CRKD-YOLO: Cross-Resolution Knowledge Distillation for Low-Resolution Remote Sensing Image Object Detection

Xiaochen Huang, Qizhi Teng, *Member, IEEE*, Hong Yang, Xiaohai He, *Member, IEEE*, Linbo Qing, *Member, IEEE*, Pingyu Wang, and Honggang Chen, *Member, IEEE*

***Abstract*— The majority of advanced remote sensing object detection technologies excel in accurately detecting objects from high-resolution images. However, in practical scenarios, it is often necessary to detect objects in images of varying resolutions due to differences in imaging equipment. When dealing with lower-resolution images, the limited detailed information and blurry boundaries lead to a noticeable decrease in detection accuracy. To address this problem, we propose an efficient object detection method for low-resolution remote sensing images based on the YOLO detector, named CRKD-YOLO. The method constructs a cross-resolution knowledge distillation (CRKD) framework to resolve the issue of feature mismatch, enabling the model with low-resolution inputs to learn more refined feature representations from high-resolution images. Furthermore, to effectively leverage the limited detailed information in low-resolution images, we propose the backbone augment feature pyramid network (BAFPN). It enhances detection accuracy for low-resolution remote sensing images while making the model more lightweight. Massive experiments on DOTA, DIOR, NWPU VHR-10, DroneVehicle, and VEDAI demonstrate that our CRKD-YOLO achieves significant improvements, even achieving higher accuracy compare to training and testing high-resolution images with baseline. Our code is published at https://github.com/Jianfantasy/CRKD-YOLO**

***Index Terms*— Cross-resolution knowledge distillation (CRKD), feature enhancement, object detection, remote sensing images.**

Received 15 July 2024; revised 3 December 2024; accepted 28 March 2025. Date of publication 10 April 2025; date of current version 21 April 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62001316 and Grant 62271336; in part by Sichuan Science and Technology Program under Grant 2024YFHZ0212; in part by the Open Foundation of Yunnan Key Laboratory of Software Engineering under Grant 2023SE206; in part by the Opening Foundation of Key Laboratory of Computer Vision and System, Ministry of Education, Tianjin University of Technology, China, under Grant TJUT-CVS20220001; and in part by the Fundamental Research Funds for the Central Universities under Grant SCU2023D062 and Grant 2022CDSN-15-SCU. The Associate Editor coordinating the review process was Dr. Dan Zhang. *(Corresponding authors: Pingyu Wang; Honggang Chen.)*

Xiaochen Huang, Qizhi Teng, Hong Yang, Xiaohai He, Linbo Qing, and Pingyu Wang are with the College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China (e-mail: 2022222050160@stu.scu.edu.cn; qzteng@scu.edu.cn; yhscu@scu.edu.cn; hxh@scu.edu.cn; qing_lb@scu.edu.cn; wangpingyu@scu.edu.cn).

Honggang Chen is with the College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China, also with Yunnan Key Laboratory of Software Engineering, Yunnan University, Kunming 650600, China, and also with the Key Laboratory of Computer Vision and System, Ministry of Education, Tianjin University of Technology, Tianjin 300384, China (e-mail: honggang_chen@scu.edu.cn).

Digital Object Identifier 10.1109/TIM.2025.3559616

I. Introduction

OBJECT detection is a fundamental task in computer vision. In recent years, the rapid advancement of deep learning, coupled with the proposal of numerous large-scale datasets [1], [2], [3], has enabled CNN-based detectors [4], [5], [6], [7] to achieve remarkable performance, significantly surpassing traditional detection algorithms. This progress has also greatly advanced object detection in remote sensing images, pushing the boundaries of accuracy and efficiency [8].

In remote sensing scenarios, the typically extensive image coverage area requires high-resolution imaging, which demands advanced imaging equipment and poses significant data transmission challenges. Consequently, images with varying resolutions, including low-resolution images, are often collected and need to be processed for detection. However, it is worth noting that most existing remote sensing object detection techniques are trained and optimized on high-quality datasets, such as DOTA [1] and DIOR [3]. As a result, when the input consists of low-resolution images, detection accuracy often drops significantly due to the lack of detailed spatial information and the presence of ambiguous object boundaries [9], [10]. In addition, detecting small targets remains a critical challenge, especially in remote sensing images, where there are numerous and dense small targets with complex backgrounds [11]. The region of interest (ROI) occupies very few pixels among them, so the target information that the network can extract is extremely limited. The inadequate image resolution further diminishes the pixel count within target regions, exacerbating the compromised accuracy resulting from a lack of intricate details.

To improve the detection accuracy of low-resolution images, an intuitive method is to use interpolation to upscale images to a higher resolution that matches the model's training resolution before inputting them into the detection model. This approach does little to recover details, limiting its ability to achieve detection accuracy comparable to true high-resolution inputs. As depicted in Fig. 1(b) and (c), the model trained on high-resolution images exhibits limited accuracy in detecting targets of low-resolution images, particularly small targets that are nearly unrecognizable. Conversely, training the model on low-resolution images, as shown in Fig. 1(d), enhances adaptability to such images to some extent. However, the

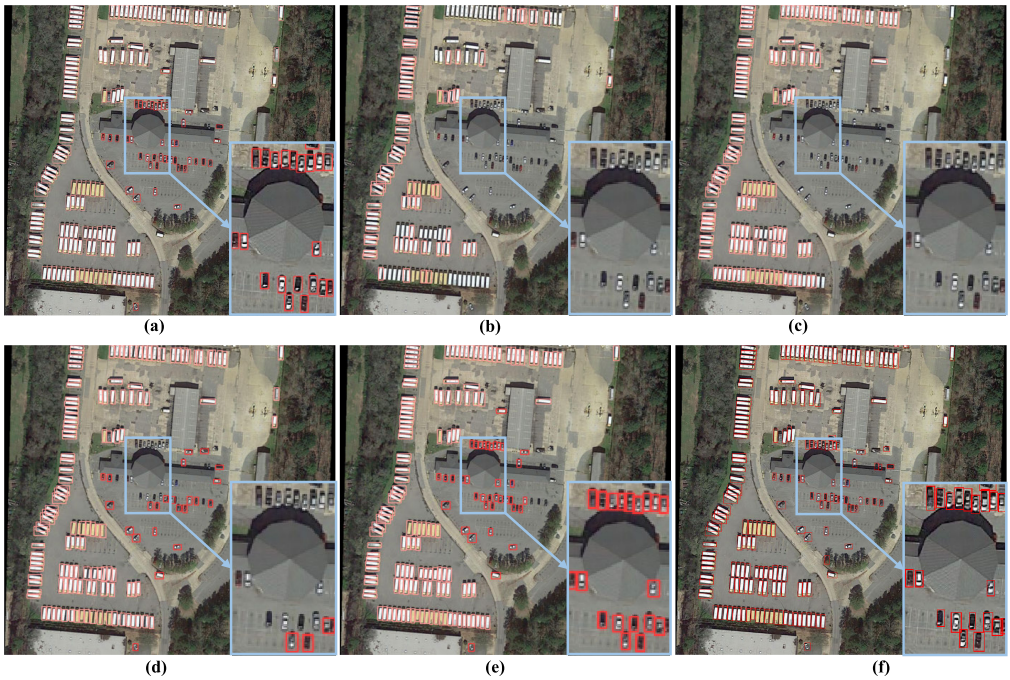

Fig. 1. Visualization of the detection results in high- and low-resolution images by different ways. Specifically, the following steps are applied. (a) Detect high-resolution images with a model trained on high-resolution images. (b) Detect low-resolution images with a model trained on high-resolution images. (c) Detect low-resolution images by resizing them to high-resolution size before inputting them into the model trained on high-resolution images. (d) Detect low-resolution images with a model trained on low-resolution images. (e) Detect low-resolution images with our CRKD-YOLO. (f) Ground truth. Notedly, we fix the size of six images to the same for comparison, with (a) and ground truth being twice the size of the others actually.

detection accuracy remains far inferior compared to Fig. 1(a), where both training and validation are conducted using high-resolution images, indicating there is still ample room for improvement. Numerous works have focused on enhancing details and texture information of input images through super-resolution [12], [13], [14], [15], [16], [17]. These efforts have demonstrated significant improvements in detector performance, especially in the detection of small targets. But the additional model parameters introduced by super-resolution networks and the increased size of super-resolution images significantly multiply computational costs, hindering their applicability in real-time object detection. Some studies adopt a joint learning paradigm [9], [18] by using a shared backbone to encode features for super-resolution and detection tasks during the training stage and subsequently removing the super-resolution module during the inference stage. Although this method avoids the additional burden associated with super-resolution, the shared feature extraction network designed for the two tasks is not fully optimized for object detection, resulting in extracted features that are not entirely adapted to the detection task.

To further address the issue of inadequate feature adaptability, we propose CRKD-YOLO, as illustrated in Fig. 2. This method leverages cross-resolution knowledge distillation (CRKD) to learn feature representations from high-resolution images, while avoiding the additional burden associated with a super-resolution module. Additionally, the novel backbone augment feature pyramid network (BAFPN) module is integrated to enhance model performance by emphasizing detailed information while reducing model parameters. These innovations enable CRKD-YOLO to deliver exceptional detection performance on low-resolution images. Despite its lightweight design, the method surpasses baseline networks applied to high-resolution images, demonstrating superior detection capabilities and an excellent trade-off between performance and complexity for low-resolution remote sensing images. As depicted in Fig. 1(e), our improved algorithm significantly elevates detection accuracy compared to methods (b)–(d), especially in detecting small vehicles. The experimental results indicate that, compared with the current method, our improvements can effectively enhance the detection performance of low-resolution images while reducing the model complexity.

On the whole, our study makes the following contributions.

1) We analyze the impact of input and feature dimension variations on detection results, proposing CRKD-YOLO to enhance low-resolution image features through information guidance from high-resolution images and reinforcement of detailed texture.
2) Our proposed CRKD leverages knowledge distillation to help the student network capture high-resolution information from low-resolution images. It enlarges low-resolution image features to enhance details and addresses feature mismatch issues caused by varying input resolutions, enabling the student network to learn refined high-resolution feature representations.
3) To capture detailed texture information comprehensively, we put forward BAFPN. It allows the model to focus more on small targets, making the model more suitable for detecting low-resolution remote sensing images while maintaining a lightweight structure.

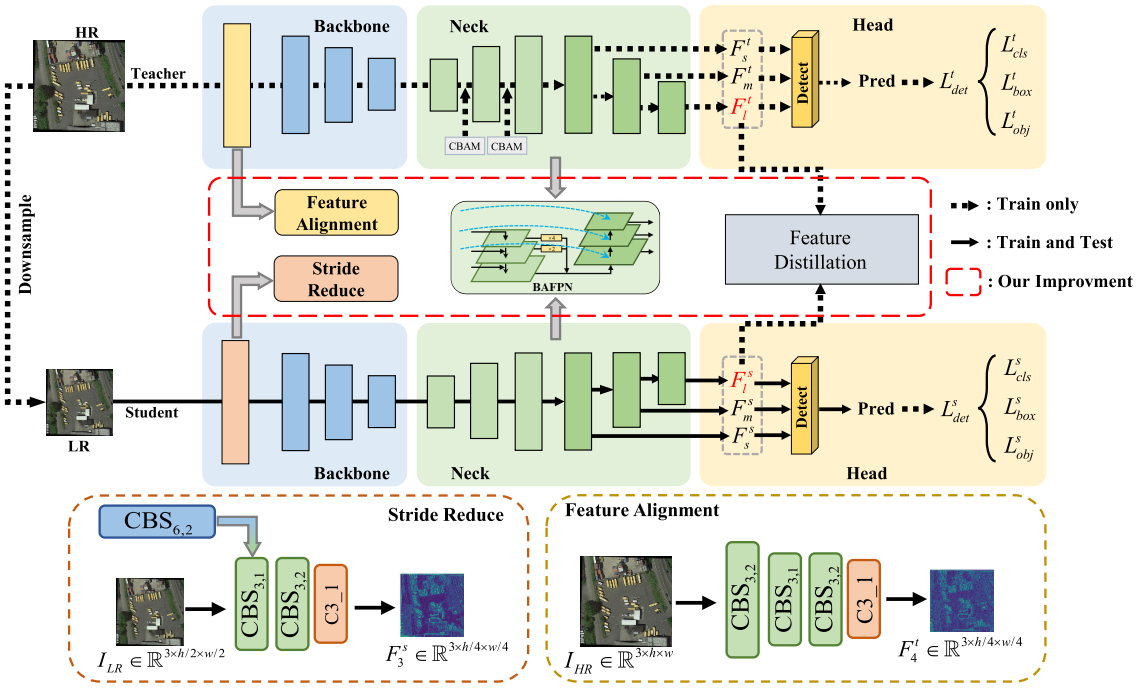

Fig. 2. Framework of CRKD-YOLO. The $\mathrm{CBS}_{k,s}$ refers to the convolution–batch normalization–Silu with a kernel size k and stride s. $C3\_n$ is the C3 module. The $\mathrm{CBS}_{6,2}$ is the CBS of the original backbone that is replaced to reduce stride. The model is trained with existing standard datasets, and only the student branch is used for detecting low-resolution images.

4) We construct CRKD-YOLO by integrating CRKD with BAFPN and conduct extensive experiments to demonstrate its significant improvement in detecting low-resolution remote sensing images, even surpassing the accuracy of baseline models trained and tested on high-resolution images.

II. Related Work

A. Object Detection With Super Resolution

In recent years, with the rapid advancement of deep learning [19], [20], [21], [22], [23], [24], [25], CNN-based algorithms have been widely applied in various domains, including ship detection [26], autonomous driving [27], defect inspection [28], medical decision-making [29], and military surveillance [30]. Currently, learning-based object detection methods can be broadly categorized into two types: one-stage detectors, represented by the YOLO series [5], [31], [32], [33], [34], [35], [36], [36], SSD [4], and RetinaNet [37], and two-stage detectors, exemplified by R-CNN [38], Fast R-CNN [6], and Faster R-CNN [7]. While two-stage detectors generally exhibit certain advantages in accuracy, they come with higher computational costs. In contrast, one-stage detectors typically offer faster inference speeds. Notably, thanks to improvements such as multiscale feature fusion [39], [40] and the application of focal loss [37], the accuracy of one-stage detectors has been significantly improved, making them more suitable for real-time object detection. These methods typically use consistent input sizes for training and testing but often exhibit diminished accuracy when processing lower-resolution images.

To improve detection performance under resolution deficiency, some studies have enhanced the network's information representation capacity across different scales using multiscale feature learning [39], [41] or texture-based detection methods [42]. Moreover, many researchers have investigated object detection supported by image super-resolution to mitigate the issue of insufficient pixel information [9], [12], [16], [18], [30], [43], [44], [45].

Among these studies, Shermeyer and Van Etten [43] discussed the application of super-resolution techniques in satellite images and their impact on object detection algorithms. Li and Shen [30] improved the YOLOv5 network by integrating super-resolution processing and multidata augmentation for input images, effectively reducing missed detections and false alarms. Haris et al. [12] proposed a joint optimization approach that balances detection loss and super-resolution loss, enabling the training of super-resolution preprocessing for any differentiable detector. Yang et al. [16] proposed a mutual-feed learning approach by establishing a feedback connection alongside the forward connection between detection and super-resolution tasks. Cui et al. [45] employed downsampling degradation as a self-supervised signal transformation to explore invariant representations across various resolutions and degradation conditions, optimizing the model jointly in an end-to-end manner. While incorporating super-resolution to support the detection task can effectively mitigate the accuracy degradation caused by low-resolution images, this approach introduces additional parameters and computational burden due to the inclusion of the super-resolution module.

To avoid the additional complexity associated with image super-resolution, Zhang et al. [18], Liu et al. [9], and Wang et al. [46] integrated image super-resolution with image

perception technology by training a single backbone network to simultaneously perform image restoration and perception using features extracted from low-resolution inputs. This approach enables the extraction of high-resolution information from low-resolution inputs and eliminates additional computational costs by removing the super-resolution module during inference. However, introducing a super-resolution network through this method would lead to a deviation between the features extracted by the backbone network and those required for detection. In other words, the backbone network is not "specialized" for the detection task and needs to make certain compromises to accommodate the effectiveness of the restoration network.

In contrast, our proposed CRKD-YOLO adopts a knowledge distillation strategy, effectively learning high-resolution feature representations tailored for detection tasks without introducing additional complexity.

B. Knowledge Distillation

Knowledge distillation generally refers to the process of transferring knowledge from a teacher network to a student network and is widely utilized for model lightweighting. Hinton et al. [47] conducted pioneering work in this area, introducing knowledge distillation as an effective technique in the context of classification tasks. Subsequently, numerous studies [48], [49], [50] have enhanced the performance of student classifiers through the application of knowledge distillation.

In recent years, a large number of works [51], [52], [53], [54], [55], [56] have migrated knowledge distillation to the field of object detection. Chen et al. [51] first proposed using knowledge distillation in object detection to balance detection accuracy and speed. Li et al. [52] chose to calculate the distillation loss between features extracted from the region proposal network (RPN) and optimized the similarity between the features. Wang et al. [53] suggested that distillation loss should pay more attention to neighboring regions of the targets. Yang et al. [54] adaptively extracted the multiscale core features of targets for distillation and focused more on the features of small targets using an area-weighted strategy. Guo et al. [55] found that both foreground and background are important in distillation, and distilling them separately provides greater benefits to the student model. By separating foreground and background, Yang et al. [56] encouraged the student model to concentrate on the critical pixels and channels in the teacher model. All of these methods are designed to improve performance on a more lightweight model.

In contrast, our proposed CRKD leverages knowledge distillation to enhance recognition performance on low-resolution images. It addresses the issue of feature spatial resolution mismatch caused by varying input sizes. By simultaneously training the teacher and student networks, the student network is guided to learn from high-resolution image features, enabling it to achieve comparable accuracy with models trained and tested on high-resolution images.

III. CRKD-YOLO Architecture

Fig. 2 provides an overview of the proposed framework, and it will be elaborated in this section. We first introduce the selected baseline, YOLOv5s, which can balance between inference speed and accuracy well. Moreover, we describe the improvements of CRKD-YOLO for low-resolution remote sensing image object detection, specifically encompassing two aspects: 1) the CRKD to get guidance from high-resolution image information and 2) the BAFPN to intensify feature extraction and fusion.

A. Baseline Architecture

As shown in Fig. 2, the YOLOv5s baseline network consists of three primary components: the backbone, the neck, and the detection head. The backbone is employed to extract both low-level texture features and high-level semantic features from images. Its structure, detailed in Fig. 3, primarily consists of modules such as CBS, C3, and SPPF. CBS, composed of convolution, batch normalization, and the SiLu activation function, is utilized to extract multiscale features while normalizing and nonlinearly transforming the extracted information. The C3 module consists of three CBS modules and a series of bottlenecks, forming deeper nonlinear representations that enable the network to better comprehend semantic information in images. The SPPF module, comprising two CBS modules and three cascaded max-pooling operations, concatenates the features from these max-pooling operations along the channel dimension. This design expands the receptive field and facilitates multiscale feature fusion.

Subsequently, as shown in Fig. 4(a), the neck of PAFPN structure (FPN [38] + PANet [40]) in YOLOv5s integrates low-level texture features at the process passing high-level semantic features from top to bottom. In PANet, the output features from FPN are propagated from bottom to top, while re-incorporating intermediate features from FPN. This fusion strategy allows the features in PANet to simultaneously retain detailed texture information and global semantic features. Eventually, the detection head decodes the three features of different sizes produced by PANet, effectively balancing detection performance across targets of varying scales and enhancing overall robustness. However, when detecting objects in low-resolution remote sensing images, the aforementioned structures encounter two significant challenges.

1) *Issue of small target:* Targets in remote sensing images are often extremely small, while the three features used for decoding in the PAFPN are reduced to scales that are 8×, 16×, and 32× smaller than the input image, respectively. These reduced feature sizes are insufficient to effectively capture and process detailed information, significantly limiting the model's ability to detect small-sized objects.
2) *Loss of target information:* Due to the scarcity of target details in low-resolution images, the bottom-layer texture features may be lost or become blurred. This results in the inadequacy of critical information when features are propagated to higher levels.

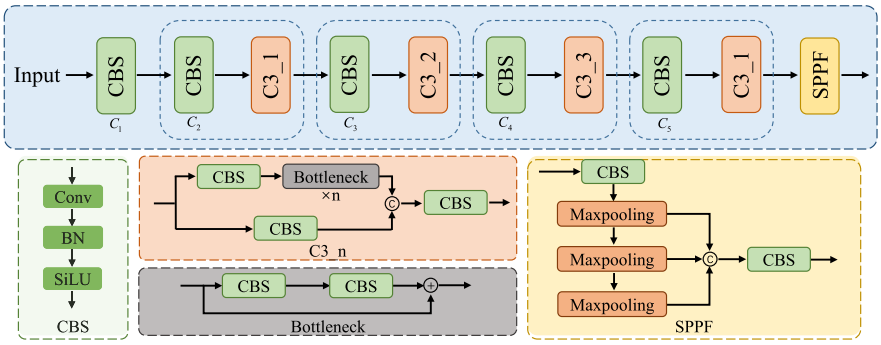

Fig. 3. Backbone network of YOLOv5s. The features from the module in a dashed line with blue frame have the same size.

B. CRKD

As shown in Fig. 2, to enhance the detection performance of low-resolution remote sensing images, we introduce three main improvements in our CRKD to effectively leverage refined information from high-resolution images.

1) *Reducing convolution stride:* This adjustment enlarges the feature representation in the student network when processing low-resolution inputs.
2) *Feature alignment:* A convolution module is inserted to align the high-resolution image features from the teacher branch with the low-resolution image features of the student branch.
3) *Regression loss constraint:* The enlarged low-resolution features and high-resolution features are constrained using a regression loss, enabling the teacher to distill more refined information to the student during training.

In many object detection networks [5], [6], [7], [33], the feature extraction stage often employs convolution with a stride of 2 to extract multiscale features, reducing feature size by half. While this approach effectively reduces the computational burden during deep feature extraction, it may result in the loss of target information due to the lower feature resolution [18], severely impacting the detection accuracy for small targets. Therefore, we adjust the stride of the student model with low-resolution inputs to be half that of the teacher model with high-resolution inputs. This ensures spatial dimension alignment between their features and helps mitigate information loss during feature extraction, particularly for low-resolution images. The equal spatial dimension of the corresponding features from the teacher and student models facilitates effective mutual learning between them during training. As presented at the bottom of Fig. 2, we reduce the stride of the first convolution module to half. This adjustment can preserve twice the spatial size in features compared to before, contributing to mitigating the problem of information shortage for small targets caused by low-resolution inputs and overly small feature sizes. More importantly, enlarging the features of low-resolution images is essential for achieving feature matching with high-resolution counterparts. Building upon this adjustment, we incorporate a teacher branch to extract features from high-resolution images, allowing the low-resolution student network to learn more refined features from their high-resolution counterparts during training. To ensure that the features from the high-resolution input of the teacher and the low-resolution input of the student have the same spatial dimensions for effective distillation, the teacher network is designed with a larger stride.

Specifically, as shown in Fig. 2, during the training process, we downsample the high-quality input $I_t \in \mathbb{R}^{3\times h\times w}$ to create pairs of high- and low-resolution images by bilinear interpolation, expressed as

$$I_s = D(I_t) \tag{1}$$

where the $D(\cdot)$ represents the downsampling by bilinear interpolation. The size of sampled I_s could be expressed as $I_s \in \mathbb{R}^{3\times h/2\times w/2}$. The student model is trained using the I_s to effectively extract features specific to such images, while the teacher model is trained with I_t to extract fine-grained features used to guide the student branch.

Then, the student branch extracts feature with larger sizes than baseline by reducing the stride of the convolution module. The teacher network uses the same model as the student but includes a CBS module at the beginning of the model to ensure consistency in feature size with the student branch. The CBS module has a stride of 2 and a kernel size of 3×3, which halves the spatial size of its output feature compared to the input. Thus, the output features after "Stride Reduction" in the student branch and "Feature Alignment" in the teacher branch can be expressed as follows:

$$F_3^s = C3\big(\mathrm{CBS}_{3,2}\big(\mathrm{CBS}_{3,1}(I_s)\big)\big) \tag{2}$$

$$F_4^t = C3\big(\mathrm{CBS}_{3,2}\big(\mathrm{CBS}_{3,1}\big(\mathrm{CBS}_{3,2}(I_t)\big)\big)\big) \tag{3}$$

where F_3^s and $F_4^t \in \mathbb{R}^{C\times h/4\times w/4}$ retain same spatial dimension. Furthermore, since the remaining modules in teacher and student network make the same dimensional transformations to their input features, F_{i+1}^t (the output of layer $i+1$ in the teacher's network) and F_i^s (the output of layer i in the student's network) share the same feature size, where $i \geq 1$. This alignment facilitates effective knowledge distillation at a feature level. Simultaneously, we observe that utilizing the last layer's feature F_l from both the teacher and student networks to calculate the distillation loss is optimal. This approach is not only training-friendly but also ensures effective parameter updates across the entire network during the backpropagation process. By leveraging the aforementioned improvements, the student network can effectively extract rich high-resolution knowledge from the teacher network, thus obtaining a high-performance model for detecting low-resolution image.

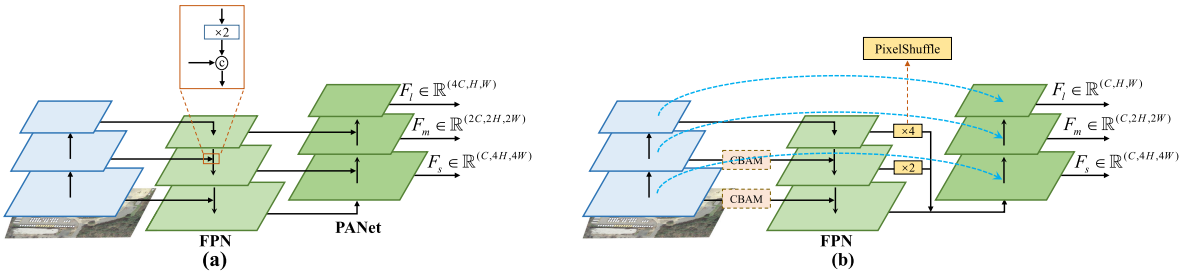

Fig. 4. Structure of (a) PAFPN of YOLOv5 and our proposed BAFPN. The CBAM is only applied when the (b) BAFPN is introduced in a teacher network.

C. BAFPN

In the low-resolution remote sensing images, due to the large area represented by a single pixel, the target size in the images is usually small, and detailed information is usually insufficient. Therefore, it is necessary to rebalance the detection performance across objects of different scales, with a greater emphasis on small-scale targets, to enhance detailed texture information. For this reason, we substitute our BAFPN for the original PAFPN in our CRKD-YOLO, as drawn in Fig. 2. It incorporates additional backbone features to augment the representation of intricate texture information and adjusts the number of channels in the features across different scales to prioritize small targets.

As depicted in Fig. 4(b), it first constructs an FPN in a top-down manner to propagate high-level semantic features while progressively incorporating low-level texture features from the backbone during this process. Then, in FPN, the three features that have, respectively, integrated the backbone features, denoted as F_1^{FPN}, F_2^{FPN}, and F_3^{FPN}, are unified in spatial size by upsampling the two smaller features using pixelshuffle. Additionally, the number of feature channels is adjusted by 1×1 convolution to ensure that they are at the same channel number. The process can be formulated as follows:

$$X_i^{\text{FPN}} = \text{Conv}_{1,1}\left(Ps^n\left(F_i^{\text{FPN}}\right)\right) \tag{4}$$

where the $\text{Ps}^n(\cdot)$ is pixelshuffle, and $\text{Conv}_{k,s}(\cdot)$ represents the convolution with the kernel size $k \times k$ and stride s. n equals 4 or 2 according to the size of upsampled features. i equals 1 or 2, representing which feature is upsampled. Subsequently, a fused feature that has rich low-level detailed texture and high-level semantic information is generated by

$$X = \text{Concat}\left(X_1^{\text{FPN}}, X_2^{\text{FPN}}, F_3^{\text{FPN}}\right)_c \tag{5}$$

where $\text{Concat}(\cdot)_c$ expresses concatenation along the channel dimension. The produced feature is gradually transferred through a bottom-up pyramid structure like PANet, resulting in three features named F_s, F_m, and F_l, which are tailored for detecting small, medium, and large targets, respectively. Among them,

$$F_s = C3\left(\text{Concat}\left(\text{Conv}_{3,1}(X), F_3^b\right)_c\right) \tag{6}$$

$$F_m = C3\left(\text{Concat}\left(\text{Conv}_{3,2}(F_s), F_2^b\right)_c\right) \tag{7}$$

$$F_l = C3\left(\text{Concat}\left(\text{Conv}_{3,2}(F_m), F_1^b\right)_c\right) \tag{8}$$

where $C3(\cdot)$ is the C3 module stated in Section III-A. The F_i^b corresponds to the backbone feature. From (6)–(8), it can be observed that features from the backbone, rather than the FPN, are integrated into the decoded features. This integration leverages more low-level features, allowing greater focus on detailed information compared to the PAFPN. Eventually, we enhance the detector's focus on small targets by rebalancing the proportion of channel numbers among F_s, F_m, and F_l This adjustment not only improves detection performance on low-resolution remote sensing images but also significantly reduces model parameters.

Eventually, CBAM [57] has demonstrated that focusing on critical channels and pixels contributes to extracting more useful information. Therefore, in our framework, as shown in Fig. 4, CBAM is only introduced when the teacher network merges backbone features into FPN, which not only contributes to enhancing the feature in FPN but also retains the essential low-level features. Through distillation and the CBAM module, the student network can focus on important channels and pixels while avoiding computational overhead introduced by the attention module.

D. Loss Function

To learn high-resolution image information from the teacher network, we introduce a distillation loss L_{dis}. It uses L1 loss between two F_l of teacher and student network, expressed as

$$L_{\text{dis}} = \left\| F_l^t - F_l^s \right\|_1 \tag{9}$$

where F_l^t is the last layer feature of teacher and F_l^s is the last layer feature of student.

The detection loss is composed of three parts: L_{box} for the position discrepancy between predicted and actual bbox, L_{obj} for bbox confidence, and L_{cls} for category prediction, which can be expressed as

$$L_{\text{det}} = \lambda_{\text{box}} L_{\text{box}} + \lambda_{\text{obj}} L_{\text{obj}} + \lambda_{\text{cls}} L_{\text{cls}} \tag{10}$$

where L_{obj} and L_{cls} are calculated by cross-entropy function, and L_{box} is calculated by CIOU loss. λ_{box}, λ_{obj}, and λ_{cls} are balance coefficients, with $\lambda_{\text{box}} = 0.05$, $\lambda_{\text{obj}} = 1$, and $\lambda_{\text{cls}} = 0.5$ in both of student and teacher branches.

Given the method of online distillation to optimize both the teacher and student network concurrently, it is imperative to compute the detection loss for both branches simultaneously. Above all, the overall loss is expressed as

$$L = \alpha^t L_{\text{det}}^t + \alpha^s L_{\text{det}}^s + \beta L_{\text{dis}} \tag{11}$$

where α^t and α^s are the coefficients for detection losses of teacher and student branches. β is the coefficients for distillation loss. We set $\alpha^s = \alpha^t = 1$ and $\beta = 0.1$.

TABLE I
TRAINING STRATEGY

Dataset	Train Size	Test Size	Batch Size	Epoch
DOTA	1024 × 1024	512 × 512	8	100
DIOR	800 × 800	400 × 400	8	150
NWPU	1024 × 1024	512 × 512	8	300
DroneVehicle	640 × 640	320 × 320	8	100
VEDAI	1024 × 1024	512 × 512	8	300

IV. EXPERIMENTAL SETTINGS AND RESULTS

This section presents the experimental details of our CRKD-YOLO. We begin by introducing the datasets and implementation details, followed by the results and analysis of the experiment.

A. Datasets

1) DOTA-v1.0: DOTA [1] is a large-scale remote sensing dataset proposed in the year 2018. It was collected from Google Earth, JL-1 satellite, and GF-2 satellite, containing 2806 large images and 188 282 instances. The original image size is 4000 × 4000, with a spatial resolution from 0.1 to 1 m. The experiment utilizes the default partition ratio, whereby 1/2 of the original images are allocated to the training set, 1/6 to the validation set, and 1/3 to the test set. We crop the image to a size of 1024 × 1024 with an overlap of 200 pixels. The images remain fixed at 512 × 512 during inference unless otherwise stated.

2) DIOR: DIOR [3] is a large-scale remote sensing dataset proposed in the year 2020 with a spatial resolution from 0.5 m to 1 m. It contains 23 463 images and 192 472 instances. The image size is 800 × 800. We choose 11 725 images as the training set and 11 738 images as the test set. During inference, the image size is fixed is 400 × 400.

3) NWPU VHR-10: NWPU VHR-10 [58] was proposed in the year 2016. It contains 800 images, of which 650 pictures contain objects, so we select 520 images as the training set and 130 images as the testing set. The dataset contains 10 categories. The size of the image is fixed to 1024 × 1024 for training and 512 × 512 for inference.

4) DroneVehicle: DroneVehicle [59] is a dataset consisting of aerial RGB-IR images captured by drones, covering various scenes from an aerial perspective. It includes five categories of target objects and contains a total of 28 439 pairs of infrared-visible image pairs. The images are categorized into three distinct scenes: day, night, and dark night. During inference, the image size is fixed at 320 × 320.

5) VEDAI: The VEDAI dataset [60] is a widely used multimodal dataset consisting of 1246 pairs of RGB-IR images, featuring diverse backgrounds such as grasslands, highways, mountains, and urban areas. Following the setup in [18], we selected 1089 images for training and 121 images for testing. The images are available in sizes of 1024 × 1024 or 512 × 512.

B. Experimental Setup

There are the training strategies for the five datasets presented in Table I unless otherwise specified. In addition, hue saturation values (HSVs), translation, flip, and mosaic are used to augment the data during the training phase. Using stochastic gradient descent (SGD) as the optimizer to train the teacher and student models synchronously, the initial learning rate is set to 0.01, the momentum to 0.937, and the weight decay to 0.0005. We convert the label to YOLO format, including the class ID, bounding box length and width, bounding box center coordinates, and normalize the length, width, and the coordinate to (0–1). Our experiments are implemented in PyTorch and run on NVIDIA 3090 GPUs.

C. Accuracy Metrics

We evaluate accuracy by mAP metric to calculate the difference between the detection results and the ground truths. It is mainly composed of three aspects: precision, recall, and intersection over union (IOU). Among them, IOU represents the intersection and union ratio between the predicted box and the real box. The prediction result is true only if the IOU is greater than a certain threshold, which is set to 50% in the subsequent experiments. Precision and recall are severally defined as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{12}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{13}$$

where the true positive (TP) and true negative (TN) are the number of correct predictions for positive and negative samples, respectively, while the false positive (FP) and false negative (FN) denote the number of false predictions for positive and negative samples, respectively. From the precision and recall of each category, we can obtain its precision–recall curve. The average precision (AP) for the corresponding category is calculated as the area under the curve, bounded by the coordinate axes. The mAP, obtained by averaging the AP values of each category, comprehensively reflects the performance of the model. In brief, AP and mAP can be calculated by

$$\text{AP} = \int_0^1 P(R) dR \tag{14}$$

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i \tag{15}$$

where P is precision, R represents recall, and N is the number of categories.

Furthermore, we evaluate the model complexity in terms of parameter size, giga floating-point operations per second (GFOLPs), and frames per second (FPS).

D. Ablation Study

In this section, we perform a series of ablation experiments on the DOTA dataset to comprehensively validate the effectiveness of our proposed method.

TABLE II
COMPARISON RESULTS OF PARAMETERS, GFLOPS, CLASS-WISE AP, AND MEAN AP mAP IN DIFFERENT BASELINE AND THE PROPOSED CRKD-YOLO

Method	Pl	Bd	Br	Gft	Sv	Lv	Sh	Tc	Bc	St	Sbf	Ra	Ha	Sp	Hc	mAP	Params	GFLOPs
YOLOv3	89.06	83.21	50.76	49.60	77.31	66.71	79.09	90.52	74.88	83.14	45.07	63.63	74.43	79.11	48.96	70.37	61.57M	49.53G
YOLOv3-tiny	82.29	62.85	31.11	34.35	47.24	50.58	53.95	90.45	59.10	56.98	22.32	37.08	61.72	66.98	27.15	52.28	8.69M	**4.14G**
YOLOv4	**89.64**	82.92	**52.11**	55.45	78.35	68.46	79.28	90.68	75.07	**84.40**	39.71	63.39	75.00	80.04	42.96	70.50	52.55M	38.15G
YOLOv5s	89.18	76.39	43.17	51.63	76.44	65.00	78.42	90.63	69.62	77.23	37.36	58.96	74.36	79.99	40.59	67.26	7.05M	5.08G
YOLOv5m	89.36	79.94	49.14	50.42	78.44	67.88	79.02	90.71	73.07	82.22	40.12	62.34	75.84	78.87	47.90	69.68	20.91M	15.37G
YOLOv5l	89.59	80.19	50.38	50.07	78.66	67.77	79.05	90.63	73.09	83.04	**47.16**	61.79	**76.54**	79.86	45.73	70.24	46.18M	34.52G
YOLOv5x	89.57	**83.32**	50.99	54.89	78.79	68.41	79.42	90.67	74.62	83.99	44.39	61.42	76.50	78.30	**49.83**	71.01	86.27M	65.30G
YOLOv7	88.12	78.30	46.30	**58.28**	73.82	70.14	79.74	90.67	**76.04**	83.21	39.91	54.18	74.68	77.70	45.88	69.13	36.55M	33.09G
CRKD-YOLO	89.61	79.87	51.14	57.12	**79.72**	**71.05**	**79.77**	**90.72**	76.01	84.24	41.88	**64.59**	76.09	**80.33**	49.52	**71.45**	**5.49M**	19.68G

1) Selection of the Baseline Framework: Our purpose is to design a universal framework for various detectors to improve their performance in detecting low-resolution images. Therefore, as shown in Table II, to select a suitable baseline, we compared the model size, inference cost, and accuracy under low-resolution input conditions for widely used YOLO series models, including YOLOv3, YOLOv3-tiny, YOLOv4, various scales of YOLOv5, and YOLOv7.

It can be seen that larger models bring higher accuracy among the baseline networks, but parameters and GFLOPs multiply. Despite achieving the highest accuracy among all baselines, the YOLOv5x's parameters and GFLOPs are more than 12 times higher than the lightweight YOLOv5s. For YOLOv5s, its mAP is slightly lower than other larger models, such as YOLOv3, YOLOv4, larger scale YOLOv5, and updated YOLOv7, but the number of parameters and GFLOPs is much lower than them. Compared with lightweight YOLOv3-tiny, the mAP of YOLOv5s is far ahead by 14.98%, and it has a lower number of model parameters (7.05 M versus 8.69 M), only 0.94G (5.08G versus 4.14G) more computations. Therefore, in practical applications, YOLOv5s can meet the requirements of real-time target detection well and is more convenient to deploy. At the same time, it needs to be emphasized that our framework can be migrated to other detection models, and the choice of the more lightweight YOLOv5s can also reduce experimental costs. These demonstrate the appropriateness for the choice of YOLOv5s as our baseline detection model.

2) Impact of Various Network Stride: As described previously in Section III-B, a larger stride results in smaller spatial dimensions of the features extracted by the model, which may lead to a loss of detailed information. To investigate the impact of reducing network stride on detection performance for low-resolution remote sensing images, we apply the method outlined in Section III-B to decrease the stride, thereby enlarging the feature sizes. Experiments are conducted on four YOLOv5 models of different scales. As shown in Table III, reducing the stride leads to significant improvements in detection accuracy. The mAP on the test set increases by 1.37% for YOLOv5s, 1.62% for YOLOv5m, 2.62% for YOLOv5l, and 2.01% for YOLOv5x. This improvement can be attributed to the reduced stride in convolution, which enlarges the feature size and refines the feature representation, making it particularly beneficial for detecting low-resolution images.

TABLE III
IMPACT OF REPLACING THE FIRST CBS WITH A SMALLER STRIDE CBS. "$s = 2$" REPRESENTS THE ORIGINAL MODEL AND "$s = 1$" REPRESENTS THAT THE CBS IS REPLACED

Method		Params	GFLOPs	mAP
YOLOv5s	s=2	7.05M	5.08G	67.26
	s=1	7.04M	19.64G	68.63 (+1.37)
YOLOv5m	s=2	20.90M	15.37G	69.68
	s=1	20.90M	60.47G	71.30 (+1.62)
YOLOv5l	s=2	46.18M	34.52G	70.24
	s=1	46.17M	136.73G	72.86 (+2.62)
YOLOv5x	s=2	86.26M	65.30G	71.01
	s=1	86.26M	259.49G	73.02 (+2.01)

However, halving the network stride doubles the features size, leading to an almost fourfold increase in computational cost. This is comparable to the expense of detecting high-resolution images prior to stride changes in the model. Nevertheless, by incorporating subsequent improvements to both the feature fusion module and distillation framework, we can make the detection of low-resolution images achieve more excellent accuracy with fewer model parameters and computational costs than detecting high-resolution images on the baseline network.

To illustrate this effect more intuitively, the features of $C_1 \sim C_5$ in the backbone network are averaged over the channel dimension, and the results are visualized in Fig. 5. Comparing the pairwise features at the first column (a) and (f), we can see that the features extracted from the C_1 would have clearer target outlines and details if we replace it by a convolution with smaller stride. In addition, although the subsequent modules do not change at all, the features from $C_2 \sim C_5$ also exhibit more refined texture due to the the larger size of their input feature.

3) Study of CRKD: This section presents the effects of different distillation methods. We use the method described in Section III-B and select YOLOv5s as the baseline network for CRKD.

First, we scrutinize the effect of feature-level and output-level distillation in Table IV. Since the spatial consistency between F_{i+1}^t and F_i^s is guaranteed, the L1 loss between F_{i+1}^t and F_i^s can be used to distill different positions by changing i. This allows exploration of which features are most suitable for distillation. When the distillation loss is calculated using all the output features from each layer in the entire network

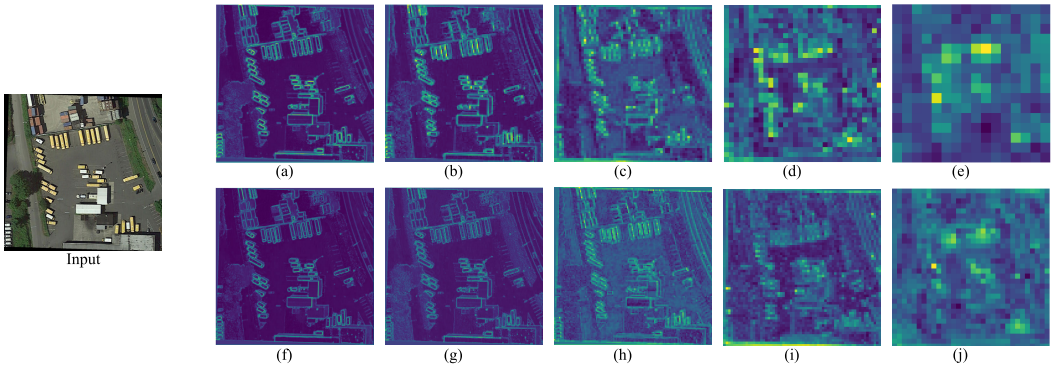

Fig. 5. Feature visualization of backbone for YOLOv5s under two kinds of convolutions with different strides as C_1. (a)–(e) Output features of $C_1 \sim C_5$ in original YOLOv5s. (f)–(j) Output features of $C_1 \sim C_5$ after the C_1 is replaced by the convolution with smaller stride. Note that the resolution of (a)–(e) and (f)–(j) are different. We have fixed them to the same size for viewing.

TABLE IV
COMPARISON OF DIFFERENT DISTILLATION POSITIONS. "BASELINE" MEANS TO TRAIN THE STUDENT BRANCH SEPARATELY IN THIS TABLE

Position	Feature-level					Output-level	Output and F_l	Baseline
	All	Backbone	Neck	F_s+F_m+F_l	F_l			
mAP	62.93 (-5.70)	68.10 (-0.53)	68.26 (-0.37)	67.53 (-1.10)	**70.75** (+2.12)	70.05 (+1.42)	70.49 (+1.86)	68.63

(backbone and neck), solely in the backbone, or solely in the neck, the mAP decreases by 5.7%, 0.53%, and 0.37%, respectively, compared to the baseline. In addition, we also try to calculate distillation loss using the three features F_s, F_m, and F_l, which are used for decoding, and the mAP also decreases by 1.10%. The reason for the decrease in accuracy may be that the distillation loss demands the network to maintain consistency across multiple features, which greatly increases the training difficulty of the network, making it hard to optimize for the detection task. In contrast, calculating the distillation loss through F_l that is also the last layer feature of model, the outputs of the network, or both of the F_l and the outputs together, the trained model achieves better performance, improving the detection accuracy by 2.12%, 1.42%, and 1.86% in mAP, respectively. The cause of this case is that the inner feature F_l is usually more informative than the outputs [61], making its distillation via F_l superior. Distilling by F_l and outputs leads to the same challenges as multilayer features, complicating training and diminishing the resulting improvements. Consequently, our framework uses only F_l for distillation loss calculation.

Then, after identifying the specific feature employed for distillation, we conduct the distillation process using three distinct approaches to examine the impact of online and offline distillation as well as the influence of employing L1 and MSE as loss functions. As shown in Table V, the best result is achieved with L1 for online distillation, improving mAP by 2.12%. Comparing with Table V(b), we can see that the L1 loss is 0.03% higher than the MSE in online distillation, and comparing with Table V(c), when L1 is selected as the distillation loss, the method of online is 0.1% higher than offline. The minor differences in the effects mentioned above demonstrate that our CRKD method is highly stable to changes in the distillation methods.

TABLE V
COMPARISON OF DISTILLATION METHODS

	Method	MSE	L1	Offline	Online	mAP
(a)	Baseline					68.63
(b)	Distill + MSE + Online	✓			✓	70.72
(c)	Distill + L1 + Offline		✓	✓		70.65
(d)	Distill + L1 + Online		✓		✓	**70.75**

TABLE VI
EFFECT OF CRKD IN IMAGE WITH VARIOUS RESOLUTIONS

Method	Train	Test	Params	GFLOPs	mAP
YOLOv5s	256	256	7.05M	1.27G	48.80
CRKD	512	256	7.04M	4.93G	65.08 (+16.28)
	1024	256	7.04M	19.53G	66.91 (+18.11)
YOLOv5s	512	512	7.05M	5.08G	67.26
CRKD	1024	512	7.04M	19.64G	70.75 (+3.49)

Moreover, we evaluate the impact of our CRKD on images with different resolutions by using relatively high-resolution images (scaled by $\times 2$ and $\times 4$) to supervise the model training for low-resolution images. As shown in Table VI, when a $\times 2$ higher-resolution image with a size of 512 is used as the teacher network input to train a student model for low-resolution images fixed at size 256, the mAP improves by 16.28% (48.80% versus 65.08%) compared to training and testing low-resolution images on the baseline YOLOv5s model. When $\times 4$ images with a size of 1024 is used as the teacher input, the mAP further increases to 66.91%. Similarly, for student inputs of size 512, the performance improves by 3.49% when supervised by a teacher network with input size fixed to 1024. However, increasing the feature size of the student network's low-resolution inputs to retain more information and align them with the high-resolution

TABLE VII

EFFECT OF CRKD IN DIFFERENT BASELINES. DUE TO LIMITED COMPUTING RESOURCES, THE BATCH SIZE OF THE EXPERIMENTS IN THIS TABLE IS SET TO 2 DURING THE TRAINING PROCESS

Method		Params	GFLOPs	mAP
YOLOv3 [31]	HR	61.57M	198.11G	72.96
	LR	61.57M	49.53G	69.07
	CRKD	61.57M	197.43G	72.02 (+2.95)
YOLOv4 [32]	HR	52.54M	152.58G	72.68
	LR	52.54M	38.15G	67.76
	CRKD	52.54M	151.90G	71.21 (+3.45)
YOLOv5s [33]	HR	7.05M	20.32G	70.26
	LR	7.05M	5.08G	63.07
	CRKD	7.04M	19.64G	69.76 (+6.69)
YOLOv5m [33]	HR	20.90M	61.49G	73.14
	LR	20.90M	15.37G	66.12
	CRKD	20.90M	60.47G	71.89 (+5.77)
YOLOv5l [33]	HR	46.18M	138.09G	73.49
	LR	46.18M	34.52G	68.57
	CRKD	46.17M	136.73G	73.15 (+4.58)
YOLOv5x [33]	HR	86.26M	261.46G	73.22
	LR	86.26M	65.37G	69.10
	CRKD	86.26M	259.49G	73.39 (+4.26)

features of the teacher network results in higher computational costs during inference. Consequently, the GFLOPs approach those of the teacher network processing high-resolution inputs. For example, testing 256-size images guided by training inputs with the size of 1024 requires 19.53 GFLOPs, compared to 4.93 GFLOPs when the training size is reduced to 512. This approach offers flexibility, allowing the choice of teacher network input size to be adjusted based on the available computational resources. In summary, our CRKD demonstrates a more pronounced improvement in accuracy for lower-resolution images but comes with a trade-off in computational cost.

Eventually, as shown in Table VII, there are three conditions to compare the performance of baselines with various scales, as follows: 1) training and testing by high-resolution images (HR); 2) training and testing by low-resolution images (LR); and 3) testing low-resolution images via the model trained by our CRKD. It is obvious that the accuracy on low-resolution images significantly decreases compared to high-resolution on various models. Compared to the baseline for detecting low-resolution images, the accuracy is improved by distilling knowledge from high-resolution teacher networks, demonstrates superior performance in mAP. Specifically, the following are observed. YOLOv3 increases by 2.95%; YOLOv4 increases by 3.45%; YOLOv5s increases by 6.69%; YOLOv5m increases by 5.77%; YOLOv5l increases by 4.58%; YOLOv5x increases by 4.26%. Although our CRKD almost quadruples the GFLOPs compared to "LR," the accuracy is almost equivalent to "HR," and parameters and computation burden are lower. This means that our CRKD achieves comparable detection accuracy to high-resolution images at a reasonable inference cost.

4) Ablation of BAFPN: As shown in Table VIII, we conduct experiments on different-scale backbones and remote sensing images with various resolutions. The experimental results demonstrate that our proposed BAFPN offers universal advantages. Specifically, compared to FPN, PAFPN

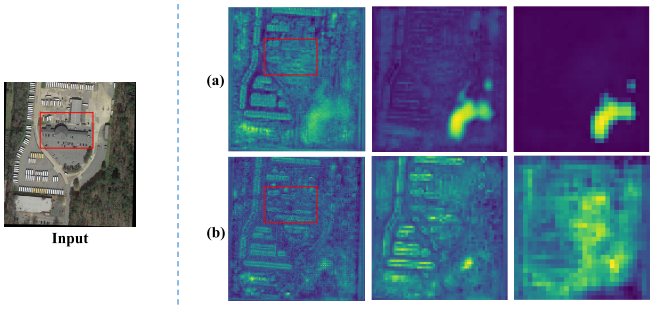

Fig. 6. Feature visualization of F_s, F_m, F_l in PAFPN and BAFPN. (a) PAFPN. (b) BAFPN. The order of the features is F_s, F_m, and F_l from left to right. Note that the features within (a) and (b) are fixed to the same size for viewing.

achieves superior performance across various backbone scales and input resolutions, attributed to the incorporation of the path aggregation network. Compared to PAFPN, the proposed BAFPN generally achieves higher accuracy. For the YOLOv5s backbone, BAFPN improves mAP metrics by 2.51%, 1.52%, and 1.33% compared to FPN, and by 1.26%, 0.64%, and 0.44% compared to PAFPN, at image sizes of 256, 512, and 1024, respectively. For the YOLOv5m backbone, BAFPN achieves increases of 0.67%, 0.66%, and 0.67% compared to FPN, and 0.46%, 0.23%, and 0.17% compared to PAFPN, across the same three sizes. For the YOLOv5l backbone, BAFPN improves detection accuracy by 1.45%, 0.54%, and 0.49% compared to FPN at scales of 256, 512, and 1024, respectively. Compared to PAFPN, it achieves improvements of 0.73% and 0.77% at scales 256 and 512, but shows a slight decrease of 0.5% at the higher resolution scale of 1024. From these results, it is evident that BAFPN outperforms both FPN and PAFPN in most cases, with the accuracy improvement being especially pronounced at lower image resolutions.

In terms of model parameters, our BAFPN is lighter than both FPN and PAFPN, making it more suitable for deployment. For example, when the backbone is YOLOv5s, the mAP metric surpasses both FPN and PAFPN across three different image resolutions, and the model parameters are lower: 6.01 M for FPN, 7.05 M for PAFPN, and 5.49 M for BAFPN. Regarding computational complexity, FPN demonstrates the lowest GFLOPs across all backbones and resolutions. While BAFPN shows a slight increase in computational cost compared to PAFPN on smaller YOLOv5s models, it exhibits a lower computational cost than PAFPN on larger YOLOv5m and YOLOv5l models.

To more intuitively illustrate the effect of our BAFPN, we construct two models with PAFPN and BAFPN, respectively, after reducing the stride of the backbone, and present their F_l, F_m, and F_s averaged over the channel dimension in Fig. 6. It can be seen that although the F_s of the PAFPN can retain the outline of large vehicles (white and yellow cars in the input image) well, there is almost no distinction among the small vehicles in the red frame. Furthermore, the F_m of BAFPN also shows the excellence of texture information extraction, and the F_l of BAFPN obviously contains more information in the ROI than that of PAFPN. The visualization results above indicate that, compared to PAFPN, the proposed

TABLE VIII
EFFECT OF FPN, PAFPN, AND BAFPN ON BACKBONE WITH DIVERSE SIZES AND IMAGES WITH VARIOUS RESOLUTIONS

Backbone	Size	Neck	Params	GFLOPs	mAP
YOLOv5s	256	FPN	6.01M	1.17G	47.51
		PAFPN	7.05M	1.27G	48.80
		BAFPN	5.49M	1.27G	**50.06**
	512	FPN	6.01M	4.68G	66.38
		PAFPN	7.05M	5.08G	67.26
		BAFPN	5.49M	5.09G	**67.90**
	1024	FPN	6.01M	18.71G	70.26
		PAFPN	7.05M	20.32G	71.15
		BAFPN	5.49M	20.36G	**71.59**
YOLOv5m	256	FPN	18.21M	3.52G	52.39
		PAFPN	20.91M	3.85G	52.60
		BAFPN	15.86M	3.79G	**53.06**
	512	FPN	18.21M	14.09G	69.24
		PAFPN	20.91M	15.37G	69.67
		BAFPN	15.86M	15.14G	**69.90**
	1024	FPN	18.21M	56.36G	72.50
		PAFPN	20.91M	61.49G	73.01
		BAFPN	15.86M	60.57G	**73.18**
YOLOv5l	256	FPN	40.74M	7.89G	53.27
		PAFPN	46.18M	8.63G	53.99
		BAFPN	34.44M	8.42G	**54.72**
	512	FPN	40.74M	31.57G	70.30
		PAFPN	46.18M	34.52G	70.07
		BAFPN	34.44M	33.66G	**70.84**
	1024	FPN	40.74M	126.28G	73.20
		PAFPN	46.18M	138.09G	**74.19**
		BAFPN	34.44M	134.65G	73.69

BAFPN is more effective at enhancing detailed feature awareness.

5) Ablation Experiment of All Components: To further assess the efficacy of our optimizations over CRKD-YOLO, we conduct a comprehensive ablation experiment by progressively incorporating or removing each of our improved modules. The detailed results can be found in Table IX.

First, the results of Table IX (a) and (b) indicate that training and testing YOLOv5s on high-resolution images achieves an accuracy of 71.15% in mAP, whereas using low-resolution images results in a lower accuracy of 67.26%. The results demonstrate that, within the same detector, a reduction in resolution by half leads to a significant decrease of 3.89% in mAP, highlighting the negative impact of resolution degradation on accuracy.

Then, as shown in Table IX(c)–(f), to improve the detection performance of low-resolution images, we add the proposed improvements one by one. After reducing the network stride to enlarge features, mAP increases by 1.37% (67.26% versus 68.63%), but the enlarged feature size increases in computational cost (5.08G versus 19.64G). On this basis, using BAFPN improves mAP by 0.69% (68.63% versus 69.32%) and reduces model complexity by 20% (7.04 M versus 5.49 M), at a cost of 0.04 (19.64G versus 19.68G) increase in computational expense. We next execute CRKD, making mAP markedly increase 1.91% (69.32% versus 71.23%) without more parameters and GFLOPs. In addition, the way we use CBAM to prompt the teacher network to concentrate on key regions and channels subsequently exerts an influence on the student branch through distillation, which makes the mAP of student improve by 0.22% (71.23% versus 71.45%). As the CBAM is only introduced in the teacher network, it will not bring any additional parameters and computational complexity to the student model. It can be seen from (e) and (f) that because of the enhanced feature by BAFPN and guidance of high-resolution knowledge, the mAP of detecting low-resolution images reaches 71.23% and 71.45%, which exceeds the accuracy of baseline in training and testing on high-resolution images (71.15% in mAP), and the number of model parameters (5.49 versus 7.05 M) and GFLOPs (19.68 versus 20.32 G) are also lower.

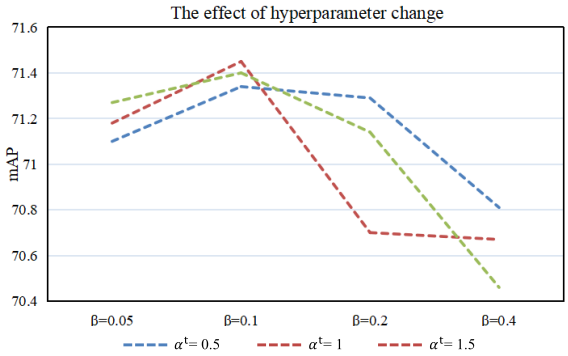

Fig. 7. Performance change with respect to coefficient α^t and β. The α^s is fixed to 1.

Finally, from Table IX(g) and (h), we can see that, based on the entire CRKD-YOLO, we removed BAFPN from both the teacher and student networks and switched to the original PAFPN, resulting in the mAP decreased by 0.48% (71.45% vs. 70.97%), indicating that BAFPN is still beneficial under the cross-resolution distillation framework. Furthermore, the exclusion of the CBAM module from the teacher network resulted in a decrease of 0.22% in mAP (70.97% vs. 70.75%), thereby validating our approach of incorporating the attention module for both PAFPN and BAFPN within the distillation framework.

These ablation experiments clearly demonstrate the value of our improvements, all of which can support object detection in low-resolution remote sensing images.

6) Parameter Sensitivity Analysis: To evaluate the impact of coefficient changes in the overall loss function (11) on model training within our framework, we adjust α^t and β while keeping α^s fixed at 1. Specifically, as shown in Fig. 7, we set α_t to 0.5, 1, 1.5, and then vary β to investigate how these changes influence model training. The results show that when β is small, increasing β appropriately improves performance. Conversely, when the β is bigger than 0.1, as β increases, the accuracy improves less and less. The model achieves the best performance with $\alpha^t = 1$ and $\beta = 0.1$. Therefore, we adopted this setting in our experiments.

7) Influence on Targets With Varying Scales: For low-resolution images, enhancing the detection performance of small targets is crucial. Therefore, we analyze the impact of our CRKD-YOLO on the detection performance of targets at different scales. The results demonstrate that our method significantly improves the detection accuracy of small targets. It is important to note that target dimensions are mapped to their corresponding high-resolution images (e.g., an object

TABLE IX
ABLATION EXPERIMENT OF ALL COMPONENTS IN CRKD-YOLO

Method	Baseline	Train size	Test size	Small stride	BAFPN	CRKD	CBAM	Params	GFLOPs	mAP
(a)	YOLOv5s	1024	1024					7.05M	20.32G	71.15
(b)		512	512					7.05M	5.08G	67.26
(c)		512	512	✓				7.04M	19.64G	68.63
(d)		512	512	✓	✓			5.49M	19.68G	69.32
(e)		1024	512	✓	✓	✓		5.49M	19.68G	71.23
(f)		1024	512	✓	✓	✓	✓	5.49M	19.68G	**71.45**
(g)		1024	512	✓		✓	✓	7.04M	19.64G	70.97
(h)		1024	512	✓		✓		7.04M	19.64G	70.75

Fig. 8. Comparison of performance on varying size targets between our CRKD-YOLO and baseline YOLOv5s. "HR" refers to training and testing the baseline model on high-resolution images and "LR" represents the baseline model trained and tested on low-resolution images.

with a length of 16 on a low-resolution image is represented with a length of 32 on a high-resolution image).

Specifically, as shown in Fig. 8, the detection performance for objects with fewer than 48^2 pixels is significantly improved, with mAP increasing by approximately 9% compared to "LR." For targets with pixel counts between 48^2 and 256^2, a noticeable improvement is also observed when detecting low-resolution images. However, for targets larger than 256^2, the detection performance of "LR" surpasses that of "HR." This may be attributed to the insufficient receptive field of the convolution network, which limits the detection accuracy of "HR" on such large targets. Overall, our method achieves a better balance in handling variations in target sizes within the images. For targets ranging from 0^2 to 256^2, the detection accuracy is comparable to that of "HR." For extremely large targets (sizes greater than 256^2), our method effectively mitigates the precision degradation commonly observed with "HR."

E. Comparisons With Previous Methods

As listed in Table II, in comparison to the YOLO series of algorithms, we achieve superior accuracy in more class-wise AP, and the mAP also outperforms the large-scale models YOLOv3, YOLOv4, and YOLOv5x while utilizing only approximately one-tenth of their parameters. Compared to the updated YOLOv7, we also have great advantages in both precision and complexity.

Currently, although there are many high-quality datasets, the effects of models trained on high-resolution images do not generalize to low-resolution images. And most existing algorithms first crop the images in the dataset to a certain size and then test the performance of the model, so that their spatial resolution matches the high-resolution images in the dataset. We resize the images in the DOTA, DIOR, and NWPU datasets to a smaller scale to simulate the low-resolution images, and compare the test results of our method with other advanced detection algorithms. As shown in Table X, to prove the superiority of CRKD-YOLO, we compare 12 typical detection algorithms, including one-stage detector: RetinaNet [37], GFL [62], FCOS [63], ATSS [64]; two-stage algorithm: Faster R-CNN [7]; light models: MobileNetV2 [65] and ShuffleNet [66]; distillation-based ARSD [54]; methods specifically designed for remote sensing images: FMSSD [67] and O2DNet [68]; SuperYOLO [18] utilizing the assisted super-resolution network to train detection model and ESR-GAN as a preprocessing.

It can be seen that our algorithm, respectively, achieves the mAP of 71.45%, 76.09%, and 93.45% on the DOTA, DIOR, and NWPU datasets even under the low-resolution input condition. Not only does it significantly outperform existing methods in accuracy, but it also exhibits far lower model complexity compared to the aforementioned one- or two-stage object detectors. Compared to ARSD [54], which employs distillation for model lightweighting, our method achieves higher precision across the three datasets while using less than half the parameters and requiring less than one-third of the computational cost. Compared to the lightweight models MobileNetV2 [65] and ShuffleNet [66], our method uses approximately half the parameters, less than one-sixth of the computational cost, and achieves over 10% higher mAP accuracy.

For FMSSD [67] and O2DNet [68], which are specialized in remote sensing detection, their accuracy matches ours on

TABLE X
PERFORMANCE OF DIFFERENT ALGORITHMS ON DOTA, DIOR, AND NWPU TESTING SET

Method	DOTA			DIOR			NWPU		
	Params	GFLOPs	mAP	Params	GFLOPs	mAP	Params	GFLOPs	mAP
Faster R-CNN [7]	60.19M	289.25G	60.64	60.21M	182.20G	54.10	41.17M	127.70G	77.80
RetainNet [37]	55.39M	293.36G	50.39	55.49M	180.62G	65.70	36.29M	123.27G	89.40
GFL [62]	19.13M	159.18G	66.53	19.13M	97.43G	68.00	19.13M	91.73G	88.80
FCOS [63]	31.57M	202.15G	67.72	31.88M	123.51G	67.60	31.86M	116.63G	89.65
ATSS [64]	18.97M	156.01G	66.84	18.98M	95.50G	67.70	18.96M	89.90G	90.95
MobileNetV2 [65]	10.30M	124.24G	56.91	10.32M	76.10G	58.20	10.29M	71.49G	76.90
ShuffleNet [66]	12.11M	142.60G	57.73	12.12M	87.31G	61.30	12.10M	82.17G	83.00
O2-DNet [68]	209.00M	-	71.10	209.00M	-	68.30	-	-	-
FMSSD [67]	136.04M	-	**72.43**	136.03M	-	69.50	-	-	-
ARSD [54]	13.08M	68.03G	68.28	13.10M	41.60G	70.10	11.57M	26.65G	90.92
ESRGAN+YOLOv5s [69]	23.75M	1195.51G	68.64	23.77M	729.72G	75.99	23.74M	1195.46G	89.73
SuperYOLO [18]	7.70M	20.89G	69.99	7.70M	20.93G	71.82	7.68M	20.86G	93.30
CRKD-YOLO	**5.49M**	**19.68G**	71.45	**5.49M**	**12.04G**	**76.09**	**5.48M**	**19.64G**	**93.45**

TABLE XI
PERFORMANCE IMPROVEMENT OF OUR METHOD ON DOTA, DIOR, NWPU, DRONEVEHICLE, AND VEDAI DATASETS. "HR" REFERS TO TRAINING AND TESTING THE MODEL ON HIGH-RESOLUTION IMAGES. "LR" REPRESENTS THE MODEL TRAINED AND TESTED ON LOW-RESOLUTION IMAGES

Method		DOTA		DIOR		NWPU		DroneVehicle		DroneVehicle-IR		VEDAI		VEDAI-IR	
		FPS	mAP	FPS	mAP	FPS	mAP	FPS	mAP	FPS	mAP	FPS	mAP	FPS	mAP
YOLOv5s	**HR**	163	71.15	232	77.10	120	90.25	238	74.53	250	79.83	169	73.68	159	65.98
	LR	227	67.26	250	74.10	167	88.21	263	66.98	256	73.05	238	57.19	232	48.52
CRKD-YOLO	**-**	170	71.45	217	76.09	139	93.45	238	74.05	238	78.98	169	72.40	175	68.14

DOTA, but their extensive parameters complicate deployment and inevitably hamper detection speed. In contrast, our CRKD-YOLO has great advantages in terms of lightweight and outperforms them on the DIOR dataset, proving its effectiveness. We also compare different approaches that introduce super-resolution to assist in low-resolution image detection, including utilizing the super-resolution network ESRGAN [69] as a preprocessing step and the SuperYOLO [18] framework. Specifically, in the "ESRGAN + YOLOv5s" paradigm, the super-resolution images generated by ESRGAN are used as input for YOLOv5s. The ESRGAN network is optimized on a joint training set of DOTA and DIOR. SuperYOLO [18], on the other hand, incorporates a super-resolution decoder in the training phase using a joint learning paradigm. The results demonstrate that using a super-resolution model as a preprocessing step significantly increases model parameters and computational cost, exceeding 50 times the computational overhead of our method. SuperYOLO [18] avoids the additional burden of a standalone super-resolution network while achieving higher accuracy. Notably, our method outperforms super-resolution-assisted approaches in terms of model complexity and performance.

In general, compared to the aforementioned algorithms, our method significantly enhances detection performance for low-resolution inputs, achieving higher accuracy than other advanced detectors while maintaining a better balance between detection cost and accuracy.

F. Generalization Analysis

To evaluate the generalization capability of our method, we conducted experiments on multiple datasets, including DOTA, DIOR, NWPU, VEDAI, and DroneVehicle. These datasets cover a wide range of resolutions, scenes, targets, and sources.

1) Generalization for Diverse Datasets: The visualization results of our CRKD-YOLO on the above datasets are presented in Fig. 9. From these results, it is evident that our method achieves excellent detection performance on both large-scale and small-scale datasets, effectively handling objects of various sizes. Additionally, Fig. 9(e) demonstrates that our method can successfully detect occluded targets and performs well in shallow night scenes captured from a drone's perspective.

The quantitative analysis of the detection accuracy and inference speed of our CRKD-YOLO is introduced by comparing its performance with both "HR" and "LR" on the baseline network. The results highlight the advantages of our approach in balancing detection accuracy and computational efficiency across various datasets and modalities, albeit at the cost of reduced inference speed compared to processing low-resolution images directly. Specifically, as shown in Table XI, our method significantly improves the detection accuracy of low-resolution images across five datasets compared to the baseline network. On single-visible datasets (DOTA, DIOR, and NWPU), the mAP metrics of low-resolution images are improved by 4.19%, 1.99%, and 5.24%, respectively. Notably, the performance on DOTA and NWPU even exceeds the accuracy of the "HR." On multimodal datasets (DroneVehicle and VEDAI), our method significantly enhances detection performance for both low-resolution infrared and visible images. For DroneVehicle, the detection accuracy increases by 7.07% for visible images and 2.93% for

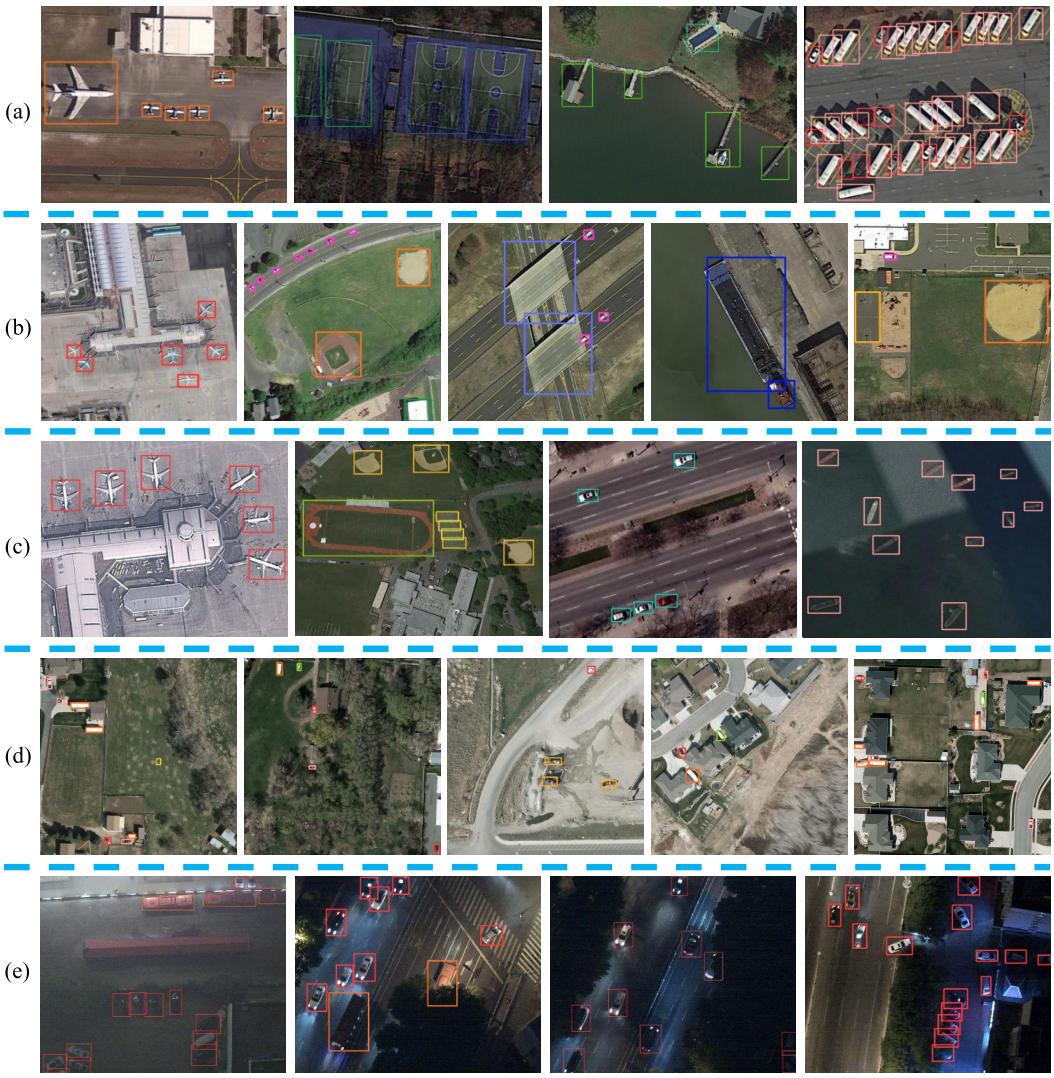

Fig. 9. Visualization results of our CRKD-YOLO on (a) DOTA, (b) DIOR, (c) NWPU, (d) VEDAI, and (e) DroneVehicle.

infrared images. On the VEDAI dataset, the improvements are even more substantial due to the prevalence of smaller targets in this dataset (as shown in Figs. 9(d) and 10), with accuracy gains of 15.21% for visible images and 19.62% for infrared images. These findings further demonstrate that our method is particularly effective in enhancing the detection performance of small targets.

2) Generalization for Noise Images: As shown in Fig. 10, we evaluate our method on low-resolution noisy images under two conditions: (a) training the model on the original dataset and (b) training the model on a noise dataset. When detecting low-resolution images with varying degrees of Gaussian noise added, using the model trained on the original VEDAI dataset (without noise), the results show that our method can accurately locate and classify targets under noise-free conditions. However, as the noise level increases, the detection performance progressively deteriorates, ultimately making it difficult to locate and identify targets, as illustrated in Fig. 10(a). In contrast, as illustrated in Fig. 10(b), this issue is effectively alleviated by introducing Gaussian noise with σ randomly varying between 0 and 30 during training. This demonstrates the generalization capability of our method when handling noisy images.

Overall, our approach provides a generalizable scheme to enhance the detection accuracy of low-resolution remote sensing images from various sources and scenes, achieving accuracy comparable to training and testing high-resolution images on baseline networks. This improvement comes at the cost of reducing the detection efficiency of low-resolution images to a level similar to that of high-resolution images.

G. Limitation

Although our CRKD-YOLO demonstrates significant improvements in detecting low-resolution remote sensing images, it still has the following limitations that require further refinement.

1) Suboptimal detection in extremely low-light scenes: The detection performance of our method on low-resolution images is less effective in extremely low-light night scenes. As shown in Fig. 11, many targets are missed in these challenging conditions due to the extremely low contrast between the target and the

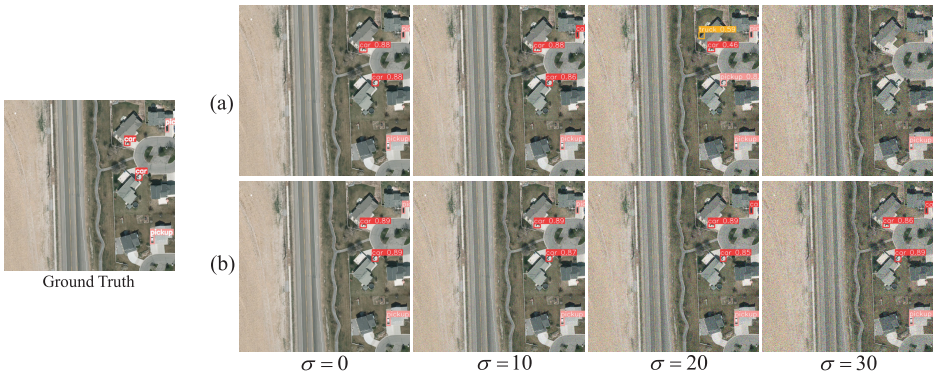

Fig. 10. Detection performance of our CRKD-YOLO on low-resolution images with various degrees of noise. (a) Results detected by the model trained on the original VEDAI dataset. (b) Results detected by the model trained on the VEDAI dataset added with Gaussian noise. The image from VEDAI dataset are fixed at 512 × 512. σ refers to the noise degree.

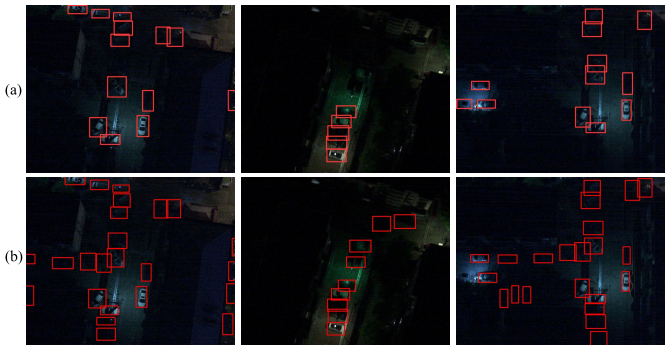

Fig. 11. Detection performance of our CRKD-YOLO on low-resolution images from an extremely low-light scene. The images are from the DroneVehicle dataset. (a) Detection results. (b) Corresponding ground truth.

background. Enhancing image contrast or incorporating infrared imaging may address the issue of reduced target recognition accuracy caused by insufficient lighting at night.

2) *Increased cost for low-resolution image detection:* While our method effectively improves the detection accuracy of low-resolution images and achieves a more lightweight design, the smaller network stride significantly increases the computational cost compared to detecting low-resolution images, making it nearly as high as that of the baseline network for detecting high-resolution images. Additionally, the two-branch distillation structure doubles the training cost, presenting challenges in computing resources. These weaknesses could be mitigated through further optimization of the model structure and feature alignment measurement.

V. Conclusion and Future Work

In this article, we propose CRKD-YOLO to tackle the challenge of detecting low-resolution remote sensing images. It optimizes the model from two main aspects: the guidance of high-resolution information and the improvement of the feature fusion module. We first enlarge the feature size of low-resolution images, which helps retain more information and serves as a prerequisite for distilling knowledge from the high-resolution teacher network. Sequentially, we construct CRKD to resolve the issues of aligning features across diverse input sizes by adjusting the stride of the student and teacher networks, thus distilling high-resolution image information into the student model for detecting low-resolution images. Moreover, we propose BAFPN to improve the feature fusion module, which makes more use of the low-level texture information to improve the attention to detail feature and reduce the model parameters. With these optimizations, the proposed CRKD-YOLO achieves mAP scores of 71.45%, 76.09%, and 93.45% on the DOTA, DIOR, and NWPU datasets, respectively, along with significant improvements on multisource datasets such as DroneVehicle and VEDAI when testing on low-resolution images. Compared to training and testing high-resolution images with the baseline, CRKD-YOLO delivers comparable or higher accuracy while reducing model complexity.

However, as discussed in Section IV-G, the generalization of our CRKD-YOLO in extremely low-light scenes remains inadequate. Additionally, compared to the baseline, the higher inference and training costs need further optimization. In future work, we will focus on reducing these excessive costs, while improving performance in complex visual tasks. This includes handling a broader range of low-quality images, such as those affected by noise, blur, darkness, clouds, and haze, to better address the challenges posed by real-world scenarios.

References

[1] 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.*, Jun. 2018, pp. 3974–3983.

[2] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. 13th Eur. Conf. Comput. Vis. (ECCV)*, Zurich, Switzerland. Cham, Switzerland: Springer, 2014, pp. 740–755.

[3] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, "Object detection in optical remote sensing images: A survey and a new benchmark," *ISPRS J. Photogramm. Remote Sens.*, vol. 159, pp. 296–307, Jan. 2020.

[4] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands. Cham, Switzerland: Springer, Oct. 2016, pp. 21–37.

[5] 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. (CVPR)*, Jun. 2016, pp. 779–788.

[6] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, Dec. 2015, pp. 91–99.

[8] X. Zhang et al., "Remote sensing object detection meets deep learning: A metareview of challenges and advances," *IEEE Geosci. Remote Sens. Mag.*, vol. 11, no. 4, pp. 8–44, Dec. 2023.

[9] Y. Liu, Z. Xiong, Y. Yuan, and Q. Wang, "Distilling knowledge from super resolution for efficient remote sensing salient object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5609116.

[10] Z. Cui et al., "RestoreDet: Degradation equivariant representation for object detection in low resolution images," 2022, *arXiv:2201.02314*.

[11] Z. Li et al., "Deep learning-based object detection techniques for remote sensing images: A survey," *Remote Sens.*, vol. 14, no. 10, p. 2385, May 2022.

[12] M. Haris, G. Shakhnarovich, and N. Ukita, "Task-driven super resolution: Object detection in low-resolution images," in *Proc. Int. Conf. Neural Inf. Process.*, Sanur, Indonesia. Cham, Switzerland: Springer, 2021, pp. 387–395.

[13] F. Xiaolin et al., "Small object detection in remote sensing images based on super-resolution," *Pattern Recognit. Lett.*, vol. 153, pp. 107–112, Jan. 2022.

[14] 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, p. 1432, 2020.

[15] M. Mostofa, S. N. Ferdous, B. S. Riggan, and N. M. Nasrabadi, "Joint-SRVDNet: Joint super resolution and vehicle detection network," *IEEE Access*, vol. 8, pp. 82306–82319, 2020.

[16] 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, 2022, Art. no. 5628016.

[17] R. Ran, L.-J. Deng, T.-X. Jiang, J.-F. Hu, J. Chanussot, and G. Vivone, "GuidedNet: A general CNN fusion framework via high-resolution guidance for hyperspectral image super-resolution," *IEEE Trans. Cybern.*, vol. 53, no. 7, pp. 4148–4161, Jul. 2023.

[18] 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.

[19] X. Ou, L. Liu, B. Tu, G. Zhang, and Z. Xu, "A CNN framework with slow-fast band selection and feature fusion grouping for hyperspectral image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5524716.

[20] X. Ou, M. Wu, B. Tu, G. Zhang, and W. Li, "Multi-objective unsupervised band selection method for hyperspectral images classification," *IEEE Trans. Image Process.*, vol. 32, pp. 1952–1965, 2023.

[21] P. Wang, Y. Yang, Y. Xia, K. Wang, X. Zhang, and S. Wang, "Information maximizing adaptation network with label distribution priors for unsupervised domain adaptation," *IEEE Trans. Multimedia*, vol. 25, pp. 1–14, 2022.

[22] Y. Yang, Y. Rao, M. Yu, and Y. Kang, "Multi-layer information fusion based on graph convolutional network for knowledge-driven herb recommendation," *Neural Netw.*, vol. 146, pp. 1–10, Feb. 2022.

[23] Y. Li, F. Melgani, and B. He, "CSVM architectures for pixel-wise object detection in high-resolution remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 9, pp. 6059–6070, Sep. 2020.

[24] H. Wang, C. Liu, Y. Cai, L. Chen, and Y. Li, "YOLOv8-QSD: An improved small object detection algorithm for autonomous vehicles based on YOLOv8," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–16, 2024.

[25] T. Ye, W. Qin, Z. Zhao, X. Gao, X. Deng, and Y. Ouyang, "Real-time object detection network in UAV-vision based on CNN and transformer," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–13, 2023.

[26] Z. Ren, Y. Tang, Z. He, L. Tian, Y. Yang, and W. Zhang, "Ship detection in high-resolution optical remote sensing images aided by saliency information," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5623616.

[27] J. Li, R. Xu, J. Ma, Q. Zou, J. Ma, and H. Yu, "Domain adaptive object detection for autonomous driving under foggy weather," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 612–622.

[28] M. A. Abou-Khousa, M. S. U. Rahman, K. M. Donnell, and M. T. A. Qaseer, "Detection of surface cracks in metals using microwave and millimeter-wave nondestructive testing techniques—A review," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–18, 2023.

[29] Y. Yang et al., "A weighted multi-feature transfer learning framework for intelligent medical decision making," *Appl. Soft Comput.*, vol. 105, Jul. 2021, Art. no. 107242. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494621001654

[30] 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, Jul. 2023, Art. no. 108962.

[31] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[32] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[33] Ultralytics. (2022). *Ultralytics/YOLOv5: V6.2*, doi: 10.5281/zenodo.7347926. Accessed: May 7, 2023. [Online]. Available: https://github.com/ultralytics/yolov5.com

[34] Y. Dai, W. Liu, H. Wang, W. Xie, and K. Long, "YOLO-Former: Marrying YOLO and transformer for foreign object detection," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–14, 2022.

[35] S. Panigrahy and S. Karmakar, "Real-time condition monitoring of transmission line insulators using the YOLO object detection model with a UAV," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–9, 2024.

[36] W. Zhou et al., "An efficient tiny defect detection method for PCB with improved YOLO through a compression training strategy," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–14, 2024.

[37] 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. (ICCV)*, Oct. 2017, pp. 2980–2988.

[38] 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. (CVPR)*, Jun. 2014, pp. 580–587.

[39] T. Y. Lin, P. Dollàr, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2117–2125.

[40] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.

[41] L. Jiang et al., "MFFSODNet: Multiscale feature fusion small object detection network for UAV aerial images," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–14, 2024.

[42] C. Chen, M.-Y. Liu, O. Tuzel, and J. Xiao, "R-CNN for small object detection," in *Proc. Asian Conf. Comput. Vis.*, Taipei, Taiwan. Cham, Switzerland: Springer, 2017, pp. 214–230.

[43] 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 (CVPRW)*, Jun. 2019, pp. 1432–1441.

[44] Y. Wang et al., "Remote sensing image super-resolution and object detection: Benchmark and state of the art," *Expert Syst. Appl.*, vol. 197, Jul. 2022, Art. no. 116793.

[45] Z. Cui et al., "Exploring resolution and degradation clues as self-supervised signal for low quality object detection," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 473–491.

[46] 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. (CVPR)*, Jun. 2020, pp. 3774–3783.

[47] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[48] F. Ruffy and K. Chahal, "The state of knowledge distillation for classification," 2019, *arXiv:1912.10850*.

[49] C. Shi, L. Fang, Z. Lv, and M. Zhao, "Explainable scale distillation for hyperspectral image classification," *Pattern Recognit.*, vol. 122, Feb. 2022, Art. no. 108316.

[50] Q. Zhang, X. Cheng, Y. Chen, and Z. Rao, "Quantifying the knowledge in a DNN to explain knowledge distillation for classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 5099–5113, Apr. 2023.

[51] G. Chen, W. Choi, Y. Xiang, T. X. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Dec. 2017, pp. 742–751.

[52] Q. Li, S. Jin, and J. Yan, "Mimicking very efficient network for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6356–6364.

[53] T. Wang, L. Yuan, X. Zhang, and J. Feng, "Distilling object detectors with fine-grained feature imitation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 4933–4942.

[54] Y. Yang et al., "Adaptive knowledge distillation for lightweight remote sensing object detectors optimizing," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5623715.

[55] J. Guo et al., "Distilling object detectors via decoupled features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2154–2164.

[56] Z. Yang et al., "Focal and global knowledge distillation for detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4643–4652.

[57] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 3–19.

[58] 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.

[59] Y. Sun, B. Cao, P. Zhu, and Q. Hu, "Drone-based RGB-infrared cross-modality vehicle detection via uncertainty-aware learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 10, pp. 6700–6713, Oct. 2022.

[60] S. Razakarivony and F. Jurie, "Vehicle detection in aerial imagery: A small target detection benchmark," *J. Vis. Commun. Image Represent.*, vol. 34, pp. 187–203, Jan. 2016.

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

[62] X. Li et al., "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 33, Dec. 2020, pp. 21002–21012.

[63] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9626–9635.

[64] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9759–9768.

[65] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[66] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.

[67] P. Wang, X. Sun, W. Diao, and K. Fu, "FMSSD: Feature-merged single-shot detection for multiscale objects in large-scale remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3377–3390, May 2020.

[68] H. Wei, Y. Zhang, Z. Chang, H. Li, H. Wang, and X. Sun, "Oriented objects as pairs of middle lines," *ISPRS J. Photogramm. Remote Sens.*, vol. 169, pp. 268–279, Nov. 2020.

[69] X. Wang et al., "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, Jan. 2019, pp. 63–79.