ESRTMDet: An End-to-End Super-Resolution Enhanced Real-Time Rotated Object Detector for Degraded Aerial Images

Fei Liu, Renwen Chen, Junyi Zhang, Shanshan Ding, Hao Liu, Shaofei Ma, and Kailing Xing

***Abstract*—The degradation of image resolution reduces the detection performance in aerial imagery because it generates a large number of small objects, and accurately detecting these small objects remains a challenge. Existing methods mostly use a superresolution (SR) model to first obtain the SR image of the low-resolution degraded image (I^{LR}) and then use this image as the input of the object detection (OD) network to solve this problem. However, this architecture that involves executing a complex SR network before the detector is time-consuming and makes it hard to achieve real-time model inference. To address this challenge, we propose a simple and effective rotated small OD method, named end-to-end superresolution enhanced real-time rotated object detector (ESRTMDet). First, we design a lightweight embedded feature map superresolution module (ESRM) embedded in the detection model to enhance and amplify the backbone output features, making the detection heads detect small objects more easily. Furthermore, we train a parallel SR network branch (PSRB) simultaneously that uses the backbone feature to restore a high-resolution image. Through our proposed feature alignment loss and feature affinity layer, our PSRB effectively guides the feature map enhancement of ESRM. Finally, through end-to-end joint optimization of the detector and PSRB, the detection performance on I^{LR} is significantly improved. Extensive experiments over DOTA and UCAS-AOD demonstrate that our method can achieve state-of-the-art results. In addition, we discard our PSRB and use I^{LR} as the input during inference, reducing the inference time-consuming of our model. Therefore, our ESRTMDet-X not only achieves 77.11% mean of average precision on the degraded DOTA dataset, but also achieves an amazing inference speed of 337 FPS, thus obtaining the best speed–accuracy tradeoff.**

***Index Terms*—Aerial image, deep learning, remote sensing, rotated object detection (ROD), small object detection (SOD), superresolution (SR).**

I. Introduction

Aerial images obtained from Earth observation and remote sensing technologies provide a bird’s-eye view of the Earth’s surface, depicting complex spatial scenes and numerous diverse objects. Image classification and object detection for aerial images (also known as remote sensing images) are among the most fundamental and challenging research topics in the geoscience and remote sensing communities. In recent years, significant progress has been achieved in these areas thanks to the development of deep learning techniques. For aerial imagery classification tasks, a combination of graph convolutional networks and convolutional neural networks (CNN) is used to extract diverse and discriminative features [1], resulting in superior classification performance for hyperspectral remote sensing images. Furthermore, a multimodal deep learning framework [2] has been developed to effectively utilize information from different modality remote sensing images, achieving state-of-the-art (SOTA) performance in pixel-level remote sensing image classification. For object detection in aerial images (ODAI), it is a challenging task due to the presence of a great number of small, cluttered, large aspect ratio, and arbitrarily oriented objects [3]. In recent years, significant progress has been achieved in ODAI with the development of deep CNN [3], [4], [5], [6], [7], [8], [9]. However, these methods rely on high-resolution (HR) aerial images (I^{HR}) that have a resolution up to half a meter and good imagery quality.

In practice, due to harsh imaging conditions, such as aerial camera shake, short transmission bandwidth, long-range shooting, and undersampled imaging, degraded aerial images [10], [11], [12], [13] are commonly captured. Hence, object detection for degraded aerial images has gained more attention in recent years. In particular, resolution degradation is a common type of degradation, and these degraded images are also called low-resolution (LR) degraded images (I^{LR}), as shown in Fig. 1. Compared to I^{HR}, I^{LR} often lack texture features, and object regions are more blurred, leading to poor detection results [14]. To address the problem of resolution degradation, recent works [14], [15], [16], [17], [18], [19] have introduced superresolution (SR) methods to restore missing texture and features in I^{LR} before or

Manuscript received 16 April 2023; revised 10 May 2023; accepted 17 May 2023. Date of publication 22 May 2023; date of current version 8 June 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 51675265, in part by the Interdisciplinary Innovation Fund for Doctoral Students of Nanjing University of Aeronautics and Astronautics under Grant KXKCXJJ202201, and in part by the Advantage Discipline Construction Project Funding of University in Jiangsu Province under Grant PAPD. *(Corresponding author: Fei Liu.)*

Fei Liu, Renwen Chen, Junyi Zhang, Shanshan Ding, Hao Liu, and Shaofei Ma are with the State Key Laboratory of Mechanics and Control for Aerospace Structures, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: liufei@nuaa.edu.cn; rwchen@nuaa.edu.cn; zhangjunyi@nuaa.edu.cn; shanshanding@nuaa.edu.cn; rtlhxx@nuaa.edu.cn; shaofeima@nuaa.edu.cn).

Kailing Xing is with the Hardware Development Department IV, Wired Product R&D Institute, System Product Wired Product Operation Division, ZTE Corporation, Nanjing 210037, China (e-mail: xingkailing@nuaa.edu.cn).

The code and models will be released at https://github.com/liufeinuaa/aisodet.git.

Digital Object Identifier 10.1109/JSTARS.2023.3278295

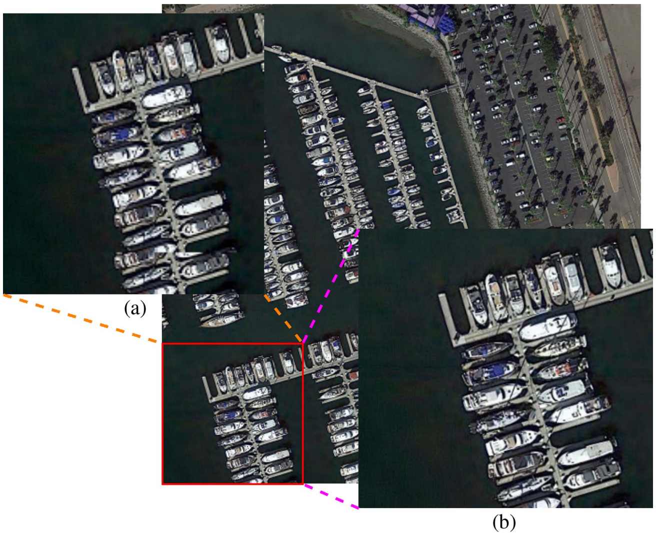

Fig. 1. Diagram of enlarged comparison of the details of objects in the same region between the resolution degraded aerial image and the original HR aerial image. (a) Detailed diagram of the detection object region under the × 2 resolution degraded image. (b) Detailed diagram of the detection object region under the HR image.

after object detection (OD). For instance, Rabbi et al. [17] proposed the edge-enhanced superresolution generative adversarial network (EESRGAN) to obtain SR images prior to executing the detector network. They backpropagated the detection loss and discriminator's loss into the generator net's parameters, optimizing the generative adversarial network (GAN) jointly to produce SR images that more closely resemble HR images. Bai et al. [20] used a two-stage OD method (FasterRCNN) to obtain object patch images first. They then employed the superresolution generative adversarial network (SRGAN) to obtain SR patches and used the discriminator to refine the classification and regression results. Yang et al. [14] proposed the mutual-feel learning (MFL) architecture, which also used SRGAN to obtain SR images prior to executing FasterRCNN. They added a feedback path to the SRGAN discriminator, forming a closed-loop structure. MFL used a discriminator to distinguish the region of interest (RoI) features extracted by the region proposed network (RPN), the RoI features cropped on the SR image, and the RoI feature cropped on the HR image. This approach makes the SRGAN pay more attention to the region where the objects may exist. However, the abovementioned methods are time-consuming and difficult to enhance useful detection features. The optimization of generation is not guided by object information and uses separated iterative optimization for these two different tasks. Moreover, these methods do not consider the problem of rotated ODAI, as they all use horizontal bounding boxes (HBBs) to represent objects. Furthermore, we note that training and inferring the detector directly on the LR input image (I^{LR}) significantly reduces the computational burden. This approach can bring more noticeable model inference acceleration than many model compression techniques [21], which is more conducive to achieving real-time inference of the model.

Overall, the motivation of our article is to use a rotated object detection (ROD) method that integrates the SR network to solve or alleviate the problem of degraded detection performance caused by image resolution degradation under the constraints of easily realize model deployment in practical unmanned aerial vehicle (UAV) systems. Through analysis of existing methods, we have identified following four remaining challenges that need to be addressed to achieve our goals.

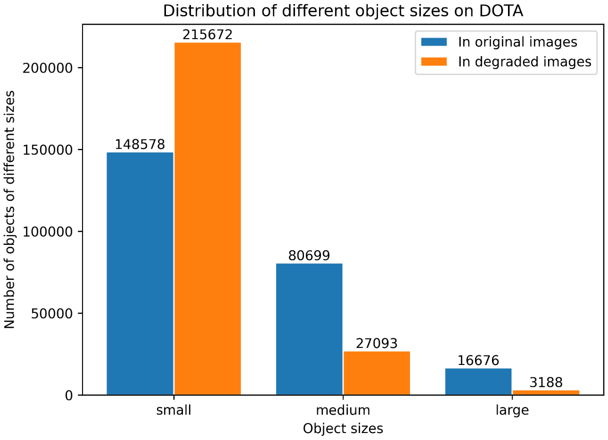

Fig. 2. Distribution of the small, medium, and large sizes of objects on the DOTA datasets with different image resolutions.

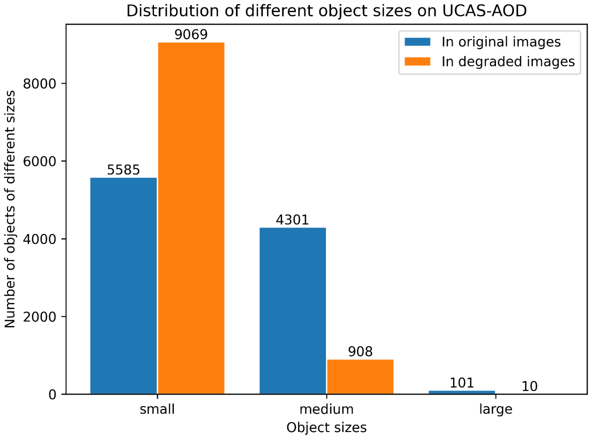

Fig. 3. Distribution of the small, medium, and large sizes of objects on UCAS-AOD datasets with different image resolutions.

1) The challenge of achieving precise detection of small rotated objects. The degradation of resolution in aerial imagery leads to the presence of numerous small targets, as shown in Figs. 2 and 3. However, existing methods continue to use HBBs to locate objects, overlooking the important feature that objects in aerial images can be arbitrarily oriented. As a result, the challenge of achieving accurate detection of small rotated objects in the field of OD remains unsolved.
2) The challenge of achieving overall model lightweight. The majority of existing OD methods that integrate the SR network use SOTA image SR methods (based on GAN) to directly upsample degraded images, then perform a two-stage detector on these SR images, which results in complex model architecture, a large number of model parameters, and difficulty achieving the overall model that is lightweight.

3) The challenge of jointly optimizing different types of models. Because SR and OD networks are designed for different types of tasks, hence, the features and concerns extracted by these two networks are quite different. As a result, the architecture for OD combined with SR is hard to realize end-to-end joint optimization. Existing methods typically adopt a training process where these two types of models are separately trained and then fine-tuned through joint training. This training pipeline is often suboptimal and inefficient [14], [22] due to the lack of exchange of information between different models, and it takes significant time spent on independent model training in advance.
4) The challenge of ensuring fast model inference. Considering the actual deployment requirements of possible UAV systems, the detection model must have a fast inference speed, achieving at least 60 FPS to ensure stable and reliable detection. But existing OD methods that combine SR usually demand full execution of the complex SR network during inference, intensifying computational burden and leading to slow inference speeds (often below 60 FPS), which fails to meet actual deployment requirements.

Therefore, to realize a real-time ROD method in aerial images, we propose a series of simple and effective models named ESRTMDet. Our method not only solves the drawbacks of the abovementioned detectors combined with the SR method but also obtains the fastest detection speed we know of. In summary, our key contributions are as follows.

1) We propose a lightweight embedded feature map super-resolution module (ESRM) that comes after the detection backbone and before the neck. Our ESRM effectively uses valuable texture enhancement features learned by the parallel superresolution network branch (PSRB) to enhance the detection head's ability to extract small object features. And our ESRM does not bring too much additional computational burden. Through ESRM, we have alleviated the challenges caused by rotating small target detection (challenge 1), and by embedding ESRM into the lightweight detection model, we ensure the lightweight of the overall model (challenge 2).
2) We use the PSRB as an auxiliary network and employ the feature affinity layer (FAL) and feature alignment loss ($\mathcal{L}_{\text{AL}}$) to guide the ESRM in restoring high-frequency texture information, thus enhancing the amplification quality of feature maps. In addition, our PSRB is not involved in model inference, ensuring real-time detection ability, which solves challenge 4.
3) To enable the PSRB to focus more on the regions where detection objects are present, we generate RoI weights using the predicted output of the classification branch of detector heads. Our RoI weights optimize PSRB, ESRM, and FAM training, allowing for effective end-to-end joint optimization between these two heterogenous learning tasks, hence, challenge 3 is also solved.
4) A series of experiments on the DOTA and the UCAS-AOD datasets demonstrate the effectiveness of our method. Our ESRTMDet-X achieves 77.11% mAP on DOTA with single-scale training and testing, as well as 89.5% and 95.0% on UCAS-AOD using VOC2007 and VOC2012 metrics, respectively, achieving SOTA detection performance on aerial I^{IR}. Furthermore, our proposed model achieves an impressive inference speed of 337 FPS, making it the best tradeoff between speed and accuracy for ROD in degraded aerial images, as far as we know. Therefore, our research provides significant practical value for the deployment of deep learning algorithms in actual UAV systems.

II. Related Work

In this section, we review recent related works on three aspects: rotated ODAI, SR networks, and the methods of combining SR and OD (SR+OD) in aerial images, as our proposed method integrates both SR and ROD.

A. Rotated ODAI

In the last decade, significant progress has been made in the field of OD, with notable advancements by [23], [24], [25], [26], [27], [28], [29]. At the same time, significant progress has been made in ODAI. For example, Wu et al. [30] combined a novel spatial-frequency channel feature with fast image pyramid estimation and ensemble classifier learning in the classic VJ [31] detection framework to achieve the most advanced detection performance among nondeep learning methods. However, this approach is limited by the detection framework, which only allows for the use of HBBs to represent objects, it is difficult to expand this approach to use more precise rotated bounding boxes (RBBs) to represent objects. Oriented OD, also known as ROD is a subfield of OD that utilizes more precise RBBs to represent objects. And in recent years, it has attracted considerable attention due to its potential applications in various fields, such as management, remote sensing, precision agriculture, national defense, emergency rescue, and disaster relief [4], [32], it has also become the most important research subtopic in remote sensing image OD tasks.

To tackle the challenge of detecting rotated objects in aerial images, one possible approach is to use rotated anchors, such as rotated RPN [32], which places anchors with different angles, scales, and aspect ratios on each location. However, densely rotated anchors result in extensive computations and memory usage. To address these issues, Ding et al. [4] proposed the RoI transformer that learns rotated RoIs from horizontal RoIs produced by RPN, which significantly improves the accuracy of oriented OD. However, this method increases the network complexity and requires fully connected layers and RoI alignment operations during the learning of rotated RoIs. On the other hand, Xu et al. [5] proposed a new representation called gliding vertices, which achieves ROD by learning four vertex gliding offsets on the regression branch of the FasterRCNN head [23]. Although this method simplifies the computation by avoiding RoI alignment and fully connected layers, it still uses horizontal RoIs and is based on a two-stage detection architecture, which is time-consuming and computationally expensive.

To overcome these limitations, some studies [6], [33] explored one-stage oriented OD frameworks based on the RetinaNet [24], which outputs object classes and RBBs without region proposal generation and RoI alignment operations. In addition, in recent years, there has been rapid development of anchor-free detectors in general OD tasks [26], [27], [28], [29], [34]. This mechanism significantly reduces the number of design parameters that require heuristic tuning and tricks for good performance. This simplifies the detector, especially during training and decoding phases [26], [34]. Several studies have explored anchor-free mechanisms for ROD. Pan et al. [35] developed a dynamic refinement network based on the anchor-free CenterNet [28]. He et al. [36] utilized attention mechanisms to refine the performance of remote sensing OD in a one-stage anchor-free network framework. Gong et al. [37] proposed an anchor-free oriented proposal generator to replace the RPN for horizontal boxes in the FasterRCNN detector, which resulted in improved performance. Li et al. [38] proposed an effective anchor-free method called oriented RepPoints, which uses an adaptive point set to capture the semantic and geometric features of an oriented object as a fine-grained representation. Liu et al. [8] used a Gaussian distribution to constrain the RBB and proposed a new assignment method suitable for rotated detection tasks. They combined this method with YOLOX to obtain stronger detection performance. However, the use of RBB representation causes problems, such as boundary discontinuity and square-like issues, making rotational IoU losses indifferentiable, which hinders the use of anchor-free methods. Therefore, Yang et al. designed GWD [39] and KLD [40] regression loss based on Gaussian Wasserstein distance and Kullback–Leibler divergence, respectively. These methods can be used with the anchor-free method FCOS [26] and result in performance improvements in ROD tasks.

B. SR Network

SR is a technique that generates an HR image using an LR image, with the aim of recovering high-frequency texture information [41], [42]. Superresolution CNN (SRCNN) [43] was the first to successfully use CNN in the SR problem. SRCNN's structure is straightforward, consisting of only three CNN layers, and it processes preupsampled images obtained by bicubic interpolation. Residual learning, which uses skipping connections to avoid gradient vanishing, makes the design of deep networks possible compared with the original stacked CNN [44]. Inspired by the ResNet architecture, the enhanced deep superresolution (EDSR) network [45] has been proposed, which removes batch normalization layers (BN) in each residual block (ResBlock) of ResNet since BNs get rid of range flexibility from the network and achieves performance improvement. With the development of deep learning, GAN has shown a remarkable ability for SR problems. Superresolution GAN (SRGAN) [46] focuses the generator on recovering high-frequency texture information using perceptual loss. Enhanced SRGAN [47] is developed by removing the BN in the generator and designing a residual-in-residual dense block to replace the normal ResBlock, achieving more significant performance improvement. However, most SR methods pursue better results in SR by using models with a large number of parameters, leading to higher computational burden and lower network inference speed.

C. Methods of Combining SR and OD (SR+OD) in Aerial Images

The use of SR as a preprocessing step in OD has proven to be effective in various OD tasks [48]. Shermeyer et al. [49] also demonstrated the usefulness of SR for OD performance on satellite imagery. Courtrai et al. [50] used an SR network based on GAN to generate SR images, which are then fed into the detector to improve detection performance. Rabbi et al. [17] used a Laplacian operator to extract edges from input images to enhance the ability to reconstruct HR images, resulting in improved performance in object localization and classification. Small-object detection (SOD)-multitask GAN (MTGAN) [20] proposed using an OD network to adaptively generate RoI object patches for subsequent restoration and detection. Wang et al. [19] introduced the effectiveness of SR for OD in the remote sensing field, as well as an SR model based on multifeature fusion and CycleGAN structure, to enhance images. Bashir et al. [18] improved the SR framework by incorporating a cyclic GAN and residual feature aggregation (RFA) and used YOLO as the detection network to detect objects on SR images. Yang et al. [14] added a feedback path to take FasterRCNN's RPN results to the SRGAN discriminator, forming a closed-loop structure and making the SRGAN pay more attention to the region where the object may exist. In these works, the SR structure has effectively addressed the challenges of small objects and LR inputs. However, compared with single detection models, additional computation is introduced due to the enlarged scale of the input image to HR size, and the cost of the SR network cannot be ignored. Unlike the aforementioned work, where SR is applied at the start stage, using the SR network only as an auxiliary method to enhance SOD performance without participating in model inference is a more promising architecture. Zhang et al. [22] adopted this architecture, using EDSR as an auxiliary network and YOLO v5 backbone fusion features as input to EDSR to restore the HR image. However, this method still lacks information communication between these two different tasks.

Moreover, Wu et al. [51] addressed SOD problems by converting them into semantic segmentation problems and proposed UIU-Net for infrared SOD by utilizing an interactive cross attention mechanism and the ReSidual U-blocks module to improve the classical UNet framework, resulting in the most advanced segmentation performance. However, while the minimum external rectangle postprocessing method can be used to obtain the rotation detection box from the mask, it still cannot be directly applied to the rotating small target detection problem of optical remote sensing images because the commonly used aerial image target detection data lacks fine semantic segmentation masks of the objects.

III. Method

In this section, we introduce our proposed method. First, we provide a brief overview of the baseline model, which is the basic rotated detection method we adopt. Second, we introduce the specific network used in our PSRB, which is a paralleled SR network branch. Next, we describe our proposed ESRM in detail. This module is embedded after the detector backbone and before the neck, and it aims to enhance the feature maps to improve the detection performance. Finally, we present the overall architecture of our proposed method and provide the optimization details.

A. Basic Rotated Detection Method as Baseline

In previous works [14], [17], [22], the object orientation in I^{LR} was not taken into account, and these methods still used HBBs to indicate objects. However, the number of small objects in resolution-degraded aerial images increases greatly, as shown in Figs. 2 and 3. Conventional HBB representation introduces background information that is not conducive to accurately locating objects. Therefore, research is necessary to detect objects in resolution-degraded aerial images using a more accurate RBB representation. The RBB is usually represented as follows:

$$(x, y, w, h, \theta) \tag{1}$$

where, $\theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ denotes the clockwise rotated angle from the image coordinate system position direction of x to the bounding box relatively coordinate system position direction of x. We use the long edge definition format [39] where the width w must be larger than the height h. Recently, the success of the transformer architecture [52], [53], [54] in the image comprehension field has drawn attention to improving classic CNNs. Among them, ConvNetXt [55] uses large kernel convolutions to increase the feature receptive field and capture global context, which overcomes the shortcomings of classic 3 × 3 kernel convolutions and achieves significant performance improvements. The real-time models for object detection (RTMDet) series model [9], based on the YOLOX series model, uses large-kernel depthwise convolutions to replace classic 3 × 3 convolutions to build basic CSP layers [56]. This balances the performance and inference overhead of convolution well. The RTMDet model not only uses large-kernel depthwise convolutions but also has compatible capacities in the backbone and neck. It introduces soft labels when calculating matching costs in the dynamic label assignment and uses better training techniques, all of which efficiently improve detection performance. Furthermore, this model can be easily modified for ROD tasks by modifying the output number of the regression branch (from 4 to 5, increasing the prediction of a rotation angle) and using the simplest rotation IoU loss, which we named rRTMDet in this article. In this article, we chose the one-stage anchor-free rotated detector rRTMDet as our basic rotated detection model and baseline. We trained rRTMDet directly on the DOTA I^{LR} and OD results shown in Table I.

B. Parallel Superresolution Network Branch

In this section, we present the structure of the PSRB, which is depicted in Fig. 5. The PSRB comprises three key components: the feature encode module (FEM), the feature decode module (FDM), and the feature up-sample module (FUM).

TABLE I
DETECTION PERFORMANCE OF THE rRTMDET SERIES MODEL ON THE DOTA I^{LR} AND I^{HR} WITH SINGLE-SCALE TRAINING AND TESTING

Methods	mAP(%) on I^{HR}	mAP(%) on I^{LR}
rRTMDet-tiny	75.36	68.93
rRTMDet-S	76.70	71.09
rRTMDet-M	78.24	73.91
rRTMDet-L	78.56	75.26
rRTMDet-X	77.31	75.78

We propose a feature encode module (FEM) based on the stem network architecture of the rRTMDet detection network, as detailed in Fig. 4. The classic SR net [45], [47] directly uses I^{LR} as input and retains low-level image structured information in its extracted features, which also exists in the stem part of the detector's backbone network. To better leverage these features and promote the learning of high-level features of the SR network, we incorporate the FEM before the classic SR network. This allows us to make the lowest input features between the two tasks as similar as possible, which facilitates end-to-end joint training and optimization. The FDM and FUM are part of the EDSR model [45], as illustrated in Fig. 5. Specifically, we adjust the number of channel dimensions in the first layer convolution of EDSR to match the output feature channel dimension of FEM. In addition, based on our experiments in Section IV-C, we propose using only four stacked layers of ResBlocks in our FDM instead of the original sixteen, since deeper PSRB did not improve performance but significantly prolonged training time. The FUM is subpixel convolution layers [57], the same as EDSR's FUM. Since the model feature maps are downsampled by a factor of 2 after FEM, we use the FUM to upsample four times, and then reduce the channel dimension to 3 through the final convolution, to obtain the final SR image. Therefore, our proposed PSRB performs ×2 image SR task.

C. Embedded Feature Map Superresolution Module

Previous studies [22] and [58], have demonstrated that incorporating PSRB into the original architecture can enhance the performance of general OD tasks and semantic segmentation tasks when using I^{LR} as input. However, in our experiments (see Section IV-C), we found that adding PSRB to the architecture for ROD tasks in aerial images resulted in only minor performance improvements.

The limited performance gains from adding a pixel-shuffle residual block (PSRB) to the original architecture can be attributed to the need for effective information interaction between the two different task models to avoid interference of background information due to more accurate RBB annotation required for ROD. Furthermore, the reduced feature size of the neck due to I^{LR} input size halves that of I^{HR}, making it harder for the model to pay attention to small objects, thus greatly reducing the rotated detection model's performance. Hence, we propose the use of a lightweight ESRM to improve the performance of ROD tasks in aerial images when using I^{LR} as the input image. We embed our ESRM after the C2, C3, and C4 output of the detector's backbone, using lightweight large-kernel depthwise

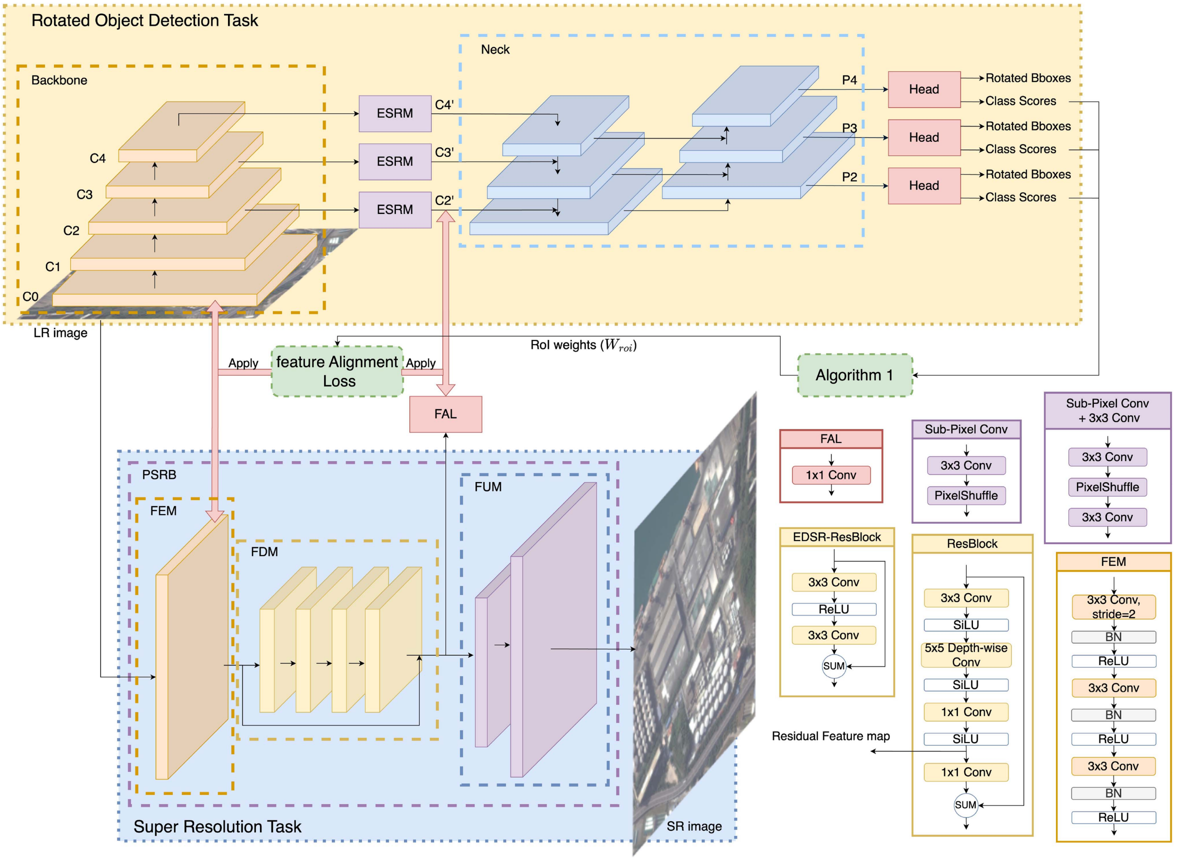

Fig. 4. Overall architecture of our proposed ESRTMDet, and the specific composition details of our proposed modules.

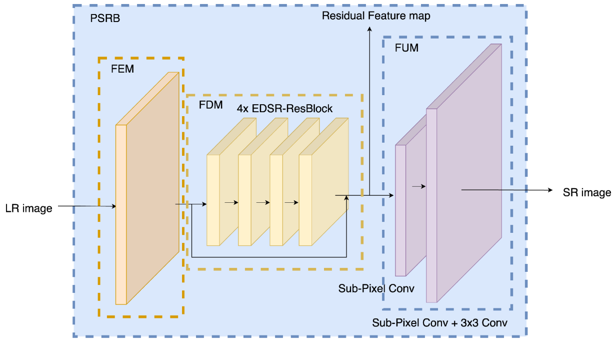

Fig. 5. Architecture of our proposed PSRB.

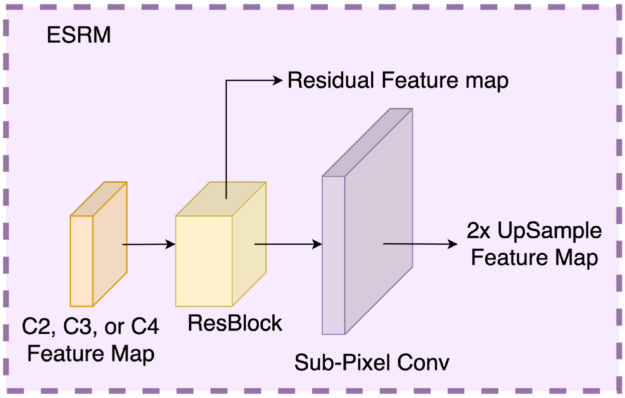

Fig. 6. Architecture of our proposed ESRM.

convolution and 3 × 3 convolution to build the basic ResBlocks, following the approach used in [9]. Our ESRM increases the output feature map size of the backbone by 2 times, equivalent to the I^{HR} as the model input. See Fig. 6 for our ESRM structure and Fig. 4 for the detailed composition of the ResBlock.

D. Feature Alignment Loss and FAL

We design an additional FAL, as shown in Fig. 4, to enhance the information interaction between the two different task models. First, we downsample the output feature map of PSRB's FDM four times and process it through the FAL. Next, we calculate our proposed feature alignment loss ($\mathcal{L}_{\mathrm{AL}}$) between the output feature of our ESRM on the C2 backbone and the FAL to minimize the similarity difference between these two output features. This enables us to jointly optimize the two types of tasks so that effective information can be exchanged between these two types of models, thus optimizing the overall architecture.

Our proposed feature alignment loss uses the normalized Gram matrix to calculate the internal structure similarity of the feature map, as shown in (5). Specifically, for any feature map $F \in \mathbb{R}^{C\times H\times W}$, we can compress its spatial dimensions to obtain a feature map $F' \in \mathbb{R}^{C\times HW}$. The F' can be represented by its row vector $f_i \in \mathbb{R}^{1\times HW}$, $i = 1,\ldots,C$ as

$F'^T = [\ f_1 \quad f_2 \quad \ldots \quad f_C\ ]$. We use the Gram matrix (2) to calculate the similarity between the row vectors. However, there may be numerical issues when using the Gram matrix directly. Therefore, we first regularize f_i as norm_$f_i = ||f_i||_2$ before calculating the Gram matrix. We actually use the normalized Gram matrix, as shown in (4), to address the numerical problems. We represent the matrix composed of the normalized norm_f_i as norm_$F = [$ norm_f_1 norm_f_2 $\ldots$ norm_f_C $]$. A more direct calculation method of the normalized Gram matrix is shown in (5).

$$G_{ij} = f_i \cdot f_j^T \tag{2}$$

$$G(F) = F' \cdot F'^T \tag{3}$$

$$G'_{ij} = \left(\frac{f_i}{||f_i||_2}\right) \cdot \left(\frac{f_j}{||f_j||_2}\right)^T \tag{4}$$

$$G'(F) = \frac{F' \cdot F'^T}{\text{norm\_}F^T \cdot \text{norm\_}F}. \tag{5}$$

We use the normalized Gram matrix to calculate the similarity between different feature maps. Specifically, we measure the structural relation difference between different input feature maps using the weighted Euclidean distance, which we define as our proposed feature alignment loss $\mathcal{L}_{\mathrm{AL}}$. This is shown in the following:

$$\mathcal{L}_{\mathrm{AL}}(F_1,F_2)=\frac{1}{C^2}\sum_{i}^{C}\sum_{j}^{C}(G'(F_1\odot W_{\mathrm{roi}})_{ij}-(G'(F_2\odot W_{\mathrm{roi}})_{ij})^2 \tag{6}$$

where, $\odot$ represents elementwise multiplication, and W_{roi} is our proposed RoI weights. F_1 and F_2 represent two input feature maps that need to be aligned with each other.

Because we want to improve the attention of the PSRB to the image RoI while enhancing the low-level structural features in the corresponding region of the detection feature. To achieve this, we generate RoI weights using the classification branch of the detection model. The detailed calculation method for generating RoI weights is provided in Algorithm 1. Among them, the variable cls_scores corresponds to the output of the detector's classification branch. The hyperparameter α serves as the weight ratio between the object-containing region and the background area. We set the default value of α to 5. Our experiments, depicted in Fig. 8, indicate that our proposed model is not highly sensitive to this hyperparameter. Our proposed weights W_{roi} are utilized as the weighted coefficients for $\mathcal{L}_{\mathrm{SR}}$ [see (8)] and $\mathcal{L}_{\mathrm{AL}}$ [see (6)] in our model. This way, the results of the detection model can influence the SR model, and through W_{roi} they form a closed loop in our overall architecture.

E. Overall Architecture and Optimization

The overall architecture of our proposed method is shown in Fig. 4, and the end-to-end training pipeline,as Algorithm 2 shows in the following.

Using this end-to-end training pipeline, the PSRB and rRTMDet models can be trained jointly. Our method does not require training the generator and discriminator separately, such as in GAN models found in the literature [14], [17], [18], nor does it require training different task networks independently. As a result, our method is simpler and easier to deploy. The detection loss $\mathcal{L}_{\mathrm{Det}}$, SR loss $\mathcal{L}_{\mathrm{SR}}$, and total loss $\mathcal{L}_{\mathrm{Total}}$ of our method is calculated as follows, respectively:

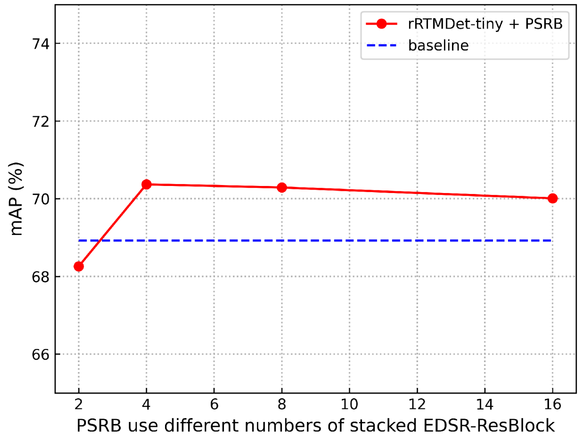

Fig. 7. Detection performance curve of rRTMDet-tiny + PSRB using different numbers of stacked layers of EDSR-ResBlock on the DOTA I^{LR} with single-scale training and testing.

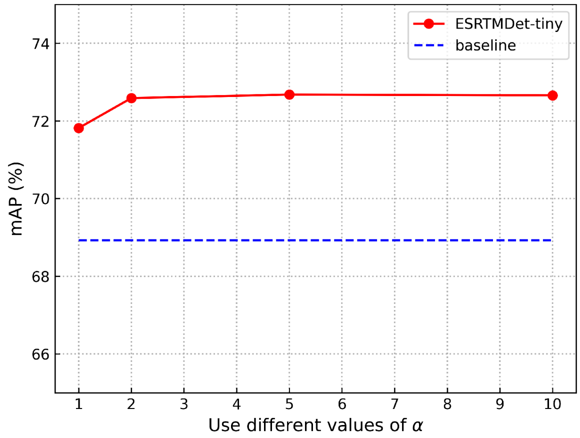

Fig. 8. Detection performance curve of ESRTMDet-tiny using different values of α in Algorithm 1 on the DOTA I^{LR} with single-scale training and testing.

$$\mathcal{L}_{\mathrm{Det}} = \frac{1}{N}\left(\lambda_1 \sum_i \mathcal{L}_{\mathrm{cls}}(p_i, l_i) + \lambda_2 \sum_i \mathcal{L}_{\mathrm{reg}}(b_i, g_i)\right) \tag{7}$$

$$\mathcal{L}_{\mathrm{SR}} = \lambda_3 \mathbf{L2}(I^{\mathrm{SR}} - I^{\mathrm{HR}}, W_{\mathrm{ar}}) \tag{8}$$

$$\begin{aligned}\mathcal{L}_{\mathrm{Total}} &= \mathcal{L}_{\mathrm{Det}} + \mathcal{L}_{\mathrm{SR}} \\ &\quad + \lambda_4\left(\mathcal{L}_{\mathrm{AL}}(A^{\mathrm{SR}'}, C2^{\mathrm{RF}}) + \mathcal{L}_{\mathrm{AL}}(E^{\mathrm{SR}}, C0)\right)\end{aligned} \tag{9}$$

where, N indicates the number of positive samples in the rRTMDet head, i is the index of a positive sample in a batch, p_i and b_i are the predicted object category and decode bounding box in the head. l_i represents the ground-truth category of ith object and g_i is the ground-truth bounding box. And we follow the RTMDet default setting employing quality focal loss [59] as the $\mathcal{L}_{\mathrm{cls}}$, use rotated IoU loss [60] as the $\mathcal{L}_{\mathrm{reg}}$. And we also follow the common practice in the general SR task using L2 loss in our

Algorithm 1: Calculation Method of RoI Weights.

```
Input:
    cls_scores, $\alpha$
Output:
    $W_{\text{roi}}$ represent RoI weights
masks = [ ]
for cls_scores in cls_scores do :
    max_score $\leftarrow$ max(cls_scores, dim = 1)
    mean $\leftarrow$ mean(max_score)
    std $\leftarrow$ std(max_score)
    mask $\leftarrow$ float(max_score $\geq$ mean + std)
    mask $\leftarrow$ interpolate(mask, scaler_factor = $2^{i+1}$)
    masks $\leftarrow$ append(mask)
masks $\leftarrow$ logical_or(masks)
attention_region $\leftarrow$ interpolate(masks, scaler_factor = $\frac{1}{4}$)
$W_{\text{roi}}$ = attention_region $\times (\alpha - 1) + 1$
return $W_{\text{roi}}$
```

Algorithm 2: End-to-end Training Pipeline.

```
Input:
    $I^{\text{LR}}$ represent LR images
Output:
    $I^{\text{SR}}$ represent SR images
    $O^{\text{Det}}$ represent detection results
for epoch in max_training_epoches do :
    Step 1: Detector forward:
        Input $I^{\text{LR}}$ into Backbone and obtain $C0$, $C1$, $C2$, $C3$, and $C4$ backbone feature maps;
        Input $C2$, $C3$, $C4$ into ESRM to obtain $C2'$, $C3'$, $C4'$ and $C2$'s ESRM output residual feature map $C2^{\text{RF}}$;
        Input $C2'$, $C3'$, $C4'$ into Neck and obtain $P2$, $P3$, and $P4$ feature maps;
        Input $P2$, $P3$, $P4$ into Head to obtain $O^{\text{Det}}$ and each classification branch output cls_scores;
    Step 2: PSRB forward:
        Input $I^{\text{LR}}$ into FEM and obtain SR encoder feature map $E^{\text{SR}}$;
        Input $E^{\text{SR}}$ into FDM and obtain SR decoder residual feature map $R^{\text{SR}}$;
        Input $R^{\text{SR}}$ into FUM and obtain $I^{\text{SR}}$;
    Step 3: Joint optimize:
        Input $R^{\text{SR}}$ into FAL and obtain SR affinity residual feature map $A^{\text{SR}}$;
        Downsample $A^{\text{SR}}$ to $C2^{\text{RF}}$'s feature map size, named $A^{\text{SR}'}$;
        Input cls_scores into Algorithm 1 to obtain attention region weights $W_{\text{roi}}$;
        Use $W_{\text{roi}}$ as the weights in Alignment loss $\mathcal{L}_{\text{AL}}$ and SR loss $\mathcal{L}_{\text{SR}}$;
        Calculate $\mathcal{L}_{\text{AL}}$ between $A^{\text{SR}'}$ and $C2^{\text{RF}}$, and $\mathcal{L}_{\text{AL}}$ between $E^{\text{SR}}$ and $C0$;
        Calculate $\mathcal{L}_{\text{SR}}$ and detection loss $\mathcal{L}_{\text{Det}}$;
        Through $\mathcal{L}_{\text{Total}}$ use arbitrary optimizer to joint optimize our model
```

$\mathcal{L}_{\text{SR}}$ and use our proposed W_{roi} as the loss weights. In addition, $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are loss balance parameter, which we set to {1, 2, 1, 10} by default.

IV. Experiments

Our method was evaluated on two challenging aerial ROD datasets, i.e., DOTA and UCAS-AOD.

A. Datasets

DOTA: [3] is a large-scale aerial OD dataset consisting of 2806 aerial images ranging from 800×800 to 4000×4000, containing a total of 188 282 instances of 15 common object categories, such as planes (PL), baseball diamonds (BD), bridges (BR), ground track fields (GTF), small vehicles (SV), large vehicles (LV), ships (SH), tennis courts (TC), basketball courts (BC), storage tanks (ST), soccer-ball fields (SBF), roundabouts (RA), harbors (HA), swimming pools (SP), and helicopters (HC). Both the training and validation sets are used for training, while the test set is used for testing. In accordance with [6], we extract a series of 1024×1024 patches with a 200-pixel overlap from the original images to create our HR (I^{HR}) datasets for experimentation. We then use the bicubic method to downsample I^{HR} by 2 times, getting in 512×512 resolution-degraded images (I^{LR}). After the image degradation process, we observe the distribution of small, medium, and large objects on the DOTA dataset in Fig. 2. Fig. 2 shows that the number of small objects increased by 45.2% after ×2 image resolution degradation, while the number of large objects decreased by 80.9%. This change in object distribution significantly affects the performance of the baseline model, as demonstrated in Table I.

UCAS-AOD: [61] is an aerial image dataset designed for rotated SOD, which contains 1510 images including 510 car images and 1000 plane images, with a total of 14 596 instances. As is customary, we randomly divided it into the training set, validation set, and test set with a ratio of 5:2:3. To experiment with the UCAS-AOD dataset, we resized all images to 836×836 to obtain HR (I^{HR}) images and used the same method as in DOTA experiments to obtain corresponding resolution-degraded images (I^{LR}) with a size of 416×416. Fig. 3 shows the change in the number of objects of different sizes on the UCAS-AOD dataset after downsampling. The analysis reveals that the number of small objects increased by 62.4% after the typical ×2 resolution degradation processing. However, detecting small objects accurately is more challenging than detecting medium and large objects, and as a result, the detection accuracy of the baseline model directly detecting on I^{LR} decreased significantly.

B. Implement Details

We followed the experimental configuration of RTMDet, using CSPNetXt [9] as the backbone and CSPNetXt-PAFPN as the neck for our ESRTMDet. For fair comparisons with other methods, we used CSPNetXt-L and CSPNetXt-X as backbones,

TABLE II
RESULTS OF ABLATION EXPERIMENTS FOR rRTMDET-TINY + PSRB ON THE DOTA I^{LR} WITH SINGLE-SCALE TRAINING AND TESTING. RRTMDET-TINY + PSRB MEANS ADDING A PSRB ON THE DETECTION NETWORK. INPUT FEATURE MAPS REPRESENT USING DIFFERENT SIZES OF FEATURE MAPS AS THE SR NETWORK INPUTS

Methods	Input feature maps	mAP(%)
rRTMDet-tiny + PSRB	$C0$	70.01
	$C1$	69.47
	$C2$	68.97
	$C3$	67.90
	$C4$	65.64
	$C2 + C4$	68.99
	$C2 + C3 + C4$	69.01
	$C0 + C1 + C2 + C3 + C4$	70.14

TABLE III
ABLATION EXPERIMENT FOR THE INSERTION POSITION OF THE FEATURE UPSAMPLING METHOD. OUR rRTMDET-TINY + BICUBIC MEANS USE THE CLASSICAL BICUBIC METHOD AS THE FEATURE UPSAMPLING METHOD. AFTER BACKBONE MEANS THE UPSAMPLING METHOD EMBEDDED IN THE POSITION BEHIND THE BACKBONE AND BEFORE THE NECK. AFTER NECK INDICATES THE UPSAMPLING METHOD INSERTED IN THE POSITION AFTER THE NECK BEFORE THE HEAD

Methods	Embeded position	mAP(%)
rRTMDet-tiny + Bicubic	after backbone	70.29
	after neck	69.09

TABLE IV
RESULTS OF EXPERIMENTS FOR rRTMDET-TINY + UP-SAMPLING ON THE DOTA I^{LR} WITH SINGLE-SCALE TRAINING AND TESTING. RRTMDET-TINY + UPSAMPLING MEANS EMBEDDING A FEATURE MAP UPSAMPLING METHOD BEFORE THE NECK. THE UPSAMPLING METHOD REPRESENTS USING DIFFERENT UPSAMPLING METHODS TO EXECUTE FEATURE MAP UPSAMPLING. ESRM IS OUR PROPOSED METHOD

Methods	The up-sampling method	mAP(%)
rRTMDet-tiny + up-sampling	Bicubic interpolation	70.29
	De-convolution [62]	70.53
	Sub-pixel convolution [57]	70.68
	ESRM	71.07

TABLE V
RESULT OF ABLATION EXPERIMENTS FOR ESRTMDET-TINY ON THE DOTA I^{LR} WITH SINGLE-SCALE TRAINING AND TESTING. √ MEANS THE MODULE IS USED. WE CHOOSE THE rRTMDET-TINY AS THE BASELINE. PSRB MEANS PARALLEL SR NETWORK BRANCH. ESRM MEANS EMBEDDED SR MODULE. $\mathcal{L}_{AL}$ MEANS USING FEATURE ALIGNMENT LOSS IN OPTIMIZATION. W/FEM INDICATES USING THE FEM IN PSRB. W/FAL INDICATES USING THE FAL TO PROCESS THE PSRB OUTPUT FEATURE BEFORE CALCULATING $\mathcal{L}_{\mathrm{AL}}$

Methods	PSRB	w/FEM	ESRM	$\mathcal{L}_{AL}$	w/FAL	mAP(%)
Baseline						68.93
ESRTMDet-tiny	√					70.01
			√			71.07
	√		√			71.69
	√	√		√		70.67
	√	√	√	√		72.35
	√	√	√	√	√	72.68

and CSPNetXt-tiny for other ablation experiments if not specified. During the model training phase, we used random flipping and rotation as augmentation techniques to avoid overfitting, following the original RTMDet series model training configuration. No augmentations were used during the testing phase. All experiments were conducted on an NVIDIA RTX 3090 GPU with a batch size of 4, using the AdamW optimizer with a base learning rate of 2.5×10^{-4}, a momentum of 0.9, and a weight decay of 0.05. We trained all models for 36 epochs for DOTA and 108 epochs for UCAS-AOD, using the same training schedules as RTMDet. Our models were implemented using the MMDetection and MMRotate open-source libraries, which are two OD toolboxes based on the PyTorch framework.

C. Ablation Studies

In this section, we conduct a series of experiments on the DOTA dataset to verify the effectiveness of our proposed method. All ablation experiments are performed using single-scale training and testing.

Evaluation of baseline performance on I^{LR}: To demonstrate the impact of resolution degradation, we conducted experiments on the DOTA dataset using rRTMDet series models trained and tested directly on I^{LR}. The detection performance of each model size is presented in Table I. We observe that the detection performance decreases as the model size decreases. The small-sized model rRTMDet-tiny shows the greatest decrease in detection performance with up to 6.43% mAP reduction, while the large-sized model rRTMDet-X only shows a 1.53% mAP reduction. We analyze that this is due to large-sized models having more channels and model parameters, which facilitate the identification of small object features compared to compact models. Overall, the performance of all models decreases significantly on I^{LR} compared to the performance on I^{HR}. We believe that the decrease in detection performance is primarily attributed to the abundance of small objects in the resolution-degraded image, as demonstrated in Fig. 2. In addition, we note that compared with performing detection on I^{HR}, direct detection on I^{LR} has less computational complexity, and the inference speed has been significantly improved, as shown in Table X. According to our analysis, this significant inference acceleration is brought about by a smaller input image, because the parameter of the model has not been reduced. Thus, it is important to investigate methods to improve the detection performance on I^{LR}. We choose these models as our baselines and use the rRTMDet-tiny model for subsequent ablation experiments.

Evaluation on PSRB: We attempted to enhance the original rRTMDet by adding a PSRB directly, utilizing EDSR as our SR network. Rather than using our proposed FEM, we incorporated the method from [22] to merge C2 and C4 feature maps as the input of the PSRB, while also conducting experiments with various backbone output feature maps as input. The outcomes are demonstrated in Table II. We discovered that using high-level feature maps or combining multiple feature maps occasionally reduced performance, which we believe is due to the absence of low-level information in high-level feature maps and the need for excessive SR multiples ratios for the EDSR. For example, C2's size corresponds to an ×8 downsampling of the input image, and our EDSR requires completing an ×2 SR compared with the input image size, which necessitates an ×16 upsampling

TABLE VI
Comparisons With Other SOTA SR+OD Methods on the DOTA I^{LR}. $\mathbf{AP}_S$, $\mathbf{AP}_M$, and $\mathbf{AP}_L$ are Evaluated on the DOTA's Val Dataset Based on COCO Metrics. MAP Metric is Evaluated on the DOTA Online Evaluation Server. And All Results are Reported on Single-Scale Training and Testing. The Best Results in Each Metric are Highlighted in Bold

Method	**Backbone**	**AP$_S$(%)**	**AP$_M$(%)**	**AP$_L$(%)**	**mAP(%) on test set**
EESRGAN+FasterRCNN [17]	ResNet50	41.2	73.8	61.5	73.40
EESRGAN+FasterRCNN+MFL [14]	ResNet50	42.6	76.7	63.3	74.81
SRCGAN+RFA+FasterRCNN [18]	ResNet50	40.0	72.1	66.0	72.00
SRCGAN+RFA+YOLOv3 [18]	CSPDarkNet53	37.6	54.1	55.9	66.32
SuperYOLO [22]	CSPDarkNet53	39.8	67.3	63.2	69.75
rRTMDet-tiny	CSPNetXt-tiny	40.0	65.9	65.3	68.93
rRTMDet-S	CSPNetXt-S	44.8	70.4	64.6	71.09
rRTMDet-M	CSPNetXt-M	50.2	73.6	69.3	73.91
rRTMDet-L	CSPNetXt-L	50.0	74.5	**72.5**	75.26
rRTMDet-X	CSPNetXt-X	51.1	**75.5**	71.5	75.78
ESRTMDet-tiny (ours)	CSPNetXt-tiny	49.8	67.8	59.9	72.68
ESRTMDet-S (ours)	CSPNetXt-S	49.7	67.0	59.9	73.10
ESRTMDet-M (ours)	CSPNetXt-M	53.5	70.4	65.3	75.72
ESRTMDet-L (ours)	CSPNetXt-L	56.1	72.2	65.0	76.96
ESRTMDet-X (ours)	CSPNetXt-X	**57.2**	72.4	65.5	**77.11**

TABLE VII
Detail Comparisons With Baselines on the DOTA I^{LR}. MAP Metric is Evaluated on the DOTA Online Evaluation Server. And All Results are Reported on Single-Scale Training and Testing. The Best Results in Each Category are Highlighted in Bold

Method	**PL**	**BD**	**BR**	**GTF**	**SV**	**LV**	**SH**	**TC**	**BC**	**ST**	**SBF**	**RA**	**HA**	**SP**	**HC**	**mAP(%)**
rRTMDet-tiny	87.08	76.69	29.17	70.45	74.50	76.09	79.26	90.86	80.98	80.97	52.53	58.93	65.10	69.67	41.75	68.93
rRTMDet-S	87.02	77.49	35.67	72.90	75.01	77.28	79.63	90.78	81.84	83.60	57.57	61.72	67.57	72.67	45.61	71.09
rRTMDet-M	86.70	81.22	40.52	73.68	74.91	77.37	86.53	90.80	84.30	84.78	57.76	64.75	75.42	77.16	52.78	73.91
rRTMDet-L	88.14	80.59	42.54	75.07	77.39	80.39	87.12	90.86	87.98	85.64	58.91	65.83	76.67	77.19	54.63	75.26
rRTMDet-X	88.33	83.37	45.01	**77.52**	78.61	80.71	87.75	**90.88**	86.38	85.96	**60.07**	65.07	76.93	72.76	57.40	75.78
ESRTMDet-tiny (ours)	**89.50**	78.84	43.23	68.98	79.19	82.31	86.98	90.83	80.54	84.47	54.33	63.02	71.11	71.84	44.96	72.68
ESRTMDet-S (ours)	89.00	76.04	43.23	69.13	**80.41**	82.05	87.97	90.84	83.89	84.82	55.34	65.24	70.63	74.35	43.54	73.10
ESRTMDet-M (ours)	89.05	82.08	43.99	72.45	79.74	83.35	88.50	90.84	86.98	85.53	59.27	65.93	75.50	**79.09**	53.49	75.72
ESRTMDet-L (ours)	89.21	83.31	51.17	74.71	79.41	83.51	**88.52**	90.86	**88.11**	**86.50**	59.28	**67.25**	76.98	77.66	57.84	76.96
ESRTMDet-X (ours)	88.90	**83.45**	**52.46**	74.71	80.15	**83.72**	**88.52**	**90.88**	87.35	86.30	58.43	65.13	**77.47**	77.92	**61.14**	**77.11**

TABLE VIII
Comparisons With Other SOTA SR+OD Methods on the UCAS-AOD I^{LR}. $\mathbf{AP}_S$, $\mathbf{AP}_M$, and $\mathbf{AP}_L$ are Evaluated on the UCAS-AOD's Test Dataset Based on COCO Metrics. MAP-07 Represent VOC2007 Metric. MAP-12 Represent VOC2012 Metric. And All Results are Reported on Single-Scale Training and Testing. The Best Results in Each Metric are Highlighted in Bold

Method	**AP$_S$(%)**	**AP$_M$(%)**	**AP$_L$(%)**	**mAP-07(%)**	**mAP-12(%)**
EESRGAN+FasterRCNN [17]	20.0	48.2	61.0	88.7	93.9
EESRGAN+FasterRCNN+MFL [14]	**20.3**	46.1	60.3	89.0	94.2
SRCGAN+RFA+FasterRCNN [18]	20.0	41.0	**72.1**	88.3	93.6
SRCGAN+RFA+YOLOv3 [18]	15.5	48.1	69.0	82.1	89.1
SuperYOLO [22]	18.0	50.6	69.5	87.5	91.3
rRTMDet-tiny	13.3	41.6	66.2	82.3	85.3
rRTMDet-S	15.5	44.8	69.0	84.0	89.1
rRTMDet-M	17.3	46.2	70.1	83.8	89.2
rRTMDet-L	16.8	47.1	70.1	84.2	90.0
rRTMDet-X	16.6	47.2	70.2	87.4	91.1
ESRTMDet-tiny (ours)	16.4	47.2	65.8	87.7	91.6
ESRTMDet-S (ours)	18.1	49.2	69.5	88.8	93.3
ESRTMDet-M (ours)	19.4	50.0	68.8	89.2	93.8
ESRTMDet-L (ours)	18.5	49.9	69.8	89.2	93.9
ESRTMDet-X (ours)	**20.3**	**51.9**	71.6	**89.5**	**95.0**

in FUM. As a result, higher level feature maps correspond to larger SR multiples ratios. Therefore, in later experiments, we only utilized C0 feature map data in the PSRB. This not only guarantees that comparable performance improvements can be achieved but also prevents the execution of excessively high SR multiples ratios of the EDSR.

In our experiments, we first used the architecture of stacked 16-layer ResBlocks (EDSR-ResBlock) as the original EDSR but found that it significantly increased the training time by approximately three times. To overcome this challenge, we attempted to reduce the number of stacked EDSR-ResBlock. Our experimental results are presented in Fig. 7. We discovered that reducing the number of stacked EDSR-ResBlock did not negatively affect detection performance, in fact, it even improved it. We believe that the excessive stacking of EDSR-ResBlock led to an increase in the number of EDSR parameters, which

TABLE IX
EACH CATEGORY COMPARES WITH BASELINES ON THE UCAS-AOD I^{LR}. AND ALL RESULTS ARE REPORTED ON SINGLE-SCALE TRAINING AND TESTING. THE BEST RESULTS IN EACH CATEGORY ARE HIGHLIGHTED IN BOLD

Method	Car	Plane	mAP-07(%)	mAP-12(%)
rRTMDet-tiny	74.5	90.0	82.3	-
	75.9	94.8	-	85.3
rRTMDet-S	77.9	90.1	84.0	-
	82.6	95.5	-	89.1
rRTMDet-M	77.5	90.2	83.8	-
	82.2	96.3	-	89.2
rRTMDet-L	78.2	90.3	84.2	-
	83.8	96.2	-	90.0
rRTMDet-X	84.6	90.2	87.4	-
	86.0	96.2	-	91.1
ESRTMDet-tiny (ours)	85.2	90.2	87.7	-
	86.9	96.2	-	91.6
ESRTMDet-S (ours)	87.4	90.3	88.8	-
	89.6	96.9	-	93.3
ESRTMDet-M (ours)	88.0	**90.4**	89.2	-
	90.6	97.0	-	93.8
ESRTMDet-L (ours)	88.0	**90.4**	89.2	-
	90.6	97.2	-	93.9
ESRTMDet-X (ours)	**88.8**	90.3	**89.5**	-
	92.6	**97.3**	-	**95.0**

resulted in longer training times required to achieve convergence. In addition, the size of the feature maps was much larger than the typical 64 × 64 sizes used in general SR tasks. Therefore, the model needed to learn more features to complete SR, which in turn required longer training iterations. Hence, when using the same number of training epochs as the detection model, the SR network with fewer stacked EDSR-ResBlock may achieve better performance. As a result, we employed a structure with only four layers of EDSR-ResBlock in our subsequent experiments.

Evaluation on ESRM: The abovementioned Table II experiments show that adding only a PSRB to the baseline has limited impact on detection performance. We believe that simply adding a parallel SR network and using backpropagation algorithms to teach the backbone network to enhance features is insufficient to improve SOD. In our analysis, we note that when using I^{LR} inputs, the feature map size is halved compared to using I^{HR} inputs. In the previous method [17], [18], [19], [50], I^{LR} images were enlarged to the size of I^{HR} and then processed by the detection network to ensure that the I^{LR} and I^{HR} are the same size as the feature map in the network. However, this requires the SR network to participate in the inference stage of the model, making it challenging to achieve real-time inference. Therefore, we adopt a more intuitive scheme, which is to directly upsampling the feature map of the model instead of enlarging the image. Our experiments demonstrate that enlarging the feature maps output by the backbone network results in more significant performance improvements than enlarging the feature maps output by the neck, as shown in Table III. We attribute this to the fact that the feature map output by the backbone network retains more low-level features, and the neck+head network is better suited to learning small object features after upsampling the feature map. But it is more challenging to extract small object features when just using only the head part of the model. Therefore, in our subsequent experiments, we incorporate an embedded feature map upsampling method in the position behind the backbone and before the neck.

The abovementioned experiment demonstrates that the scheme of directly enlarging the feature map can effectively amplify the characteristics of small objects and allow the model to focus more on them. So, then we tested several upsampling methods, as shown in Table IV, including the bicubic interpolation method, the deconvolution [62] method commonly used in semantic segmentation, and the subpixel convolution [57] method mainly applied to SR tasks, as well as our proposed method (see Section III-C). Our proposed ESRM demonstrated the best performance, indicating our ESRM has the strongest feature maps upsampling effect. We analyze that the performance improvement is mainly due to our ESRM method is adds a lightweight residual structure on top of the subpixel convolution method. This modification enables the module to have a consistent architecture with the EDSR while maintaining our lightweight design. As a result, the ESRM module can be easily inserted into detection models without significantly increasing computational burden, as Table X shows, and the ESRM module demonstrates powerful SR performance, making it become an effective feature map upsampling method for improving SOD. Accordingly, in subsequent experiments, we used ESRM as the feature maps upsampling method.

Evaluation on FAL and feature alignment loss: Table V shows that directly combining PSRB and ESRM results in a detection performance of 71.69% mAP. In addition, we propose a feature alignment loss $\mathcal{L}_{\mathrm{AL}}$ in Section III-D to allow for more effective information interaction and joint optimization of the two networks. However, we found that directly using the backbone C0 feature map as input for PSRB limits the flexibility of SR task learning. We analyzed that the reason for this phenomenon is that the features extracted from the lightweight backbone are more inclined to high-level features for detection. Consequently, the ability to extract low-level features for SR tasks is limited. To address this, we added FEM to PSRB and combined it with $\mathcal{L}_{\mathrm{AL}}$, resulting in improved performance, as shown in Table V. To further reduce the training instability caused by the discrepancy of feature distribution between rRTMDet and PSRB, we appended a FAL in the output feature map of PSRB before applying our proposed feature alignment loss. This FAL is a 1x1 convolution layer, as shown in Fig. 4. Combining all of these improvements resulted in a 3.75% mAP improvement over the baseline.

In addition, we conducted ablation experiments on the hyperparameter α, which is used in Algorithm 1 to calculate W_{roi} for $\mathcal{L}_{\mathrm{AL}}$, as shown in Fig. 8. The experimental results demonstrate that the impact of different values of α is negligible as long as it is greater than 1. When α is set to 1, W_{roi} becomes a unit matrix, indicating that the weight is evenly distributed between the foreground and background regions, rendering it ineffective.

Visualization and qualitative analysis: From the perspective of qualitative analysis, we visualize the feature map learned by ESRTMDet-X in Figs. 9 and 10. Fig. 9 illustrates that our proposed $\mathcal{L}_{\mathrm{AL}}$ and FAL can effectively restore more low-level high-frequency information to the upsampling feature map C2'

TABLE X
ESRTMDet Series Model and rRTMDet Series Model Parameters, MACs, FLOPs, FPS, and mAP Comparison. The mAP is the Result of DOTA With Single-Scale Training and Testing. MACs, FLOPs, and FPS are the Results of Inference With 512×512 Image Size Under a Single 3090 Condition

Method	Params(M)	MACs(G)	FLOPs(G)	FPS(/s)	mAP(%)
rRTMDet-tiny (on I^{HR})	4.88	10.2	20.45	94.6	75.36
rRTMDet-S (on I^{HR})	8.86	18.8	37.62	94.7	76.70
rRTMDet-M (on I^{HR})	24.67	49.8	99.76	94.2	78.24
rRTMDet-L (on I^{HR})	52.27	102.1	204.21	94.1	78.56
rRTMDet-X (on I^{HR})	94.79	180.5	361.03	93.9	77.31
rRTMDet-tiny	4.88	2.6	5.11	342.2	68.93
rRTMDet-S	8.86	4.7	9.41	341.3	71.09
rRTMDet-M	24.67	12.5	24.94	340.8	73.91
rRTMDet-L	52.27	25.5	51.05	339.6	75.26
rRTMDet-X	94.79	45.2	90.26	341.4	75.78
ESRTMDet-tiny (ours)	12.75	9.8	19.54	338.9	72.68
ESRTMDet-S (ours)	22.24	17.4	34.9	340.6	73.10
ESRTMDet-M (ours)	53.71	43.2	86.57	327.8	75.72
ESRTMDet-L (ours)	103.16	84.2	168.44	337.6	76.96
ESRTMDet-X (ours)	173.74	143.0	286.05	337.6	77.11

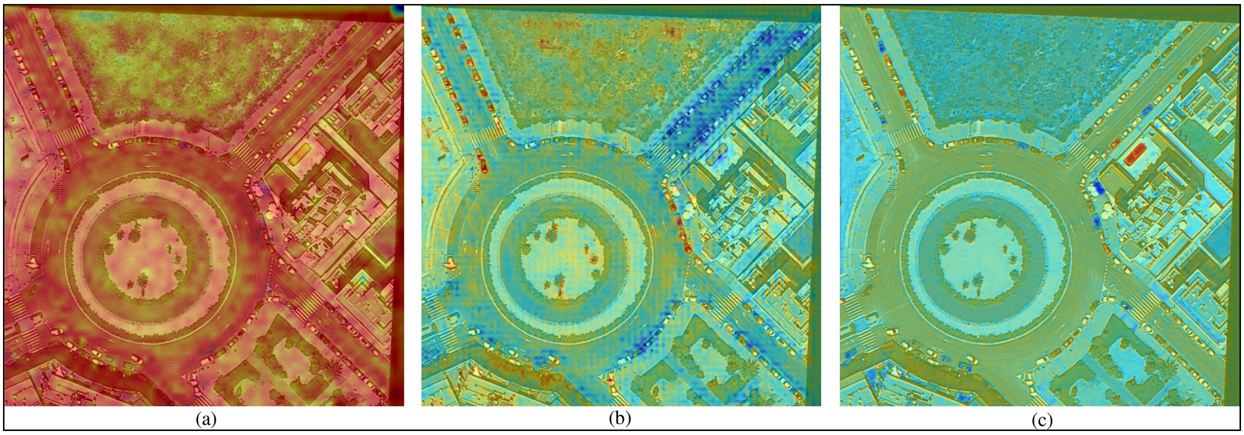

Fig. 9. Visual feature heat-maps of our ESRTMDet-X. ESRTMDet-X uses single-scale training and testing on the DOTA I^{LR}. (a) Visual heat-map of the C2 feature map. (b) Visual heat-map of the C2' feature map which is after our ESRM upsampling process. (c) Visual heat-map of the PSRB output residual feature map.

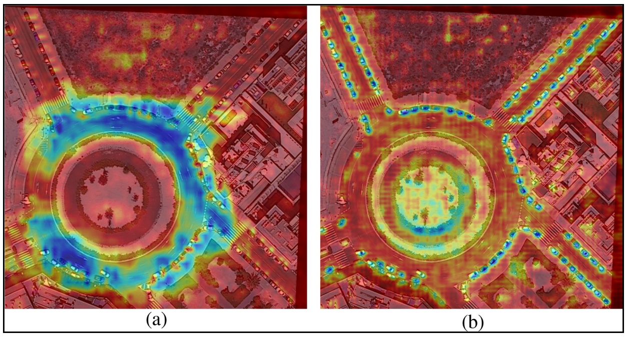

Fig. 10. Comparison between different model's visual heat-map of the P2 feature map. (a) Visual heat-map of the rRTMDet-X's P2 feature map. (b) Visual heat-map of our ESRTMDet-X's P2 feature map.

by comparing the visual heat-map of the C2 feature map [see Fig. 9(a)] with the visual heat-map of the C2' feature map [see Fig. 9(b)] after ESRM. These low-level high-frequency features exist in the PSRB's FDM features map [see Fig. 9(c)]. Similarly, Fig. 10 compares the visual heat-map of the baseline model neck P2 feature map [see Fig. 10(a)] with the heat-map of the ESRTMDet neck P2 feature map [see Fig. 10(b)] and shows that our model can more effectively focus on small objects. Thus, compared to the baseline model, our model's detection accuracy of small objects has significantly improved, as shown in Tables VI and VIII.

D. Comparision With SOTA

In this section, we compare our proposed ESRMDet with other SOTA methods on two challenging aerial detection datasets, i.e., DOTA, and UCAS-AOD.

Results on DOTA: In Table VI, we compare the performance of our ESRTMDet series method with other SOTA SR+OD methods on DOTA task 1 (i.e., rotated detection task). As the test set annotations for DOTA are not available, we use the evaluation metrics of the COCO dataset to assess the detection accuracy of small ($\mathbf{AP}_S$), medium ($\mathbf{AP}_M$), and large ($\mathbf{AP}_L$) objects on the DOTA validation dataset. Our ESRTMDet-X model achieves the highest accuracy in detecting small objects, and the $\mathbf{AP}_S$ of our

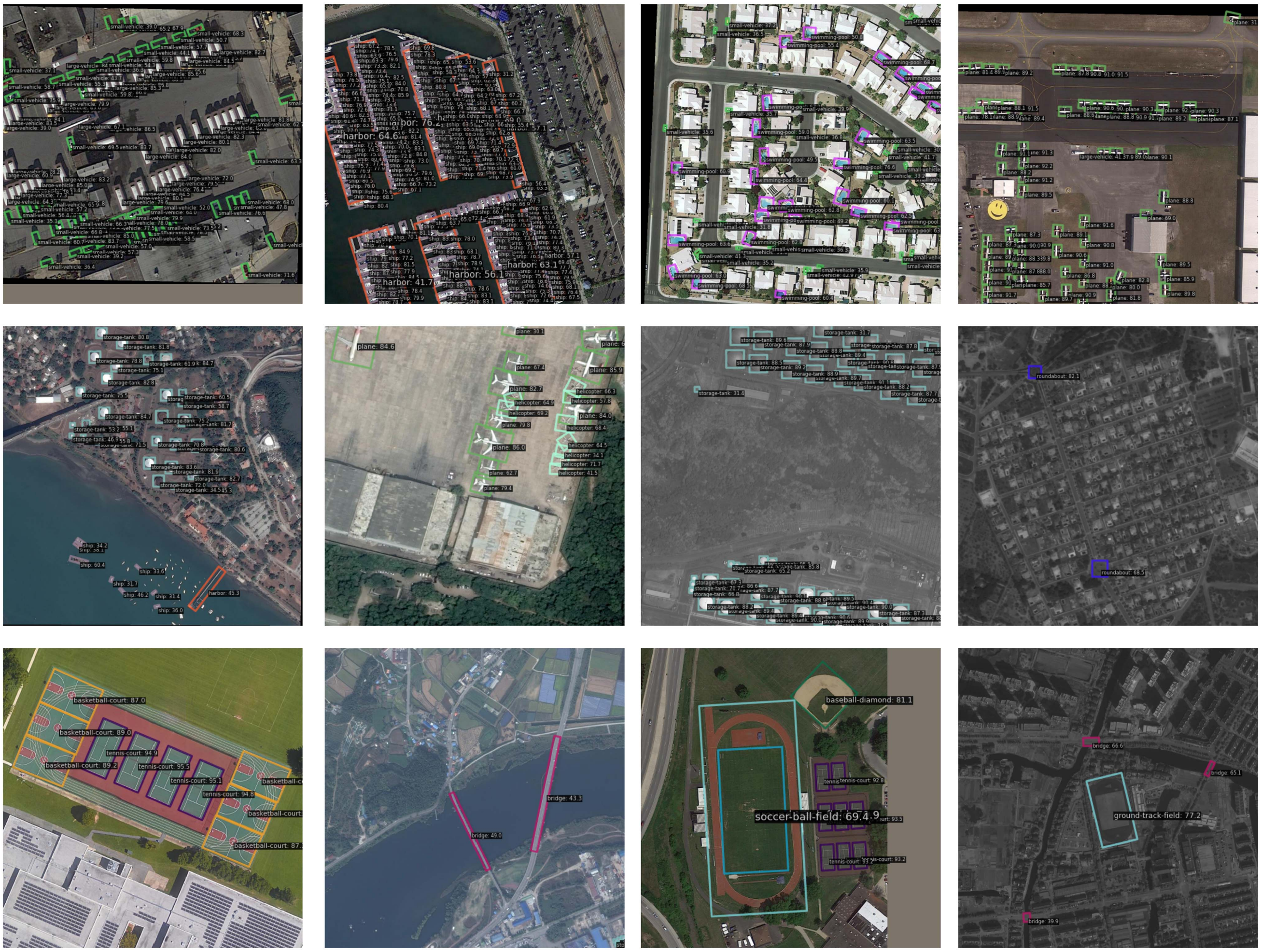

Fig. 11. Some detection results of our proposed ESRTMDet-X by the single-scale training and testing on the DOTA I^{LR} dataset. The confidence threshold is set to 0.05 when visualizing these results, and one color stands for one object class.

ESRTMDet series models surpasses that of their corresponding baseline models. These results confirm the effectiveness of our proposed method.

In addition, we achieved the new SOTA performance of 77.11% mAP through our ESRTMDet-X model. This performance is comparable to that of many famous anchor-based two-stage and one-stage methods on the original DOTA. As shown in Table VII, our method significantly improves the detection accuracy of small objects, especially in the SV, LV, SH, and HC categories. Qualitative detection results of our proposed ESRTMDet are shown in Fig. 11.

Results on UCAS-AOD: The UCAS-AOD dataset contains a large number of small objects, which are often overwhelmed by complex surrounding scenes in aerial images. To comprehensively compare our method with other SOTA SR+OD methods, we use the VOC2007 and VOC2012 metrics to evaluate detection performance. In addition, we followed the same evaluation metrics as in the DOTA experiments and used the COCO evaluation metrics on the UCAS-AOD test dataset to obtain the detection accuracy of small ($\mathbf{AP}_S$), medium ($\mathbf{AP}_M$), and large ($\mathbf{AP}_L$) objects. As shown in Table VIII, our ESRTMDet-X model outperforms other methods with mAP values of 95.0% and 89.5% for VOC2012 and VOC2007 metrics, respectively. These results demonstrate the superiority of our proposed method, particularly on the $\mathbf{AP}_S$ and $\mathbf{AP}_M$ metrics. The detection accuracy of each category on UCAS-AOD is presented in Table IX. We also visualize the results of vehicle and airplane detection in Fig. 12.

V. Discussion

We chose the most advanced ROD model rRTMDet as our baseline, which can achieve satisfactory results on degraded images without using any SR enhancement. The baseline outperforms several SR+OD methods based on FasterRCNN, as Tables VI and VIII show. Combining the rRTMDet baseline with our proposed SR enhancement method resulted in a performance close to directly using the I^{HR} as input. According to Tables VII and IX, all models achieved a performance improvement of about 2% mAP compared to their corresponding baseline models. The performance improvement for small objects was more significant, confirming the efficacy of our proposed SR enhancement method.

Using the I^{LR} as the input image allowed our ESRTMDet to achieve faster inference speeds (using FPS for quantification), smaller computational burdens (using FLOPs quantification),

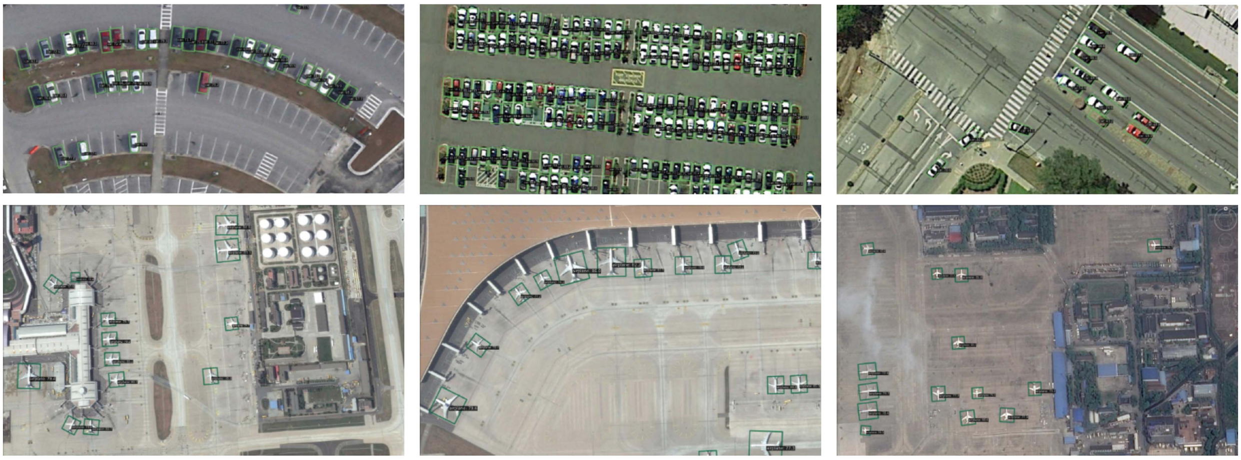

Fig. 12. Examples of detection results on the UCAS-AOD dataset using our proposed ESRTMDet-X.

and lower computational complexity (using MACs quantification) compared to the baseline that used I^{HR} as input, as shown in Table X. Our models add only a small number of additional parameters (using Params for quantification) but achieve better detection performance (achieving 77.11% mAP) and retain impressive inference speed (achieving 330+ FPS). Our method's inference speed has significantly exceeded many model cropping and compression methods that are directly executed on I^{HR}. Thanks to our design of a lightweight ESRM and the PSRB not participating in model inference, our method adds minimal parameters and computational burden, as shown in Table X. These results confirm the efficiency of our methods. Our proposed method achieves real-time inference on our device, making it suitable for deployment on actual drone platforms in the future. From the perspective of macromodel architecture, our method can be regarded as a form of knowledge distillation (KD) between heterogeneous tasks, which differs from the traditional distillation approach. The traditional KD method [63] aims to train a compact model using a large model's knowledge and ensure that the detection performance of the small model is comparable to that of the large model. In contrast, our objective is to leverage an SR model's knowledge to enhance the OD model's performances, which are two distinct tasks. We employ an end-to-end joint optimization training pipeline and do not require a pretrained SR model. Instead, we allow the SR model to learn useful information flexibly through our training process (Algorithm 2) and optimize it in conjunction with the detection model.

In this study, we only investigate the most representative ×2 resolution degradation issues. However, more severe resolution degradation can be addressed by increasing the upsampling multiple in our ESRM and the SR multiples ratios in our PSRB. Nonetheless, we have yet to explore the impact of more severe blur and irregular noise, which we intend to study in future work.

In future research, we plan to explore the combination of traditional KD methods, which jointly utilize the large-size model of the detection task and the SR model to enhance the detection ability of a compact model. In addition, we will consider multitask optimization and adopt a more appropriate method to optimize these heterogeneous tasks simultaneously. Furthermore, inspired by the UIU-Net [51], we can convert the problem of rotated SOD into one of semantic segmentation. Through this problem conversion enables us to utilize the most advanced foundation models for computer vision, such as the segment anything model [64], to effectively solve the rotated SOD problem. This technique is also a promising direction for our future research.

VI. Conclusion

In this article, we propose ESRTMDet, an end-to-end real-time object detector for degraded aerial images that incorporates SR techniques. We enhance the baseline using the PSRB and ESRM models and employ feature alignment loss and FAL to enable interaction between the different tasks. We extensively evaluate our method on two challenging aerial OD benchmarks. Our ESRTMDet-X model achieves a remarkable 77.11% mAP and an impressive 337 FPS, which not only outperforms other SR+OD methods in terms of detection accuracy but also achieves the best inference speed.

In future work, we plan to enhance the detection performance of compact models, such as ESRTMDet-tiny or ESRTMDet-S by combining traditional KD methods, with the goal of deploying these models in actual UAV systems. In addition, we will also investigate the impact of more severe aerial image degradation to further improve the robustness of our model.

References

[1] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5966–5978, Jul. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9170817/

[2] D. Hong et al., "More diverse means better: Multimodal deep learning meets remote-sensing imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 4340–4354, May 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9174822/

[3] G.-S. Xia et al., "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3974–3983.

[4] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning RoI transformer for oriented object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 2844–2853.

[5] Y. Xu et al., "Gliding vertex on the horizontal bounding box for multi-oriented object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 4, pp. 1452–1459, Apr. 2021.

[6] J. Han, J. Ding, J. Li, and G.-S. Xia, "Align deep features for oriented object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–11, 2021.

[7] X. Xie, G. Cheng, J. Wang, X. Yao, and J. Han, "Oriented R-CNN for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3520–3529.

[8] F. Liu, R. Chen, J. Zhang, K. Xing, H. Liu, and J. Qin, "R2YOLOX: A lightweight refined anchor-free rotated detector for object detection in aerial images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5632715.

[9] C. Lyu et al., "RTMDet: An empirical study of designing real-time object detectors," *arXiv:2212.07784*.

[10] Y. Gong, Y. Li, and H. Zhu, "Restoration algorithm of blurred UAV aerial image based on generative adversarial network," in *Proc. 40th Chin. Control Conf.*, 2021, pp. 7201–7206.

[11] L.-J. Wang and W.-S. Hsieh, "Toward an improvement of UAV-aerial image using non-linear image enhancement," in *Proc. 32nd Int. Conf. Adv. Inf. Netw. Appl. Workshops*, 2018, pp. 623–626.

[12] J. Wang, Y. Ye, L. Shen, Z. Li, and S. Wu, "Research on relationship between remote sensing image quality and performance of interest point detection," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2015, pp. 545–548.

[13] Y. Peng, S. Sun, Z. Wang, Y. Pan, and R. Li, "Robust semantic segmentation by dense fusion network on blurred VHR remote sensing images," in *Proc. 6th Int. Conf. Big Data Inf. Analytics*, 2020, pp. 142–145.

[14] J. Yang, K. Fu, Y. Wu, W. Diao, W. Dai, and X. Sun, "Mutual-feed learning for super-resolution and object detection in degraded aerial imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Aug. 2022, Art. no. 5628016. [Online]. Available: https://ieeexplore.ieee.org/document/9854813/

[15] R. Li and Y. Shen, "YOLOSR-IST: A deep learning method for small target detection in infrared remote sensing images based on super-resolution and YOLO," *Signal Process.*, vol. 208, Art. no. 108962. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0165168423000361

[16] V. Magoulianitis, D. Ataloglou, A. Dimou, D. Zarpalas, and P. Daras, "Does deep super-resolution enhance UAV detection," in *Proc. IEEE 16th Int. Conf. Adv. Video Signal Based Surveill.*, 2019, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8909865/

[17] J. Rabbi, N. Ray, M. Schubert, S. Chowdhury, and D. Chao, "Small-object detection in remote sensing images with end-to-end edge-enhanced, GAN and object detector network," *Remote Sens.*, vol. 12, no. 9, 2020, Art. no. 1432. [Online]. Available: https://www.mdpi.com/2072-4292/12/9/1432

[18] S. M. A. Bashir and Y. Wang, "Small object detection in remote sensing images with residual feature aggregation-based super-resolution and object detector network," *Remote Sens.*, vol. 13, no. 9, 2021, Art. no. 1854. [Online]. Available: https://www.mdpi.com/2072-4292/13/9/1854

[19] Y. Wang et al., "Remote sensing image super-resolution and object detection: Benchmark and state of the art," *Expert Syst. Appl.*, vol. 197, 2022, Art. no. 116793. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417422002524

[20] Y. Bai, Y. Zhang, M. Ding, and B. Ghanem, "SOD-MTGAN: Small object detection via multi-task generative adversarial network," in *Proc. Eur. Conf. Comput. Vis.*, V. Ferrari, M. C. Hebert Sminchisescu, and Y. Weiss, Eds. Springer International Publishing, 2018, pp. 210–226. [Online]. Available: http://link.springer.com/10.1007/978-3-030-01261-8_13

[21] L. Qi et al., "Multi-scale aligned distillation for low-resolution detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14443–14453.

[22] J. Zhang, J. Lei, W. Xie, Z. Fang, Y. Li, and Q. Du, "SuperYOLO: Super resolution assisted object detection in multimodal remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–15, 2023.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Adv. Neural Inf. Proces. Syst.*, vol. 28, 2015.

[24] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[25] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," in *Computer Vision and Pattern Recognition*, vol. 1804, Berlin, Heidelberg, Germany: Springer, 2018, pp. 1–6.

[26] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9627–9636.

[27] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "RepPoints: Point set representation for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, Korea, 2019, pp. 9656–9665.

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

[29] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 734–750.

[30] X. Wu, D. Hong, J. Tian, J. Chanussot, W. Li, and R. Tao, "ORSIm detector: A novel object detection framework in optical remote sensing imagery using spatial-frequency channel features," *IEEE Trans. Geosc. Remote Sens.*, vol. 57, no. 7, pp. 5146–5158, Jul. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8654203/

[31] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, pp. 137–154, 2004.

[32] J. Ma et al., "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3111–3122, Nov. 2018.

[33] X. Yang, J. Yan, Z. Feng, and T. He, "R3Det: Refined single-stage detector with feature refinement for rotating object," *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 4, pp. 3163–3171, 2021.

[34] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021. *arXiv:2107.08430*.

[35] X. Pan et al., "Dynamic refinement network for oriented and densely packed object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11204–11213.

[36] X. He, S. Ma, L. He, F. Zhang, X. Liu, and L. Ru, "AROA: Attention refinement one-stage anchor-free detector for objects in remote sensing imagery," in *Image and Graphics*, Y. Peng, S.-M. Hu, M. Gabbouj, K. Zhou, M. Elad, and K. Xu, Eds. Berlin, Germany: Springer, 2021, pp. 269–279.

[37] G. Cheng et al., "Anchor-free oriented proposal generator for object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–11, 2022.

[38] W. Li and J. Zhu, "Oriented RepPoints for aerial object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 1829–1838.

[39] X. Yang, J. Yan, Q. Ming, W. Wang, X. Zhang, and Q. Tian, "Rethinking rotated object detection with Gaussian Wasserstein distance loss," in *Int. Conf. Mach. Learn.*, 2021, pp. 11830–11841.

[40] X. Yang et al., "Learning high-precision bounding box for rotated object detection via kullback-leibler divergence," *Adv. Neural Inf. Proc. Syst.*, vol. 34, pp. 18381–18394, 2021.

[41] S. M. A. Bashir, Y. Wang, and M. Khan, "A comprehensive review of deep learning- based single image super-resolution," *PeerJ. Comput. Sci.*, vol. 7, p. e621, 2021.

[42] H. Chen, X. He, L. Qing, Y. Wu, C. Ren, and C. Zhu, "Real-world single image super-resolution: A brief review," *Inf. Fusion*, vol. 79, pp. 124–145, 2022.

[43] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Comput. Vis. 13th Eur. Conf.*, Zurich, Switzerland, Sep. 2014, pp. 184–199.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[45] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 136–144.

[46] C. Ledig et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 105–114.

[47] X. Wang et al., "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2018.

[48] G. Chen et al., "A survey of the four pillars for small object detection: Multiscale representation, contextual information, super-resolution, and region proposal." *IEEE Trans. Syst., Man, Cybern.: Systems*, vol. 52, no. 2, pp. 936–953, 2020.

[49] J. Shermeyer and A. Van Etten, "The effects of super-resolution on object detection performance in satellite imagery," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 1432–1441.

[50] L. Courtrai, M.-T. Pham, and S. Lefevre, "Small object detection in remote sensing images based on super-resolution with auxiliary generative adversarial networks," *Remote Sens.*, vol. 12, no. 19, 2020, Art. no. 3152.

[51] X. Wu, D. Hong, and J. Chanussot, "UIU-Net: U-net in U-net for infrared small object detection," *IEEE Trans. Image Process.*, vol. 32, pp. 364–376, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9989433/

[52] A. Vaswani et al., “Attention is all you need,” *Adv. Neural Inf. Proc. Syst.*, vol. 30, 2017.

[53] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv:2010.11929*.

[54] Z. Liu et al., “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.

[55] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A ConvNet for the 2020s,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11966–11976.

[56] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, “CSPNet: A new backbone that can enhance learning capability of CNN,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 1571–1580. [Online]. Available: https://ieeexplore.ieee.org/document/9150780/

[57] W. Shi et al., “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1874–1883. [Online]. Available: https://ieeexplore.ieee.org/document/7780576/

[58] L. Wang, D. Li, Y. Zhu, L. Tian, and Y. Shan, “Dual super-resolution learning for semantic segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3773–3782. [Online]. Available: https://ieeexplore.ieee.org/document/9157434/

[59] X. Li et al., “Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 21002–21012.

[60] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized intersection over union: A metric and a loss for bounding box regression,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 658–666.

[61] H. Zhu, X. Chen, W. Dai, K. Fu, Q. Ye, and J. Jiao, “Orientation robust object detection in aerial images using deep convolutional neural network,” in *Proc. IEEE Int. Conf. Image Process.*, 2015, pp. 3735–3739.

[62] H. Noh, S. Hong, and B. Han, “Learning deconvolution network for semantic segmentation,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1520–1528. [Online]. Available: https://ieeexplore.ieee.org/document/7410535/

[63] W. Cao, Y. Zhang, J. Gao, A. Cheng, K. Cheng, and J. Cheng, “PKD: General distillation framework for object detectors via pearson correlation coefficient,” in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=Q9dj3MzY1o7.

[64] A. Kirillov et al., “Segment anything,” *arXiv:2304.02643*.

Fei Liu received the B.S. degree in aircraft design and engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2018. He is currently working toward the Ph.D. degree in instrument science and technology with the State Key Laboratory of Mechanics and Control for Aerospace Structures, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics.

His research interests include object detection, aerial image processing, and remote sensing.

Renwen Chen received the Ph.D. degree in measuring and testing technologies and instruments from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1999.

He was a Visiting Professor with the University of California, Berkeley, CA, USA. He is a Full Professor in intelligent monitoring and control with the State Key Laboratory of Mechanics and Control for Aerospace Structures, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics. His research interests include energy harvesting, structural health monitoring, and fault diagnosis, contactless signal and power transmission, computer measurement and control technology, image processing, and machine vision.

Dr. Chen is a Member of the AVIC measurement and control technology development center, and a Member of the online monitoring committee of the Jiangsu instrumentation society.

Junyi Zhang received the B.S. degree in aircraft design and engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2020. He is currently working toward the Ph.D. degree in instrument science and technology with the State Key Laboratory of Mechanics and Control for Aerospace Structures, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics.

His research interests include image processing and machine vision.

Shanshan Ding received the B.S. degree in automation from Wanjiang Institute of Technology, Ma’anshan, China, in 2017, and the M.Sc. degree in control engineering from Anhui University of Technology, Ma’anshan, China, in 2019. He is currently working toward the Ph.D. degree in instrument science and technology with the State Key Laboratory of Mechanics and Control for Aerospace Structures, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China.

His research focusses on the fault diagnosis of aeromechanical products and image processing.

Hao Liu received the B.S. degree in electrical engineering and automation and the M.S. degree in electronics and communication engineering from North University of China, Taiyuan, China, in 2017 and 2021, respectively. He is currently working toward the Ph.D. degree in instrument science and technology with the State Key Laboratory of Mechanics and Control for Aerospace Structures, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China.

His research interests include optimization theory, wireless sensor networks, and image processing.

Shaofei Ma received the B.S. degree in aircraft design and engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2018. He is currently working toward the master’s degree in mechanics with the State Key Laboratory of Mechanics and Control for Aerospace Structures, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics.

His research interests include energy harvesting, piezoelectric elements, acoustic black hole, and image processing.

Kailing Xing received the B.S. and M.S. degrees in electrical engineering and automation from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2018 and 2022, respectively.

She is a Hardware Engineer with Hardware Development Dept. IV, Wired Product R&D Institute, System Product Wired Product Operation Division, ZTE Corporation, Nanjing, China. Her research interests include control algorithms, FPGA development, and image processing.