters \( \eta, \beta, \xi, \psi, \gamma, k, \varepsilon \). We observe that the optimal values lead the SR to generate real-looking images with clear textures and sharpness regions. In Tables 8 and 9, we show the performance of our model by varying these hyperparameters on COWC.

5.3 The effectiveness of RRRDB

We use the RRRDB and RRDB as basic components to reconstruct images. We find that RRRDB can further improve the recovered textures, as shown in Figure 10. From Table 10,
**FIGURE 8** Detection results on the DIOR dataset. The first row represents the low-resolution image, and the second row represents the detection results on a low-resolution image. The third row represents the high-resolution image, and the fourth row represents the detection results on an SR image.

**TABLE 10** Performance of our SR model trained with RRRDB and RRDB

| Methods     | PI↓  | PSNR↑ | SSIM↑ | Training time (min) |
|-------------|------|-------|-------|---------------------|
| With RRDB   | 3.5124 | 25.477 | 0.8392 | 110                 |
| With RRRDB  | 3.4122 | 26.9620 | 0.8354 | 95                  |

We see that the PSNR increases by 1.485% and the training time reduces 15 min when the RRRDB is used. The results indicate that RRRDB enforces the reconstructed images to be similar to real, object-specific, and localizable high-resolution images with high-frequency details.

**5.4 Comparisons using reference-free indexes**

To further demonstrate the efficiency of the proposed SR method on the real-world low-resolution samples without simulated sampling, we conduct other experiments for comparison. Different from the above-mentioned assumption that the LR image is down-sampled from the HR image, here we follow the real-world degradation settings that directly feed the LR image into the SR network. In addition, because of the lack of HR ground truth, two reference-free image quality evaluation indexes (average gradient (AG) [46] and natural image quality evaluator (NIQE) [47]) are introduced for comparison. The comparison results are reported in Table 11. Note that the larger the AG and the smaller the NIQE, the clearer the image.

In addition, we use the wavelet transform–based features to compare our method with other GAN-based SR methods. First, we use the wavelet transform to obtain the horizontal component of an image. Then, we use the entropy of the horizontal component to compare the performance of different methods. The comparison results are reported in Table 12. Note that the smaller the entropy value, the clearer the image.

**5.5 Influence of the regression loss**

From Table 13, we see that the AP50 performance increases by approximately 2.7% when the classification loss is incorporated.
FIGURE 9  Examples of detection results on VisDrone. The first row represents the low-resolution image, and the second row represents the detection results on the low-resolution image. The third row represents the high-resolution image, and the fourth row represents the detection results on the SR image. The results indicate that our method successfully locates small objects.

FIGURE 10  Visual comparison of SR performance between RRRDB and RRDB. (a) Ground truth, (b) RRDB, (c) RRRDB

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 region of interest (ROI) images.

5.6  Influence of the classification loss

As shown in Table 13, the $\text{AP}_{50}$ performance increases by nearly 1.9% by using the regression loss to train the generator network. Similar to the classification loss, the regression loss drives the
TABLE 11  Comparison results of AG and NIQE on COWC with the scale factor of 4

| Dataset | Methods  | NIQE↓ | AG↑ |
|---------|----------|-------|------|
| COWC    | SRCNN [48] | 7.536 | 3.789 |
|         | VSDR [49]  | 7.455 | 3.810 |
|         | DRRN [50]  | 7.387 | 3.863 |
|         | BTSRN [51] | 7.166 | 3.977 |
|         | DNCL [52]  | 6.996 | 4.009 |
|         | WSDSR [53] | 6.921 | 4.104 |
|         | NDRCN [54] | 6.806 | 4.198 |
|         | LESRCNN [55] | 6.792 | 4.222 |
|         | A2FSR [56] | 6.526 | 4.481 |
|         | Ours       | 6.445 | 4.508 |

TABLE 12  Using wavelet transform based features

| Dataset | Methods | Entropy value↓ |
|---------|---------|----------------|
| COWC    | LR      | 2.5097         |
|         | Bic     | 2.2838         |
|         | HR      | 0.8774         |
|         | SRGAN [30] | 1.7294       |
|         | PPON [25]  | 1.5832         |
|         | SinGAN [38] | 1.4338        |
|         | ESRGAN [17] | 1.3270        |
|         | SFTGAN [21]  | 1.2701        |
|         | RankSRGAN [22] | 1.1894      |
|         | EEGAN [19]  | 1.0438         |
|         | SSRResCGAN [23] | 0.9921       |
|         | SPSR [20]   | 0.9580         |
|         | Ours       | 0.8980         |

FIGURE 11  Detection illustration on occluded objects. (a) Ground truth, (b) Detection results

5.7 | Occluded SOD

To investigate whether our method can detect occluded objects, we conduct a group of experiments, as shown in Figure 11. The experimental results show that our detector can detect partially occluded objects and fails to detect fully occluded objects.

6 | CONCLUSION

We propose a new GAN-based SOD framework that takes LR imagery as input and yields detection results on corresponding the super-resolved version. In the generator, we introduce two strategies to recover detailed information. In the discriminator, we introduce FRCNN for the task of object detection. Specifically, the discriminator outputs a distribution rather than a single scalar as the measure of realness. During training, the losses of the detection task are backpropagated to the generator. The experimental results show that the proposed SR network with FRCNN yields satisfactory results.

ACKNOWLEDGEMENTS

This work is supported in part by the National Key Research and Development Project under Grant 2016YFB0501003, in part by the National Natural Science Foundation of China under Grant 61690214, Grant 11972182, in part by the Shanghai Technological Innovation Project under Grant 19511120900.

ORCID

Jinzhen Mu  https://orcid.org/0000-0002-0702-4705
Shuang Li    https://orcid.org/0000-0001-9142-5036
Yan Zhou    https://orcid.org/0000-0002-2372-4947
REFERENCES

1. Gong, Y., et al.: Context-aware convolutional neural network for object detection in VHR remote sensing imagery. IEEE Trans. Geosci. Remote Sens. 58(1), 34–44 (2020)

2. Liu, Y., et al.: RoadNet: Learning to comprehensively analyze road networks in complex urban scenes from high-resolution remotely sensed images. IEEE Trans. Geosci. Remote Sens. 57(4), 2043–2056 (2019)

3. Redmon, J., et al.: You only look once: Unified, real-time object detection. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 799–788 (Jun. (2016))

4. Tian, Z., Chen, C., Chen, H.: FCOS: Fully convolutional one-stage object detection. In Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), pp. 9626–9635 (Apr. (2019))

5. Liu, W., et al.: SSD: Single shot multibox detector. In Proc. IEEE Int. European Conf. Computer Vision (ECCV), pp. 21–37 (Dec. (2016))

6. Ren, S., et al.: Faster R-CNN: Towards real-time object detection with region proposal networks. In Proc. Conf. Neural Inf. Process. Syst. (NIPS), pp. 91–99 (Jan. (2015))

7. Law, H.H., Deng, J.: CornerNet: Detecting objects as paired keypoints. [Online]. Available: arXiv:1808.01244 (2019)

8. He, K., Zhang, X., Ren, S.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. 37(9), 1904–1916 (2015)

9. Lin, Y., et al.: Microsoft COCO: Common objects in context. In Proc. IEEE Int. European Conf. Computer Vision (ECCV), pp. 740–755 (Feb. (2015))

10. Chen, X., et al.: 3D object proposals for accurate object class detection. In Proc. Conf. Neural Inf. Process. Syst. (NIPS), pp. 424–432 (Apr. (2015))

11. Cheng, F., et al.: DSSD: deconvolutional single shot detector. [Online]. Available: arXiv:1701.06659 (2017)

12. Ke, X., Li, P.: Dense small face detection based on regional cascade multi-scale method. IEEE Image Process. 13(14), 2796–2804 (2019)

13. Yang, F., Choi, W., Lin, Y.: Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 2129–2137 (Jun. (2016))

14. Li, H., et al.: A convolutional neural network cascade for face detection. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 5325–5334 (Mar. (2015))

15. Bell, S., et al.: Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks. [Online]. Available: arXiv:1512.04143 (2015)

16. Kisanal, M., et al.: Augmentation for small object detection. [Online]. Available: arXiv:1902.07296 (2019)

17. Deng, C., et al.: Extended feature pyramid network for small object detection. [Online]. Available: arXiv:2003.07021 (2020)

18. Singh, B., Najibi, M., Davis, L.S.: Sniper: Efficient multi-scale training. In Proc. Conf. Neural Inf. Process. Syst. (NIPS), pp. 9310–9320 (May. (2018))

19. Zhu, C., et al.: Seeing small faces from robust anchor's perspective. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 5127–5136 (Feb. (2018))

20. Yang, X., et al.: Serdet: Towards more robust detection for small, cluttered and rotated objects. In Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), pp. 3411–3420 (Oct. (2019))

21. Tayara, H., Soo, K., Chong, K.: Vehicle detection and counting in high-resolution aerial images using convolutional regression neural network. IEEE Access 6, 2220–2230 (2017)

22. Ji, H., et al.: Improved faster R-CNN with multi-scale feature fusion and homography augmentation for vehicle detection in remote sensing images. IEEE Trans. Geosci. Remote Sens. 16(11), 2220–2230 (2019)

23. Shen, J., Liu, N., Sun, H.: Vehicle detection in aerial images based on lightweight deep convolutional network and generative adversarial network. IEEE Access 7, 148119–148130 (2019)

24. Jakaia Rabbi, R., et al.: Small-object detection in remote sensing images with end-to-end edge-enhanced GAN and object detector network. Remote Sensing 12(9), 1–5 (2020)

25. Bai, Y., et al.: SOD-MTGAN: Small object detection via multi-task generative adversarial network. In Proc. IEEE Int. European Conf. Computer Vision (ECCV), pp. 206–221 (2018)

26. Li, J., et al.: Perceptive generative adversarial networks for small object detection. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp.1951–1959 (Jun. (2017))

27. Mostofa, M., et al.: Joint-SRVDNet: Joint super resolution and vehicle detection network. [Online]. Available: arXiv:2005.00983 (2020)

28. Dong, Z., et al.: Object detection in high resolution remote sensing imagery based on convolutional neural networks with suitable object scale features. IEEE Trans. Geosci. Remote Sens. 58(3), 2104–2114 (2020)

29. Pang, J., et al.: R2-CNN: Fast small object detection in large-scale remote sensing images. IEEE Trans. Geosci. Remote Sens. 57(8), 5512–5524 (2019)

30. Cheng, M., et al.: Structure-preserving super resolution with gradient guidance. [Online]. Available: arXiv:2003.1308 (2020)

31. Wang, T., et al.: ESRGAN: Enhanced super-resolution generative adversarial networks. [Online]. Available: arXiv:1809.00219 (2018)

32. Rakotonirina, N., Rasoanaivo, A.: ESRGAN+: Further improving enhanced super-resolution generative adversarial network. [Online]. Available: arXiv:2001.08073 (2020)

33. Jiang, K., et al.: Edge-enhanced GAN for remote sensing image super-resolution. IEEE Trans. Geosci. Remote Sens. 57, 5799–5812 (2019)

34. Cheng, M., et al.: Structure-preserving super resolution with gradient guidance. [Online]. Available: arXiv:2003.1308

35. Wang, X., Yu, K., Dong, C.: Recovering realistic texture in image super-resolution by deep spatial feature transform. [Online]. Available: arXiv:1804.02815 (2018)

36. Zhang, W., et al.: RankSRGAN: Generative adversarial networks with ranker for image super-resolution. [Online]. Available: arXiv:1908.06382 (2019)

37. Muhammad, R., Foresti, G., Micheloni, C.: Deep generative adversarial residual convolutional networks for real-world super-resolution. [Online]. Available: arXiv:2005.08953 (2020)

38. Shaham, T., Dekel, T., Michaeli, T.: SinGAN: Learning a generative model from a single natural image. [Online]. Available: arXiv:1905.01164 (2019)

39. Hui, Z., Li, J., Gao, B.: Progressive perception-oriented network for single image super-resolution. [Online]. Available: arXiv:1907.10399 (2019)

40. Xiangli, Y., et al.: Real or not real, that is the question. [Online]. Available: arXiv:2005.08719 (2020)

41. Johnson, J., Alahi, A., Li, F.: Perceptual losses for real-time style transfer and super-resolution. In IEEE Int. European Conf. Computer Vision (ECCV), pp. 694–711 (Mar. (2016))

42. Sheng, L., Mohammad, Z.: Biomedical compound figure detection using deep learning and fusion techniques. IET Image Processing, 12 (6), 1031–1037 (2018)

43. Wang, M., et al.: Optimal segmentation of high-resolution remote sensing image by combining superpixels with the minimum spanning tree. IEEE Trans. Geosci. Remote Sens. 56(1), 228–238 (2017)

44. Haur, J., et al.: Remote sensing image super-resolution using deep residual channel attention. IEEE Trans. Geosci. Remote Sens. 57(11), 9277–9289 (2019)

45. Pan, X., et al.: Super-resolution based on compressive sensing and structural self-similarity for remote sensing images. IEEE Trans. Geosci. Remote Sens. 51(9), 4864–4876 (2013)

46. Chen, A., et al.: A novel stochastic stratified average gradient method: Convergence rate and its complexity. [Online]. Available: arXiv:1710.07783 (2017)

47. Mitra, A., Soundararajan, R., Bovik, A.: Making a ‘completely blind’ image quality analyzer. IEEE Signal Process Lett. 20(3), 209–212 (2013)

48. Dong, C., et al.: Image super-resolution using deep convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell. 38(2), 295–307 (2016)
49. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In Proc. CVPR, pp. 1646–1654 (Jun. (2016))

50. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 3147–3155 (Jun. (2017))

51. Fan, Y., et al.: Balanced two-stage residual networks for image super-resolution. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 161–168 (Mar. (2017))

52. Xie, C., Zeng, W., Lu, X.: Fast single-image super-resolution via deep network with component learning. IEEE Trans. Circuits Syst. Video Technol. 29(12), 3473–3486 (Dec. (2018))

53. Cruz, C., et al.: Single image super-resolution based on wiener filter in similarity domain. IEEE Trans. Image Process. 27(3), 1376–1389 (2017)

54. Cao, F., Chen, B.: New architecture of deep recursive convolution networks for super-resolution. Knowl.-Based Syst. 178, 98–110 (2019)

55. Tian, C., et al.: Lightweight image super-resolution with enhanced CNN. Knowl.-Based Syst. 205, 106235 (2020)

56. Wang, X., et al.: Lightweight single-image super-resolution network with attentive auxiliary feature learning. [Online]. Available: arXiv: 2011.06773 (2021)

57. Tian, C., Xu, Y., Zhang, D.: Asymmetric CNN for image super-resolution. [Online]. Available: arXiv: 2103.13634 (2021)

58. Ferdous, S., Mostofa, M., Nasrabadi, N.: Super resolution assisted deep aerial vehicle detection. Artif. Intell. Machine Learning Multi-Domain Operat. Appl. 11006, 1100617 (2019)

How to cite this article: Mu, J., et al.: Integration of gradient guidance and edge enhancement into super-resolution for small object detection in aerial images. IET Image Process. 1–16 (2021).
https://doi.org/10.1049/ipr2.12288