Small-Object Detection in Remote Sensing Images With Super-Resolution Perception

Jiahang Liu, *Member, IEEE*, Jinlong Zhang, Yue Ni, Weijian Chi, and Zitong Qi

***Abstract*—Small objects are widely distributed on remote sensing images (RSIs), and most of them are achieved by super-resolution (SR) reconstruction followed by detection. However, due to the independent training of the SR network and the detection network, the lack of interaction between them leads to the limited performance of small object detection (SOD). Furthermore, time accountability is increased since the SR task is performed before the detection task. To address these problems, we develop a new SOD network to improve the SOD performance in RSIs, which embeds the SR task into the SOD task. First, a channel attention weighting module is proposed before the backbone to assign weights to different channels of the input image, allowing the network to selectively focus on different channels. Second, a self-attention encoding module is designed between the backbone and neck to add self-attention weight to the features extracted from the backbone and enhance the feature representation ability of small objects. Finally, the SR perceptual branch and perceptual loss are designed so that the SR task and the detection task can be associated through the SR perceptual loss, and the SR perceptual branch can guide the backbone network to learn high-resolution features through joint training, thus improving the detection performance of small objects. In the inference phase, the SR perceptual branch has been removed to improve the speed. Extensive experimental results on VEDAI and DOTA datasets show that the proposed method achieves an accuracy of 82.63% and 78.63%.**

***Index Terms*—Attention mechanism, remote sensing images (RSIs), small object detection (SOD), super-resolution (SR), YOLOV5.**

I. Introduction

REMOTE sensing image (RSI) object detection refers to obtaining the position and class of objects of interest from RSIs [1]. RSI object detection is widely used in search and rescue, automatic driving, defect detection, UAV observation [2], and intelligent traffic monitoring [3]. Therefore, RSI object detection is a very hot research issue.

With the development of deep learning technology in recent years, RSI object detection based on deep learning shows powerful advantages. Such as Cheng et al. [4] proposed a method to learn a rotationally invariant convolutional neural network (CNN) model, which can significantly improve the performance of object detection by the inclusion of a rotationally immune layer. Long et al. [5] proposed a new framework for object localization in RSIs, which includes three main processes that enable the precise positioning of objects in RSIs. Dong et al. [6] proposed a gated context-aware module, which was embedded into a feature pyramid network (FPN) to enable the FPN to effectively detect objects in RSIs. Ye et al. [7] proposed a convolutional network model with an adaptive attention fusion mechanism. The model can effectively realize object detection in RSIs.

All the abovementioned methods can achieve good results for object detection in RSIs. However, due to the long imaging distance, the large field of view, and the relatively low resolution (LR) of RSIs, a large number of small objects (less than 32 × 32 pixels [8]) are present in RSIs. Although there has been important advancement in object detection based on deep learning, these methods are typically designed for medium or large objects. Compared with medium and large objects, small objects are difficult to have rich features because they contain fewer pixels. In addition, due to the downsampling operation of the deep neural network, as the network deepens, the deep features contain less and less useful information about small objects. The abovementioned problems make it very difficult for deep neural networks to obtain the available information about small objects. Therefore, using common object detection based on deep learning for small object detection (SOD) cannot achieve satisfactory results.

The most popular approach for SOD in RSIs is to use super-resolution (SR) reconstruction combined with object detection. One approach is to utilize a SR reconstruction network to recover a high-resolution (HR) image containing texture and detail features from a LR image before the SOD network. This HR image can then be utilized for SOD. It is an effective method for improving the accuracy of SOD. For example, Rabbi et al. [9] proposed the edge-enhanced SR generative adversarial network (EESRGAN). However, this method has some properties, such as large parameters, difficulty in network training, and the GAN lack of guidance from the detection network. Another approach is to use the SR network as a parallel branch of the SOD network to guide the detection network by reconstructing the features extracted from the detection network or to directly utilize the reconstructed features for SOD. Liu et al. [10] proposed ESRTMDet which embeds the SR network into the detection network to generate HR features and improve the detection accuracy of small objects. However, these methods do not consider the

Received 19 May 2024; revised 30 June 2024; accepted 19 August 2024. Date of publication 2 September 2024; date of current version 13 September 2024. This work was supported in part by the Innovative Talent Program of Jiangsu under Grant JSSCR2021501, and in part by the High-Level Talent Plan of NUAA, China. *(Corresponding author: Jiahang Liu.)*

The authors are with the College of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: jhliu@nuaa.edu.cn; zjinlong@nuaa.edu.cn; melony1997@nuaa.edu.cn; wj_chi@nuaa.edu.cn; qzt9588@nuaa.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2024.3452707

differences between SR and SOD tasks. As a result, SR and SOD networks cannot be optimized together to improve the detection accuracy of small objects.

This article aims to develop a network for SOD in RSIs using SR techniques. This will address the issues of high model complexity, difficulty in convergence, and poor performance of SOD in RSIs. To achieve high-precision detection of small objects in RSIs, we proposed simple and effective methods. Our approach addresses the shortcomings of the previously mentioned detection network combined with SR methods and achieves SOTA detection performance. The contributions of this article can be summarized as follows.

1) We designed a channel attention weighting module (CWM) to replace the Focus module in YOLOv5 [11]. This module enables the backbone to focus on useful channels, thereby generating more effective features, which significantly improves the SOD performance.
2) We designed a self-attention coding module (SCM). This module enhances the network's capacity to obtain global contextual features and improves the representation of small objects, which in turn enhances the SOD performance.
3) A parallel SR perceptual branch (SRPB) and perceptual loss are presented. This structure guides the backbone of the SOD network to learn HR features and improves the expressiveness of the features extracted by the backbone. The modules in the inference stage are removed and a lightweight detection model is implemented to improve the inference speed.

II. Related Work

This section reviews recent related work from three aspects: object detection based on deep learning, SR techniques, and SOD in RSIs based on SR techniques. Because the proposed method integrates both object detection and SR techniques.

A. Object Detection Based on Deep Learning

In recent years, the task of object detection has reached an unprecedented level of development due to the rapid advancement of CNN technology [12]. There are two categories of object detection algorithms based on deep learning: two-stage and one-stage methods.

Two-stage algorithms first extract many candidate boxes, then process the candidate boxes through operations such as classification, bounding box regression, and bounding box filtering, and finally obtain the detection results. The first two-stage detector is the region proposals CNN (RCNN) [13]. It uses a selective search algorithm for region proposals and passes the proposals to the CNN to generate feature vectors for each proposed region. The final classification task is performed using a support vector machine. FastRCNN [14] was proposed as an improvement to RCNN. It uses deep convolutional networks for efficient classification of object proposals and employs a region of interest (RoI) layer to address the challenges posed by scale variations. However, its speed is slow due to the complex algorithms used for generating region proposals. Later, FasterRCNN [15] was proposed as an improvement to FastRCNN. It introduces a region proposal network to extract candidate boxes, which improves the computational speed. The two-stage algorithms have high accuracy but do not meet speed requirements in real-time detection.

The one-stage detector does not generate region proposal boxes in advance but detects them directly through a process. This greatly improves object detection speed. The YOLO series [16], [17], [18], [19] is a typical one-stage detector that enables direct object detection. This feature simplifies the network structure and accelerates the object detection. Particularly, YOLOv5 introduces the GIoU [20] loss function and uses Adam [21] as an optimization function. These improvements make YOLOv5 faster and more accurate than other YOLO series in detecting densely occluded objects. YOLOv8 [22] is the latest model in the YOLO series. It uses a new structure, convolutional layers, and anchor-free detector heads to further improve the speed of object detection. Other popular one-stage object detection algorithms include the SSD [23] and the RetinaNet [24] algorithm. The SSD algorithm assigns a score to the default boxes on the feature map and modifies them for object detection. However, it is dependent on human experience and requires manual parameter setting for the preselected frames. RetinaNet solves the problem of category imbalance by using the Focal loss function. However, it cannot detect objects in real-time and is less effective at detecting small or multiple objects.

B. SR Technology

To address the SR problem of images, early approaches utilized interpolation techniques based on sampling theory [25]. Tai et al. [26] utilized natural image statistics to reconstruct HR images. Dong et al. [27] were the first to introduce deep learning approaches to SR, and a series of SR methods based on deep learning followed. Kim et al. [28] proposed VDSR, which can process multiple scales of SR in a single network. However, VDSR requires bicubic interpolated images as inputs, resulting in increased computational time and memory usage compared with other up-sampling methods. SRResNet [29], which was put forward later, effectively solved the abovementioned time and memory problems and had good performance. However, it simply adopts the ResNet [30] architecture without significant modifications. It is important to note that the original ResNet is more suited to the resolution of complex computer vision problems, such as image classification and detection. Therefore, it may not be optimal to apply the ResNet directly to simple vision problems such as SR. In 2017, Lim et al. [31] proposed enhanced deep residual networks (EDSR), which remove the BN layer from the ResNet architecture. This is because BN removes the range flexibility in the network and significantly improves the quality of image SR. The method can realize SR with arbitrary multiplicity. The SR problem has been effectively addressed by GAN [32] through deep learning. Enhanced SRGAN [33] has achieved even more significant performance improvement by removing the BN from the generator and designing a residual dense block in place of the normal ResBlock. K. Jiang et al. [34] proposed an edge enhancement network based on a GAN for robust satellite image SR reconstruction. Wang et al. [35] proposed a technique called dual SR learning, which has been demonstrated to effectively improve segmentation accuracy without introducing additional computational cost. In general, the majority of SR methods aim to achieve optimal results by increasing the complexity and parameters of the model. However, such a complex SR network limits the application of SR technology in other fields, such as object detection.

C. SOD in RSIs Based on SR

Small objects are usually difficult to distinguish from other categories in RSIs, which can lead to inaccurate detection of small objects [36]. Data augmentation is an effective method for improving the performance of SOD [37]. It has been proven effective in sampling small objects of interest and uses SR as a preprocessing step for data in the detection task [8]. Shermeyer and Van Etten [38] explored the effect of super-resolution techniques on the performance of object detection algorithms and demonstrated their effectiveness. Courtrai et al. [39] improved the dimensions and details of the objects to be detected by SR to solve the problem of detecting small objects in satellite or aerial RSIs. Bashir and Wang [40] improve SOD performance by enhancing the SR framework in combination with cyclic generative adversarial networks and residual feature aggregation. Ferdous et al. [41] developed a framework for vehicle detection in LR aerial images using SR techniques. Zhang and Ma [42] used a pseudo-label generation approach and weakly supervised learning to learn object detection in RSIs.

The preceding studies have demonstrated that the challenging issue of SOD in LR images can be effectively addressed through the application of the SR. However, the resolution of the input image must be increased by the SR network, which incurs additional computational costs compared with a single detection model. Unlike the previous work, the integration of SR networks as a supplementary approach to improve the SOD performance is more promising. Zhang et al. [43] used a lightweight SR network as a parallel auxiliary network and used the fused features extracted from the YOLOv5 backbone as input to the SR network to restore HR images. However, this method ignores the differences between the backbone features and the HR images, leading to poor detection results. Although it avoids the high computational overhead associated with SR networks, it reduces the detection accuracy.

III. Method

In this section, we will carefully present the proposed method. First, we give an overview of the baseline model. Second, the CWM is introduced, which adds the channel weights of the image before it enters the backbone. Next, we introduce the SCM located between the backbone and neck. This module adds self-attention coding weights to the features extracted from the backbone. Then, we present the SRPB and perception loss we designed. The SRPB can guide the backbone to learn HR features. The perception loss can reduce the difference between the detection task and the SR task. By doing so, the two tasks can promote each other to achieve optimal results. Finally, we present the overall structure of our proposed framework.

TABLE I
Comparison Results of Model Size and Inference Ability in Different Baseline YOLO Frameworks on LR Images of the VEDAI Validation Set

Method	Layers	Params(M)	GFLOPs	mAP0.5(%)
YOLOv5s	224	7.073	16.4	62.4
YOLOv5m	308	21.07	50.4	64.2
YOLOv5l	397	46.64	114.2	65.6
YOLOv5x	476	87.13	217.5	64.0
YOLOv8s	225	7.073	28.5	62.0

A. Baseline Structure

Since the first generation of the YOLO model was put forward, researchers have made several updates and iterations to improve its performance. Although YOLOv8 is the latest version in the YOLO series, it is less stable than other versions. Additionally, experiments in Table I have shown that YOLOv8 does not perform as well as YOLOv5 in terms of SOD, the number of model parameters, and the number of GFLOPs. Therefore, we have chosen YOLOv5 as our basic framework.

Fig. 1 shows the structure of the YOLOv5, which is divided into three parts: the backbone, the neck, and the detection head. The backbone is composed of Convolution-Batch-normalization-SiLu (CBS) [44], cross-stage partial (CSP) [45], and spatial pyramid pooling (SPP) [46] modules. Its main function is to extract shallow texture features and deep semantic features. CSPNet extracts feature information from the backbone, which includes the CBS and CSP modules. The feature maps of the previous layer are assigned to the two branches through the CSP module. One branch is connected to the end of the module, while the other is used as input for the ResNet block or the CBS block. Finally, the two feature maps are connected, and the features are merged before being fed into the CBS block. The CSP can reduce the number of channels by half using a 1×1 convolution to decrease computation. The SPP module comprises parallel max-pooling layers with varying kernel sizes to extract multiscale depth features. The CSP can reduce the number of channels by half using a 1×1 convolution to decrease computation. The SPP module comprises parallel max-pooling layers with varying kernel sizes to extract multiscale depth features.

The neck of YOLOv5 utilizes the FPN [47] and PANet [48] structures to enhance the features extracted by the backbone and address the multiscale problem in object detection. It passes deep semantic features and shallow texture features. The neck is composed of columns of CBS, Up-sample, Concat, and CSP modules. The CBS module used in the neck is the same as the one used in the backbone. The Up-sample module is used to up-sample the feature map by a factor of 2 using the nearest-neighbor up-sample method. The CSP module used in the neck network differs slightly from the one used in the backbone, as shown in Fig. 1. It does not have a Resblock module because it

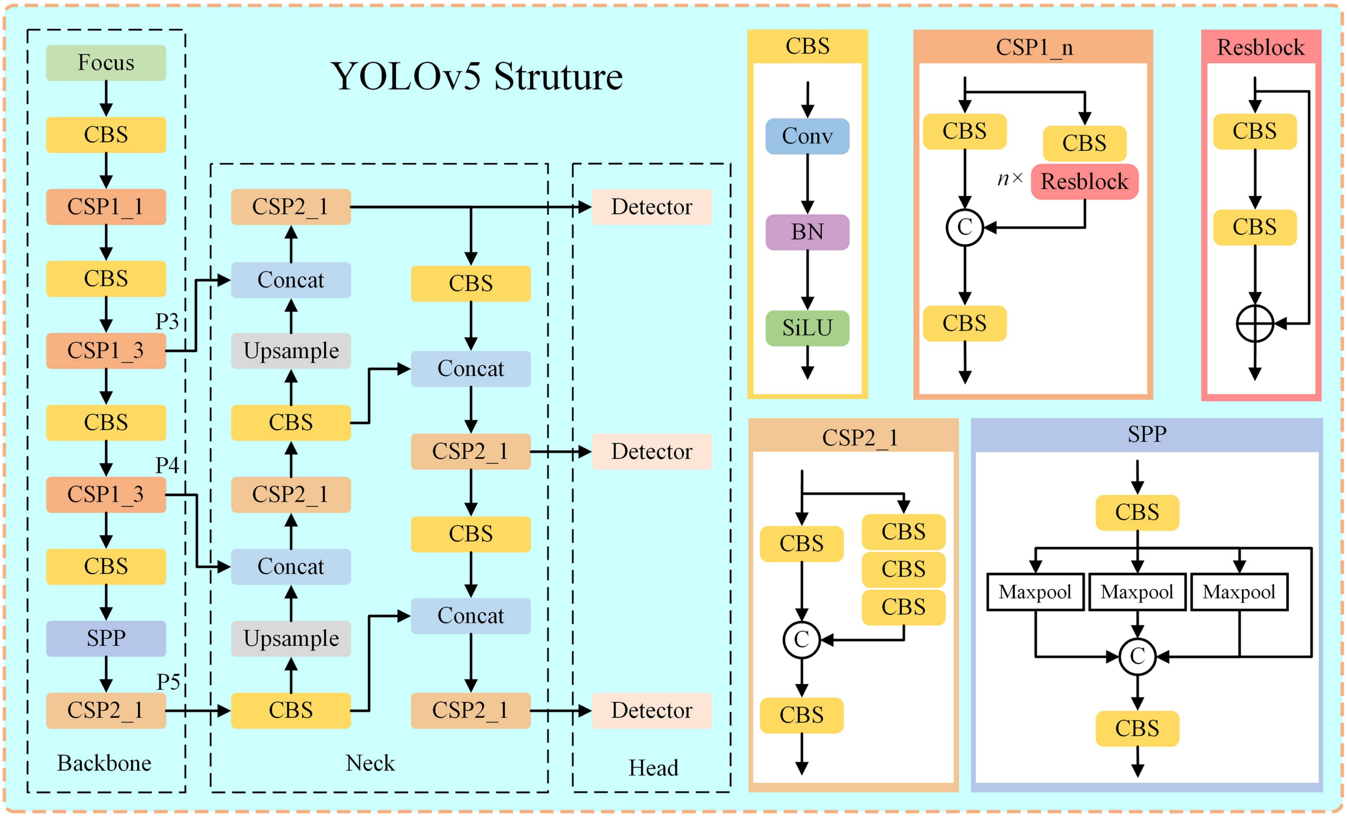

Fig. 1. Structure of YOLOv5. The low-level texture and high-level semantic features are extracted by stacked CSP, CBS, and SPP structures.

is unnecessary to further deepen the network in the neck. It is more appropriate to use the C3 module without residuals. The role of the Concat module is to splice the feature map based on the channel dimension.

The role of the detection head is to finalize the object classification and position regression. YOLOv5 has three detection heads that address the multiscale issue in object detection by using different feature scales in the feature pyramid.

B. Channel Attention Weighting Module

RSIs often include bands beyond red-green-blue (RGB), with the near-infrared (NIR) band being the most used. Research has demonstrated that combining RGB and NIR images considerably enhances SOD performance [43]. Some studies have treated RGB and NIR images as two completely distinct modalities and have developed separate branches for multimodal image fusion [49]. The imaging mechanism is the same for RGB and NIR images, they are both solar radiations reflected from ground objects received by the same sensor. So, we think RGB-NIR image fusion with two different branches for SOD is unlikely to improve performance and may even increase model complexity. To reduce the complexity of the model and enhance the performance of SOD, we treat the NIR image as having the same properties as the RGB image. Consequently, we designed a CWM to facilitate SOD in RGB-NIR images and RGB images. This module applies to both ordinary RGB images and RGB-NIR images, as shown in Fig. 2. It adds channel attention weights to the input images while fusing RGB-NIR images, enabling the network to pay attention to the channel features of the images of different bands and improving the expression ability of the subsequent backbone features. We replaced the Focus module in YOLOv5 with this module, inspired by [43], which significantly improved the detection performance of small objects.

First, the RGB and NIR images are combined into a four-channel image using channel concatenation. If the input image is a standard RGB image, there is no need for a channel concatenation operation. The resulting images are then normalized to the [0, 1] interval before being fed into the first convolutional module to obtain the mask feature m_{input}

$$m_{input} = f\left(X_{input}\right) \tag{1}$$

$f(\boldsymbol{.})$ denotes the convolution of 1×1. The mask features are then multiplied with the input image to add mask weights

$$X_{mask} = X_{input} \otimes m_{input} \tag{2}$$

$\otimes$ denotes element-wise matrix multiplication.

Second, to avoid losing information from the original image, the resulting image must be convolved with the input image before extracting further image features

$$X_{full} = f\left(X_{mask} + X_{input}\right). \tag{3}$$

Finally, the image is fed into the SE module [50]. This module extracts the internal channel features of the image, which are then fed into the backbone. This enables the backbone to better focus on the internal channel features of the image.

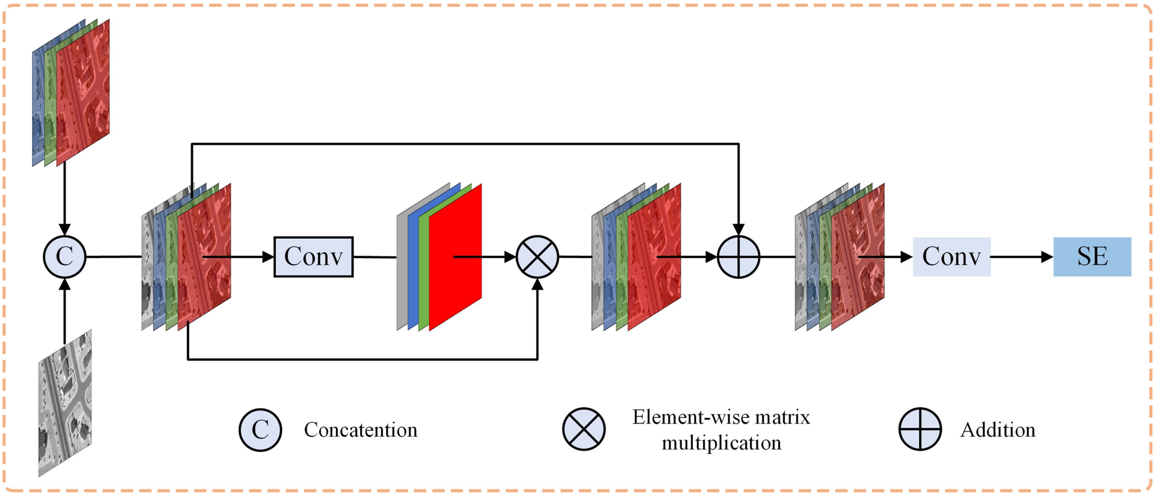

Fig. 2. Architecture of the channel-attention weighting module.

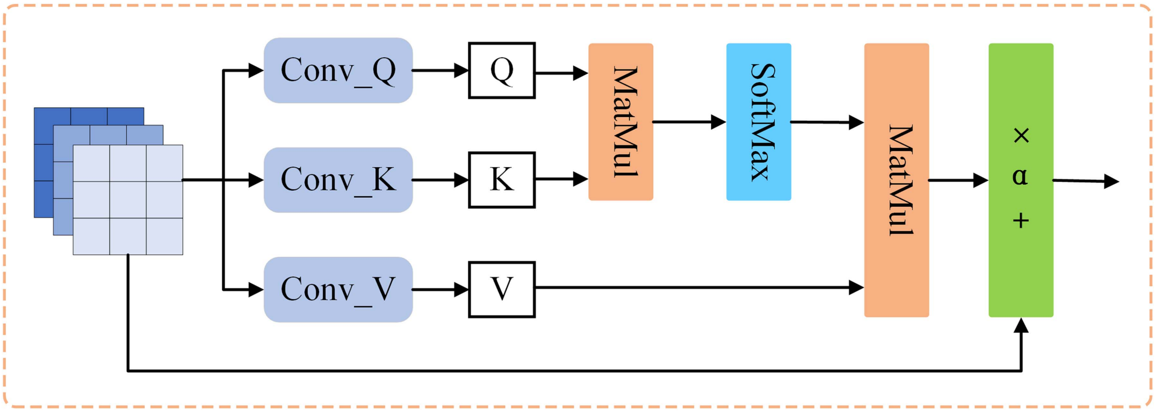

Fig. 3. Principle of self-attention coding module.

As illustrated in Fig. 1, the Focus module within the YOLOv5 backbone reduces the quantity of input data by reducing the spatial dimension of the input image and increasing the channel dimension. Consequently, this process results in the loss of spatial information of small objects. We replace the Focus module in the YOLOv5 backbone with CWM to avoid the degradation of the input image resolution and thus improve the detection of small objects.

C. Self-Attention Coding Module

The YOLOv5 backbone is a stack of convolutional layers. In comparison to medium and large objects, small objects exhibit a faster loss of information as the network deepens. So, the deeper features contain minimal information about the small objects themselves. This is one of the main reasons for the current poor detection of small objects. The optimal use of global information related to small objects is a key to enhance SOD. To enhance the global information of the features extracted by the backbone and the long-range dependency between pixels, an SCM was designed between the backbone and the neck. This module augments the global information in the feature map through interactions between feature image pixels, as shown in Fig. 3.

The YOLOv5 network utilizes a feature map extracted by backbone from P3, P4, and P5 in Fig. 1 for feature fusion and detection. To improve this, we have implemented self-attentive coding for these three feature maps (P3, P4, and P5), denoted by $F_{input} \in \mathbb{R}^{C\times W\times H}$. These three features are obtained through different convolution and reshape operations are defined as follows:

$$F_{Q/K/V} = reshape\left(f_{Q/K/V}\left(F_{input}\right)\right) \tag{4}$$

$f_{Q/K/V}(\cdot)$ represents three different 1×1 convolutions, and $reshape(\cdot)$ represents the reshaped operation on the features.

To obtain the interactions between each pixel in the feature map, $F_Q \in \mathbb{R}^{N\times C}(N = W \times H)$ is matrix multiplied with $F_K \in \mathbb{R}^{C\times N}$, and the SoftMax operation is performed to obtain the weight $F_{attention} \in \mathbb{R}^{N\times N}$. This weight reflects the interactions between each pixel of the feature. It can reflect the global context of the feature.

Finally, $F_{attention}$ and F_V are multiplied and then deformed to obtain the feature map with the added attention weights. The

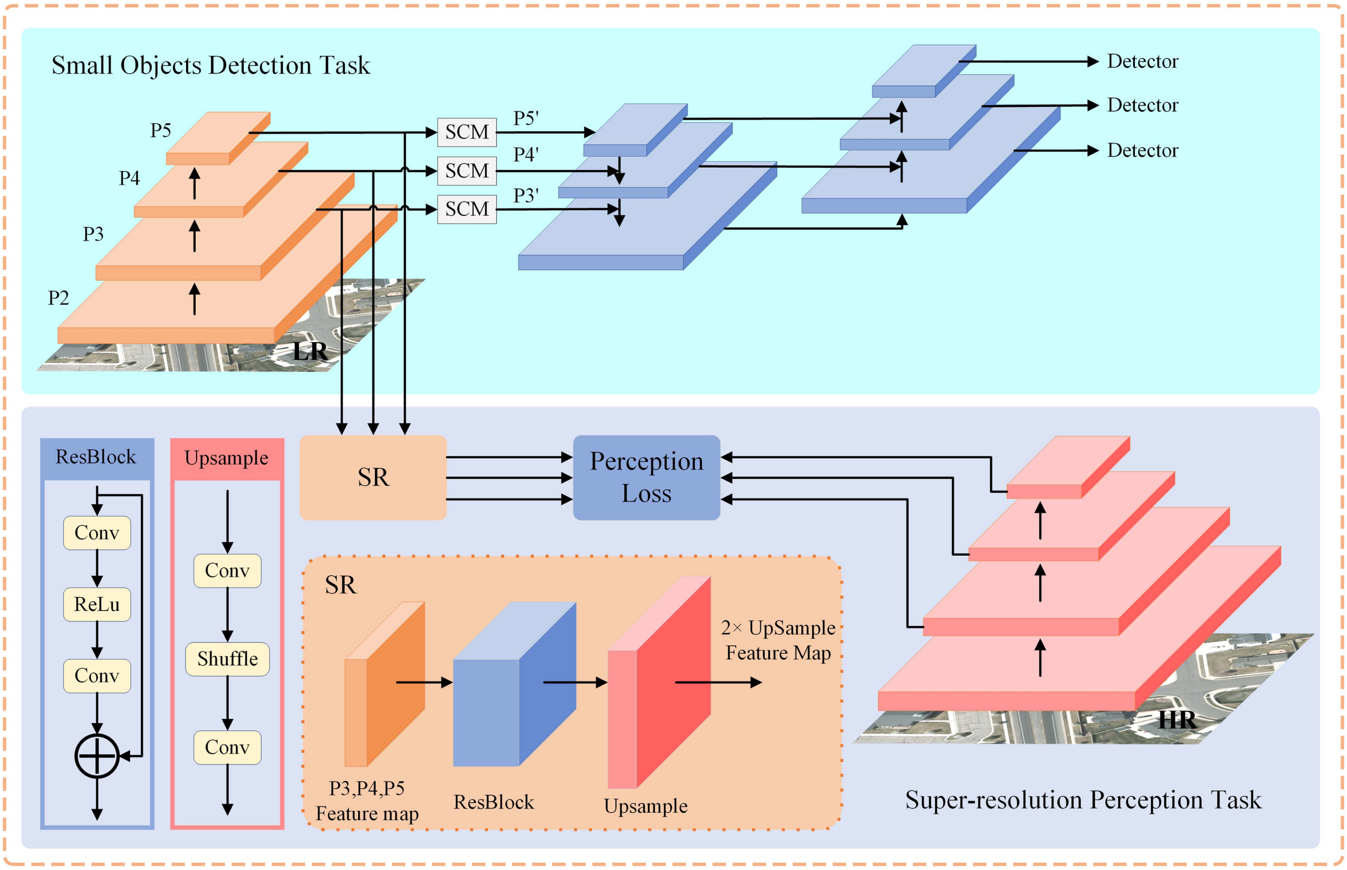

Fig. 4. Overall architecture of our proposed methodology, and the specific composition details of our proposed modules.

output features are

$$F_{output} = \alpha \left(reshape\left(F_{attetion} \times F_V\right)\right) + F_{input}. \quad (5)$$

To enhance the fitting ability of the module, we introduce a learnable parameter α. During the initial training of the network, the output result after this module remains equal to the input result and α changes gradually as the model is trained.

The SCM is designed to improve global context information in the feature map by capturing long-range dependencies between features through the self-attention mechanism. This allows for interaction between each pixel and feature. By enhancing the feature information of small objects through their interaction with surrounding pixels and other small objects, we can improve their feature expression and subsequently enhance the detection performance for small objects with weak information.

D. SR Perception Branch

This section presents the designed SRPB and the overall structure of our proposed framework, illustrated in Fig. 4. The SRPB comprises two main parts: the super-resolution reconstruction module (SRM) and the parallel perception module (PPM).

The primary function of the SRM is to transform the LR features of the backbone into HR features, allowing the backbone to learn HR features. This is comparable to the EDSR structure, which is composed of multiple ResBlock modules that are stacked on top of each other and up-sample convolution. The channel number of SRM was adjusted to ensure that the input LR features and output HR features have the same number of channels. It can reduce the loss of information. The role of up-sample convolution is to up-sample the LR features twice. As a result, our SRM can achieve a twofold SR task.

In the article [43], the HR features output from the SR module and the HR image are computed as a loss. This method ignores the difference between the feature and the image and does not consider the difference between the object detection task and the SR task, which leads to the two different tasks cannot guide each other to achieve the best performance. To connect the SR network and the object detection network, we designed a PPM inspired by the perceptual loss in the field of SR reconstruction. The main role of the PPM is to extract features of interest to the object detection network in HR images. The paper proposes a PPM with the same backbone as the detection network. This same structure ensures that the extracted HR features are very similar to the LR features extracted by the object detection network.

The SRM can transform the LR features extracted by the detection network into HR features. PPM is capable of extracting HR features from HR images. Since the structure of the PPM is the same as the backbone of the detection network, the HR features extracted by the PPM are exactly those needed by the detection network. These two HR features are then used to calculate the perceptual loss for subsequent optimization of the detection network. Using PPM allows the SRM to complete a process from LR features to HR features, avoiding the LR feature to image that would result from the direct use of the SR network. This makes the HR features reconstructed by the SRM more in line with the features needed by the SOD network, which in turn can enable the SOD network to achieve better detection performance. The addition of PPM enables the SRM to better guide the detection network, allowing it to achieve better results.

E. Loss Function

The loss of our proposed network has two main parts: the detection loss L_O and the perception loss L_S, which can be expressed as follows:

$$L_{total} = c_1 L_O + c_2 L_S \tag{6}$$

the coefficients c_1 and c_2 balance the detection and SR perception tasks. The L_O loss [51], which consists of the localization loss L_{box}, classification loss L_{cls}, and confidence loss L_{obj}, which can be expressed as follows:

$$L_O = \lambda_{box} \sum_{l=0}^{2} a_l L_{box} + \lambda_{cls} \sum_{l=0}^{2} b_l L_{cls} + \lambda_{obj} \sum_{l=0}^{2} c_l L_{obj} \tag{7}$$

l represents the feature layer used for the detector, while a_l, b_l, and c_l are the weights of the three loss functions in the different detection layers. Additionally, the weights λ_{box}, λ_{cls} and λ_{obj} are used to adjust the bias of the weights between the three losses.

The loss function L_S can be expressed as follows:

$$L_S = L_{low} + L_{med} + L_{high} \tag{8}$$

the L_S loss comprises three types of feature loss: low-level (L_{low}), medium-level (L_{med}), and high-level (L_{high}). These losses are computed from the L1 loss [51] and correspond to the features of P3, P4, and P5 in after SR reconstruction.

IV. Experimental Results

A. Dataset

The experiments utilized the Vehicle Detection in Aerial Imagery (VEDAI) dataset [52], which consists of cropped images from the Utah Automated Geographic Reference Center dataset. Each image collected from the same altitude has a resolution of approximately 12.5 cm × 12.5 cm per pixel and contains about 16 000 × 16 000 pixels. The images in the same scenes are available in RGB and NIR. The VEDAI dataset contains 1246 images with backgrounds including grass, highways, mountains, and urban areas. All images contain both 1024 × 1024 and 512 × 512 sizes. The objective is to detect 11 classes of vehicles, such as cars, pickups, campers, and trucks.

B. Implementation Details

The proposed framework was implemented using the PyTorch framework and tested on a server with NVIDIA 3090 Ti. We trained our module on the VEDAI dataset, which consisted of 1089 training images and 121 testing images. The annotation for each object in the image contains the coordinates of the center of the bounding box, the orientation of the object relative to the positive x-axis, the coordinates of the four corners, the category ID, a binary flag indicating whether the object is occluded, and another binary flag indicating whether the object is cropped. Categories with less than 50 instances in the dataset, such as airplanes, motorcycles, and buses, were excluded. The annotations of the VEDAI dataset were converted to YOLO format, and we transferred the ID of the interested class to 0, 1, …, 7. During training, the input image size for the detection network is 512 × 512, while for the SRPB, it is 1024 × 1024. During testing, the input image size is consistent with other comparison algorithms, which is 512 × 512. We also utilized data augmentation techniques such as mosaic and flip during training, which were not used during inference. The standard stochastic gradient descent [53] is used to train the network with a momentum of 0.937, a weight decay of 0.0005 for the Nesterov accelerated gradients utilized, and a batch size of 2. The learning rate is set to 0.01 initially. The entire training process involves 300 epochs.

C. Accuracy Metrics

Accuracy metrics are used to evaluate the difference between the predicted results and the ground truth. The recall (R), precision (P), and mean Average Precision (mAP) are commonly employed as accuracy metrics to evaluate the performance of the method being compared. The definitions of P and R are defined as follows:

$$\mathrm{P} = \frac{TP}{TP + FP} \tag{9}$$

$$\mathrm{R} = \frac{TP}{TP + FN} \tag{10}$$

where the true positive (TP) and true negative (TN) denote correct prediction, and the false positive (FP) and false negative (FN) denote incorrect outcome. The mAP is the average precision across all categories, and the average precision is the area enclosed by the precision and recall curves, so the mAP is calculated by

$$mAP = \frac{AP}{N} = \frac{\int_0^1 p\,(r) dr}{N} \tag{11}$$

where p is precision, r is recall, and N is the number of categories. $mAP_{0.5}$ is the average precision when the IOU threshold is 0.5.

In addition to the evaluation metrics, we employ two further metrics to assess model complexity and computation: giga floating point operations per second and parameter sizes.

D. Ablation Study

In this section, we designed a large number of ablation experiments to validate the effectiveness of our proposed method.

1) Validation of the Baseline Frameworks on LR Datasets: We compared several models from the more maturely developed YOLOv5 series and one model from the current SOTA YOLOv8

TABLE II
TEST RESULTS OF OUR PROPOSED CWM FOR THREE DIFFERENT INPUT IMAGE, RGB, NIR, AND RGB-NIR

method			layers	Params(M)	GFLOPs	mAP0.5(%)
YOLOv5s	RGB	Focus	224	7.100	16.4	62.4
		CWM	289	7.098	68.1	68.2
	IR	Focus	224	7.100	16.4	55.2
		CWM	289	7.098	68.1	62.1
	RGB+IR	MF	291	7.099	67.9	69.9
		CWM	289	7.098	68.6	70.9

series. Table I shows the results of the evaluation of the various baseline frameworks on the VEDAI dataset. The detection performance of the models is evaluated using mAP0.5. Among the models compared, YOLOv5l achieves the best detection performance with a mAP0.5 of 65.6%, however, its layers and parameter sizes (Params) are 1.8 and 6.6 times larger than those of YOLOv5s, respectively, and GFLOPs are 6.9 times larger than those of YOLOv5s. It should be noted that while YOLOv8 is the latest model in the YOLO series, it is not without flaws due to its recent release. YOLOv8s has more layers, parameters, and GFLOPs than YOLOv5s, and its mAP0.5 for small objects is slightly lower than that of YOLOv5s (62.0% versus 62.4%). Concerning YOLOv5s, although its mAP0.5 is slightly lower than YOLOv5l, its smaller size in terms of layers, parameters, and GFLOPs makes it more suitable for deployment in real-time applications, resulting in higher inference capabilities. Therefore, considering the detection performance, model complexity, and inference ability, we choose YOLOv5s as our baseline framework.

2) Validation of CWM: The Focus module, shown in Fig. 1, will resize the input image before entering the backbone. However, this operation can cause resolution degradation and loss of spatial information for small objects. Previous research [43] has demonstrated that removing the Focus module significantly improves the YOLOv5 model's accuracy in detecting small objects. As stated in Section III-B, the NIR images in the VEDAI dataset are considered the same as the RGB images. Therefore, the input image is treated as a four-channel image. We designed a channel-weighting module to replace the Focus module in the original YOLOv5 model. The experimental results are presented in Table II.

The effectiveness of CWM was validated in three cases: RGB image only, NIR image only, and RGB-NIR images. When the input images are only RGB and NIR images, using CWM instead of the Focus module significantly improves the detection performance of small objects. Our proposed CWM resulted in a 5.8% (68.2% versus 62.4%) and 6.9% (62.1% versus 55.2%) improvement in SOD performance, respectively. The size of the layers and parameters in our proposed CWM are comparable to those of Focus, while the GFLOPs are higher. When the input image is RGB-NIR images, we compared the detection performance of the CWM and the MF [43]. The CWM resulted in a 1.0% (70.9% versus 69.9%) improvement, and its Layers, Params, and GFLOPs are comparable to those of the MF. The reason for not differentiating between RGB and NIR in the VEDAI dataset is that transportation objects have similar performance characteristics on both types of images. Therefore, distinguishing between them would not effectively improve the detection performance of transportation objects. Additionally, it would increase the complexity of the network, as demonstrated by the abovementioned experiments.

It was found that the detection performance is significantly higher when the input image is an RGB-NIR images compared with RGB or NIR images (70.9% versus 68.2%), (70.9% versus 62.1%), respectively. Although the performance characteristics of NIR images on objects such as vehicles are not fundamentally different from RGB images, the addition of NIR images increases the information on small objects. This allows the network to acquire more features of small objects and thus improves the detection performance of small objects.

3) Validation of SCM on RGB-NIR Images Dataset: One reason for the low accuracy of SOD is the limited information contained within the small object. Research has shown that increasing contextual information is crucial for improving the accuracy of SOD. The self-attention mechanism can consider both global and local features, making it effective for detecting small objects. To improve the network's attention to global features, we inserted a SCM between the backbone and the neck.

The detection results of adding SCM under two different fusion methods, the channel connection operation (Contact) and CWM, were compared, as shown in Table III. The models with the addition of SCM showed significantly higher mAP0.5 compared to those without SCM. Specifically, mAP0.5 increased by 1.3% (69.9% versus 68.6%) and 7.2% (78.1% versus 70.9%), respectively. When using the Contact, the model with the addition of SCM showed higher detection performance for Camping, Other, Boats, and Vans compared with the model without SCM. The mAP0.5 of the model improved by 10.4%, 4.8%, 0.7%, and 2.0%, respectively. When using the CWM, the model with the addition of SCM shows higher detection performance for all classes compared to the model without SCM. Specifically, Truck and Van show an improvement of more than 10% (14.8% and 10.0%). Pickup, Camping, Other, and Boat more than 5% (9.5%, 7.8%, 5.8%, and 8.2%). The Car and Tractor classes show an improvement of 1.2% and 0.2%, respectively.

The study found that using both CWM and SCM simultaneously resulted in a significant improvement of 7.2% in the model's mAP0.5. This is because both modules enhance the detection performance of small objects, and their combined use allows for co-optimization to improve the detection of small

TABLE III
TEST RESULTS OF OUR PROPOSED SAEM ON THE VEDAI VALIDATION SET

Method		Car	Pickup	Camping	Truck	Other	Tractor	Boat	Van	mAP0.5(%)
YOLOv5s	Contact	87.4	83.2	61.5	75.9	47.3	81.7	42.7	69.1	68.6
	Contact+SCM	86.9	81.8	71.9	75.1	52.1	76.7	43.4	71.1	69.9
	CWM	87.2	76.5	68.0	70.8	58.7	81.2	55.9	69.0	70.9
	CWM+SCM	88.4	86.0	75.8	85.6	64.5	81.4	64.1	79.0	78.1

TABLE IV
ABLATION EXPERIMENT RESULTS ABOUT THE INFLUENCE OF SRPB ON DETECTION PERFORMANCE ON THE VEDAI VALIDATION SET

Method	CWM	SEM	SRPB	mAP0.5(%)
baseline				68.6
Ours	√			70.9
		√		69.9
			√	76.0
	√	√		78.1
	√		√	76.2
		√	√	79.6
	√	√	√	82.6

objects. These experimental results demonstrate that the inclusion of the self-attentive coding module leads to a significant improvement in the accuracy of SOD.

4) Validate the Effects of SRPB: Some ablation experiences about the SRPB are completed in Table IV. The YOLOv5s Focus module was replaced with Contact as the baseline. Compared with the baseline, the model only with SRPB added gets mAP0.57.4% (76.0% versus 68.6%) better than the baseline. Compared with the model with only CWM, the model with both CWM and SRPB gets mAP0.55.3% (76.2% versus 70.9%) than the model with only CWM. Compared with the model with only SCM, Models using both SCM and SRPB get mAP0.59.7% (79.6% versus 69.9%) than models with only SCM. Models that utilized CWM, SEM, and SRPB achieved the highest mAP0.5, which was 4.5% (82.6% versus 78.1%) higher than models that only used CWM and SCM.

Based on the experimental results, it was found that the inclusion of SRPB significantly enhances the detection performance of small objects in RSIs. Because, SRPB can generate HR features, our design of SR perceptual loss links the relationship between the detection task and the SR task. This allows the SR perceptual loss to guide the detection network in learning HR features, resulting in improved detection performance for small objects.

E. Comparisons With Previous Methods

The VEDAI dataset was used to compare several RGB-NIR detection algorithms with modified unimodal detection algorithms. Three of the more advanced RGB-NIR detection algorithms, SuperYOLO [43], YOLOFusion [49], and YOLOrs [55], were compared with three current SOTA algorithms, EESRGAN [9] and ESRTMDet [10], which use SR techniques for object detection. The experimental results are presented in Table V. The proposed method achieves the highest detection performance with a mAP0.5 of 82.63%. It also outperforms other algorithms in detecting Camping, Truck, Other, Tractor, and Van. However, its performance is lower than other algorithms in detecting Cars, Pickup, and Boat. This is because the similarity in appearance between Cars and Pickups makes it challenging to differentiate between the two objects. The ESRTMDet uses SR to recover HR images or features, resulting in improved resolution of Car and Pickup features. This makes it easier for the network to distinguish between the two types of objects.

Fig. 5 shows a comparison of the visual inspection results between our proposed method, SuperYOLO, and YOLOFusion. Our proposed method achieves detection results closest to the ground truth. It effectively addresses missed detection cases of SuperYOLO and YOLOFusion (Truck class in groups a and b) and also performs well in misdetection cases of SuperYOLO and YOLOFusion (Truck class in group b and Other class in groups c and d). We have observed that our proposed method achieves the highest confidence level even for objects that are correctly detected by the other two methods. For instance, in group b, our proposed method achieves an average confidence level of 0.55 for the truck class, which is 0.1 higher than YOLOFusion. Similarly, in group c, our proposed method achieves an average confidence level of 0.9 for the Camping class, which is 0.3 higher than YOLOFusion. For the Camping class in group d, our proposed method achieved an average confidence level of 0.67. This is 0.1 and 0.2 higher than the SuperYOLO and YOLOFusion methods, respectively. In summary, our proposed method achieved the best detection performance.

Grad-CAM can be used to analyze the region of interest of the network model for a certain category, which can be localized to a specific image region. This makes the decision-making process of the neural network more interpretable and visualizable. The Grad-CAM visualization results of our proposed algorithm and the SuperYOLO method are shown in Fig. 6 Upon analyzing the heat map, it is evident that our proposed method outperforms SuperYOLO in terms of focusing on the region of interest. Our algorithm excels in complex backgrounds (such as groups a and c) by accurately identifying the object of interest and avoiding interference from the complex background. Our proposed algorithm can accurately localize the region of interest and detect the object of interest even when the objects are small and blend into the background (e.g., shown in group b).

F. Experimental Results on Other Datasets

We validate our proposed method on the DOTA dataset. The DOTA dataset is a large-scale aerial object detection dataset,

TABLE V
EXPERIMENTAL RESULTS OF THE COMPARISON WITH OTHER ALGORITHMS ON THE VALIDATION SET OF VEDAI DATASET

Method	Car	Pickup	Camping	Truck	Other	Tractor	Boat	Van	mAP0.5	Params	GFLOPs
YOLOrs [55]	84.15	78.27	68.81	52.60	46.75	67.88	21.49	57.91	59.73	-	-
EESRGAN [9]	86.61	82.18	71.88	68.11	50.26	85.28	49.41	85.12	72.36	23.798	259.4
ESRTMDet [10]	**93.07**	**87.58**	72.35	67.63	63.42	86.11	54.17	86.41	76.35	12.737	217.3
YOLOFusion [49]	89.09	81.99	81.45	70.38	71.34	86.38	58.16	82.72	77.69	11.414	113.5
SuperYOLO [43]	91.82	87.26	78.31	86.59	72.88	79.86	**74.40**	70.08	80.15	**7.071**	**67.9**
Ours	89.76	87.07	**81.64**	**87.47**	**73.03**	**87.03**	66.71	**88.34**	**82.63**	7.791	73.7

The significance of the bold values are the best results for each methodology.

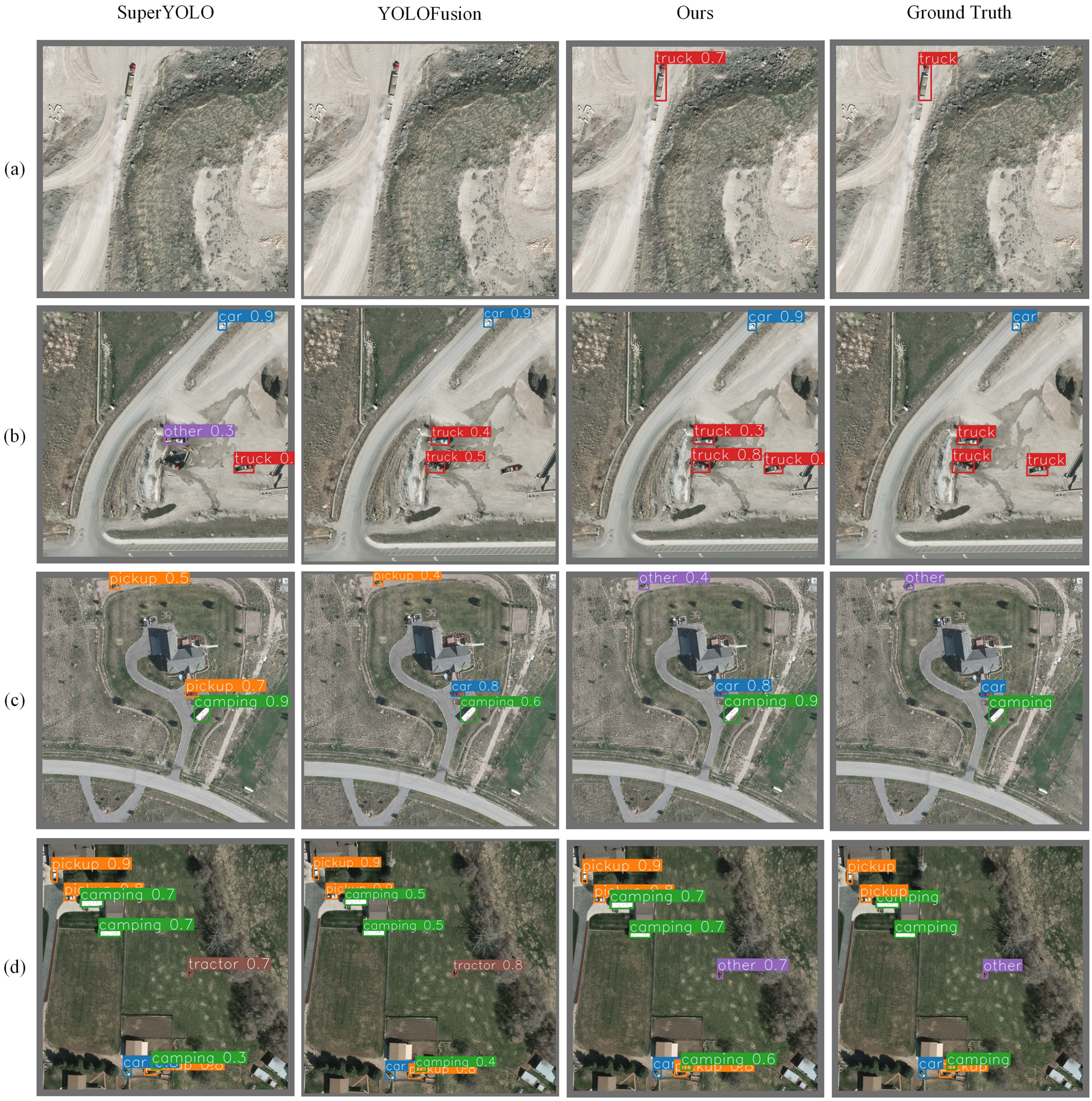

Fig. 5. Visual results of SOD using different methods involving SuperYOLO, YOLOFusion, and our proposed methods.

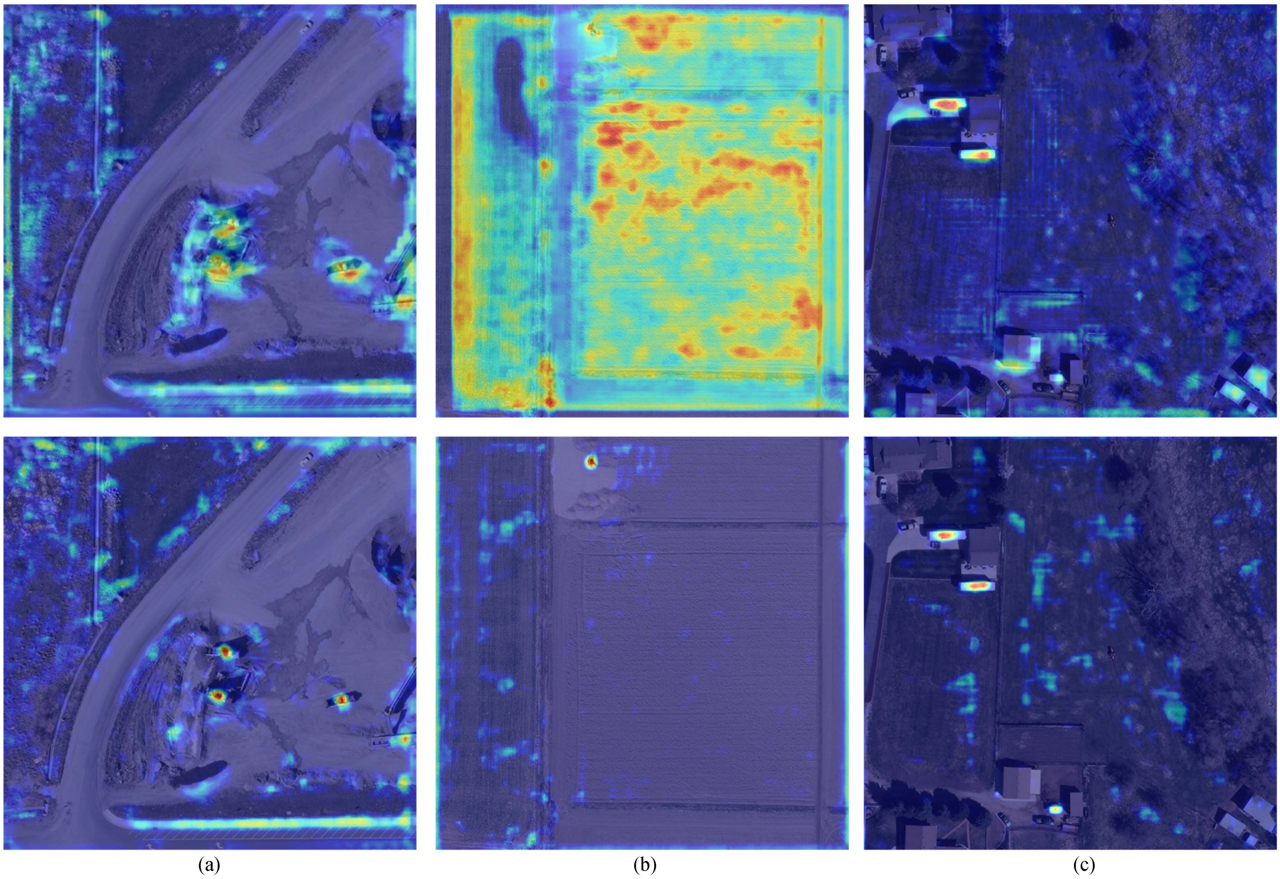

Fig. 6. Grad-CAM visualization results. The first line is the result of the SuperYOLO method and the second line is the result of our proposed method.

TABLE VI
EXPERIMENTAL RESULTS OF THE COMPARISON WITH OTHER ALGORITHMS ON THE DOTA DATASET

Method	SV	PL	SH	HE	P	R	mAP0.5	mAP0.5-0.95
YOLOv5s [11]	68.70	93.50	85.01	55.25	76.94	73.76	75.62	44.72
YOLOv5m [11]	70.01	94.70	86.21	46.21	77.82	73.95	74.28	46.30
YOLOv5l [11]	67.38	94.81	85.91	48.58	79.46	72.28	74.17	46.49
YOLOv5x [11]	67.93	94.70	86.79	51.21	80.33	72.74	75.16	46.71
YOLOv8s [22]	71.71	94.35	89.04	41.92	76.17	72.15	74.26	49.14
RTDETR [54]	74.43	93.24	86.11	25.82	72.85	66.53	69.89	43.79
EESRGAN [9]	75.26	93.55	89.64	33.93	68.61	**78.82**	73.10	49.57
SuperYOLO [43]	67.20	95.59	88.61	42.17	74.26	75.02	73.39	49.06
ESRTMDet [10]	73.63	95.19	**90.20**	36.71	71.76	76.36	73.93	48.16
Ours	**75.86**	**97.01**	89.98	**51.66**	**81.13**	75.93	**78.63**	**50.80**

The significance of the bold values are the best results for each methodology.

which consists of 2806 aerial images ranging from 800 × 800 to 4000 × 4000, and contains a total of 188282 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), helicopters (HC) and other categories. We cut the original image into 1024 × 1024 subimages with an overlap region of 200 pixels, take the 1024 × 1024 subimages as the HR image, and bilinear down-sample the HR image twice to get 512 × 512 images as the low-resolution image. To meet the definition of small objects, we removed the targets with too large size in the original dataset and kept those with smaller size as our targets of interest, i.e., we selected the four categories of planes (PL), small vehicles (SV), ships (SH), and helicopters (HC) as our object of interest for detection. Finally, 7185 images were obtained and 80% of them were used as training sets and 20% as validation and test sets.

To verify the superiority of our proposed method in this article, we selected 9 generic methods for comparison: one-stage detection algorithms, YOLOv5s-x [11], RTDETR [54], and YOLOv8s [22] and three algorithms, EESRGAN, SuperYOLO, and ESRTMDet, which utilize SR techniques to improve object detection. In addition to the previously mentioned evaluation parameters of accuracy (P), recall (R), and mean accuracy at an IOU threshold of 0.5 (mAP0.5), we also utilized the mAP at different IoU thresholds (ranging from 0.5 to 0.95 with a step size of 0.05) (mAP0.5-0.95) as an additional evaluation metric.

As presented in Table VI. The results indicate that our proposed algorithm achieves the optimal detection result. Compared with the two most advanced SR object detection algorithms, SuperYOLO and ESRTMDet, our proposed algorithm achieves the best detection results for three types of objects: small vehicles (SV), planes (PL), and helicopters (HC). Our proposed method improves the precision by 6.89% and 9.37%, mAP50 by 5.24% and 4.7%, and mAP0.5-0.95 by 1.74% and 2.68%, respectively.

V. Conclusion

In this article, we propose a SOD network for RSIs with SR perception, which is a lightweight SOD network built on the widely used YOLOv5 framework that can be used to enhance the detection performance of small objects in RSIs. First, we target RGB images and RGB-NIR images, which are often used in the field of RSI processing. We designed a CWM, that can realize both modal fusion and channel weighting for RGB-NIR RSIs and channel weighting for unimodal (RGB) RSIs so that the network can realize different attention for different channels of RSIs. The module was used to replace the Focus module in YOLOv5, resulting in a significant improvement in the detection accuracy of small objects in RSIs. The following section addresses the issue of poor performance in SOD, which is caused by the limited amount of available information on small objects in RSIs. Then we address the problem of poor performance of SOD caused by little available information on small objects in RSIs. We designed a SCM between the backbone and the neck. This module allows the network to understand the influence relationship between each pixel in the feature map. As a result, the network can focus on the global information of the image while also paying attention to specific pixels. This approach increases the useful information of small objects in RSIs and improves the detection performance of small objects. Finally, we address the issue of the separation between the detection and SR tasks in the current SR-based SOD neighborhood of RSIs. Currently, there is no established interrelationship between the two tasks, which leads to low accuracy in SOD in RSIs. The SRPB and perceptual loss are designed to connect the object detection and SR tasks. HR features from the SR task guide the detection network, allowing the two networks to interact during training for optimal detection results. By removing the SRPB during the inference stage, small objects can be detected without altering the network's original structure. This results in faster inference speed. With the combined contribution of these ideas, our proposed model achieves state-of-the-art detection on the VEDAI dataset, reaching 82.6% mAP0.5, which is slightly higher than the SuperYOLO algorithm in terms of the number of network parameters and GFLOPs. Additionally, our model achieves optimal mAP0.5 in several other categories.

The main limitation of our proposed method is that the model must be trained by HR and LR RSIs. Where LR images are involved in the detection task and HR images are involved in the perception task. Therefore, such training conditions limit the usage scenarios of the proposed method to some extent. However, our proposed method only needs LR data in the inference stage to achieve a higher detection effect than the general model that only uses LR data.

This article highlights the role of SR techniques in SOD in RSIs. It has been demonstrated that SR techniques are important for enhancing the detection of small objects in RSIs. Our goal is to continue improving our model to increase its speed and accuracy in detecting small objects in real-time RSIs.

References

[1] D. Liu, J. Zhang, Y. Qi, Y. Wu, and Y. Zhang, "Tiny object detection in remote sensing images based on object reconstruction and multiple receptive field adaptive feature enhancement," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5616213, doi: 10.1109/TGRS.2024.3381774.

[2] L. P. Osco et al., "A review on deep learning in UAV remote sensing," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 102, Oct. 2021, Art. no. 102456, doi: 10.1016/j.jag.2021.102456.

[3] Y. Yu, T. Gu, H. Guan, D. Li, and S. Jin, "Vehicle detection from high-resolution remote sensing imagery using convolutional capsule networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 12, pp. 1894–1898, Dec. 2019, doi: 10.1109/LGRS.2019.2912582.

[4] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016, doi: 10.1109/TGRS.2016.2601622.

[5] Y. Long, Y. Gong, Z. Xiao, and Q. Liu, "Accurate object localization in remote sensing images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2486–2498, May 2017, doi: 10.1109/TGRS.2016.2645610.

[6] X. Dong, Y. Qin, R. Fu, Y. Gao, S. Liu, and Y. Ye, "Remote sensing object detection based on gated context-aware module," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 6518605, doi: 10.1109/LGRS.2022.3223069.

[7] Y. Ye et al., "An adaptive attention fusion mechanism convolutional network for object detection in remote sensing images," *Remote Sens.*, vol. 14, no. 3, 2022, Art. no. 516, doi: 10.3390/rs14030516.

[8] 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.: Syst.*, vol. 52, no. 2, pp. 936–953, Feb. 2022, doi: 10.1109/TSMC.2020.3005231.

[9] 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, May 2020, Art. no. 1432, doi: 10.3390/rs12091432.

[10] F. Liu et al., "ESRTMDet: An end-to-end super-resolution enhanced real-time rotated object detector for degraded aerial images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 4983–4998, 2023, doi: 10.1109/JSTARS.2023.3278295.

[11] G. Jocher., "YOLOv5 (version 5.0)," Ultralytics, Los Angeles, CA, USA, 2020, doi: 10.5281/zenodo.3908559.

[12] G. Cheng et al., "Towards large-scale small object detection: Survey and benchmarks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13467–13488, Nov. 2023, doi: 10.1109/TPAMI.2023.3290594.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

[14] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[17] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.

[18] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” Apr. 2018, *arXiv:1804.02767*.

[19] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” Apr. 2020, *arXiv:2004.10934*.

[20] 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, doi: 10.1109/CVPR.2019.00075.

[21] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” Dec. 2014, *arXiv:1412.6980*.

[22] G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics YOLO (version 8.0.0),” Ultralytics, Los Angeles, CA, USA, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[23] W. Liu et al., “SSD: Single shot multibox detector,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[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. 2999–3007, doi: 10.1109/ICCV.2017.324.

[25] L. Zhang and X. Wu, “An edge-guided image interpolation algorithm via directional filtering and data fusion,” *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2226–2238, Aug. 2006, doi: 10.1109/TIP.2006.877407.

[26] Y.-W. Tai, S. Liu, M. S. Brown, and S. Lin, “Super resolution using edge prior and single image detail synthesis,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2400–2407, doi: 10.1109/CVPR.2010.5539933.

[27] C. Dong, C. C. Loy, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 184–199.

[28] J. Kim, J. K. Lee, and K. M. Lee, “Accurate image super-resolution using very deep convolutional networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1646–1654, doi: 10.1109/CVPR.2016.182.

[29] 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, doi: 10.1109/CVPR.2017.19.

[30] 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, doi: 10.1109/CVPR.2016.90.

[31] 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. 1132–1140, doi: 10.1109/CVPRW.2017.151.

[32] M. Krichen, “Generative adversarial networks,” in *Proc. 5th Int. Conf. Comput. Commun. Technol.*, 2023, pp. 1–7, doi: 10.1109/ICCCNT56998.2023.10306417.

[33] X. Wang et al., “ESRGAN: Enhanced super-resolution generative adversarial networks,” in *Proc. Eur. Conf. Comput. Vis.*, 2019, pp. 63–79.

[34] K. Jiang, Z. Wang, P. Yi, G. Wang, T. Lu, and J. Jiang, “Edge-enhanced GAN for remote sensing image superresolution,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5799–5812, Aug. 2019, doi: 10.1109/TGRS.2019.2902431.

[35] L. Wang, D. Li, Y. Zhu, L. Tian, and Y. Shan, “Dual super-resolution learning for semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3773–3782, doi: 10.1109/CVPR42600.2020.00383.

[36] 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, doi: 10.1016/j.eswa.2022.116793.

[37] M. Kisantal, Z. Wojna, J. Murawski, J. Naruniec, and K. Cho, “Augmentation for small object detection,” Feb. 2019, *arXiv:1902.07296*.

[38] J. Shermeyer and A. Van Etten, “The effects of super-resolution on object detection performance in satellite imagery,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 1432–1441, doi: 10.1109/CVPRW.2019.00184.

[39] L. Courtrai, M.-T. Pham, and S. Lefèvre, “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, doi: 10.3390/rs12193152.

[40] 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, doi: 10.3390/rs13091854.

[41] S. N. Ferdous, M. Mostofa, and N. M. Nasrabadi, “Super resolution-assisted deep aerial vehicle detection,” *Proc. SPIE*, vol. 11006, 2019, Art. no. 1100617, doi: 10.1117/12.2519045.

[42] L. Zhang and J. Ma, “Salient object detection based on progressively supervised learning for remote sensing images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 11, pp. 9682–9696, Nov. 2021, doi: 10.1109/TGRS.2020.3045708.

[43] 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, 2023, Art. no. 5605415, doi: 10.1109/TGRS.2023.3258666.

[44] S. Elfwing, E. Uchibe, and K. Doya, “Sigmoid-weighted linear units for neural network function approximation in reinforcement learning,” *Neural Netw.*, vol. 107, pp. 3–11, 2018, doi: 10.1016/j.neunet.2017.12.012.

[45] C.-Y. Wang, H.-Y. M. 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 Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 1571–1580, doi: 10.1109/CVPRW50498.2020.00203.

[46] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015, doi: 10.1109/TPAMI.2015.2389824.

[47] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.

[48] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8759–8768, doi: 10.1109/CVPR.2018.00913.

[49] F. Qingyun and W. Zhaokui, “Cross-modality attentive feature fusion for object detection in multispectral remote sensing imagery,” *Pattern Recognit.*, vol. 130, 2022, Art. no. 108786, doi: https://doi.org/10.1016/j.patcog.2022.108786.

[50] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141, doi: 10.1109/CVPR.2018.00745.

[51] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, “Loss functions for image restoration with neural networks,” *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 47–57, Mar. 2017, doi: 10.1109/TCI.2016.2644865.

[52] S. Razakarivony and F. Jurie, “Vehicle detection in aerial imagery: A small target detection benchmark,” *J. Vis. Commun. Image Representation*, vol. 34, pp. 187–203, 2016, doi: 10.1016/j.jvcir.2015.11.002.

[53] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in *Proc. Int. Conf. Comput. Statist.*, 2010, pp. 177–186.

[54] Y. Zhao et al., “DETRs beat YOLOs on real-time object detection,” Apr. 2023, *arXiv:2304.08069*.

[55] M. Sharma et al., “YOLOrs: Object detection in multimodal remote sensing imagery,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 1497–1508, 2021, doi: 10.1109/JSTARS.2020.3041316.

Jiahang Liu (Member, IEEE) received the B.S. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2000, the M.S. degree in tectonics from the Institute of Geology, China Earthquake Administration, Beijing, China, in 2003, and the Ph.D. degree in pattern recognition and intelligent system from Shanghai Jiao Tong University, Shanghai, China, in 2011.

He has been a Full Professor with Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include remote sensing, image processing, computer vision, and digital twin.

Jinlong Zhang received the B.S. degree in optoelectronic information science and engineering in 2022 from Nanjing University of Aeronautics and Astronautics, Nanjing, China, where he is currently working toward the M.S degree in electronic information.

His research interests include remote image processing and computer vision.

Yue Ni received the B.E. degree in information engineering from Nanjing University of Information Science and Technology, Nanjing, China, in 2019, and the M.S. degree in communication and information engineering in 2022 from Nanjing University of Aeronautics and Astronautics, Nanjing, China, where he is currently working toward the Ph.D. degree in optical engineering with the College of Astronautics.

His current research interests include Semantic segmentation of remote-sensing images, remote-sensing image processing, and deep learning.

Weijian Chi received the B.E. degree in information engineering in 2020 from the College of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing, China, where he is currently working toward the Ph.D. degree in optical engineering.

His research interests include infrared small target detection, unsupervised semantic segmentation, deep learning, and pattern recognition.

Zitong Qi received the B.S. degree in optoelectronic information science and engineering in 2022 from Nanjing University of Aeronautics and Astronautics, Nanjing, China, where he is currently working toward the M.S degree in electronic information.

His research interests include remote image processing and computer vision.