

YOLO-Ant: A Lightweight Detector via Depthwise Separable Convolutional and Large Kernel Design for Antenna Interference Source Detection

Xiaoyu Tang^{id}, *Member, IEEE*, Xingming Chen^{id}, Jintao Cheng^{id}, Jin Wu^{id}, *Member, IEEE*, Rui Fan^{id}, *Senior Member, IEEE*, Chengxi Zhang^{id}, *Member, IEEE*, and Zebo Zhou^{id}

Abstract—In the era of 5G communication, removing interference sources that affect communication is a resource-intensive task. The rapid development of computer vision has enabled unmanned aerial vehicles (UAVs) to perform various high-altitude detection tasks. Because the field of object detection for antenna interference sources has not been fully explored, this industry lacks dedicated learning samples and detection models for this specific task. In this article, an antenna dataset is created to address important antenna interference source detection issues and serves as the basis for subsequent research. We introduce YOLO-Ant, a lightweight convolutional neural network (CNN) and transformer hybrid detector specifically designed for antenna interference source detection. Specifically, we initially formulated a lightweight design for the network depth and width, ensuring that subsequent investigations were conducted within a lightweight framework. Then, we propose a DSLK-Block module based on depthwise separable convolution and large convolution kernels to enhance the network’s feature extraction ability, effectively improving small object detection. To address challenges such as complex backgrounds and large interclass differences in antenna detection, we construct DSLKVit-Block, a powerful feature extraction module that combines DSLK-Block and transformer structures. Considering both its lightweight

design and accuracy, our method not only achieves optimal performance on the antenna dataset but also yields competitive results on public datasets.

Index Terms—Antenna interference source detection, convolutional neural network (CNN)–transformer fusion, lightweight, small object detection, YOLO-Ant.

I. INTRODUCTION

TO ENSURE high-quality communication in people’s work and daily lives, various wireless devices operate in different frequency bands. 5G communication is of particular note due to its introduction of new frequency bands into everyday communication. However, due to the presence of numerous private wireless signals that have not undergone spectrum allocation by communication regulatory authorities, the 5G communication network has accumulated a considerable number of sources of interference. If individuals operate in the same geographical areas and occupy similar or adjacent frequency bands as these interference signals in their everyday communication, this will result in a significant deterioration in communication quality, as shown in Fig. 1. Regular remediation of radio interference sources is vital for communication departments to alleviate this situation. The identification of signal interference sources necessitates monitoring personnel to visually inspect areas where communication quality is compromised due to the presence of suspicious antennas elevated at high altitudes, constituting a time-consuming and labor-intensive task. In light of the mature advancements in unmanned aerial vehicle (UAV) cruising technology and object detection techniques within computer vision, unmanned drones have become viable alternatives for handling complex and challenging detection tasks previously performed by humans. For example, Lu et al. [1], Yang et al. [2], and Hao et al. [3] noted that object detection tasks in deep learning combined with UAVs have been useful in production and other areas. The success of these approaches has demonstrated the feasibility of utilizing UAVs for object detection tasks aimed at interference source antennas. However, due to the nascent stage of this detection task within the current domain of object detection, the creation of a suitable antenna dataset and the exploration of appropriate object detection methodologies are of paramount importance.

Convolutional architectures are the basis for most object detection frameworks in industrial scenarios and rely on

Manuscript received 26 May 2023; revised 1 January 2024; accepted 28 January 2024. Date of publication 19 March 2024; date of current version 5 April 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62001173 and Grant 42074038, in part by the Project of Special Funds for the Cultivation of Guangdong College Students’ Scientific and Technological Innovation (“Climbing Program” Special Funds) under Grant pdjh2022a0131 and Grant pdjh2023b0141, and in part by the Fundamental Research Funds for the Central Universities and Xiaomi Young Talents Program. The Associate Editor coordinating the review process was Dr. Shovan Barma. (*Corresponding author: Xiaoyu Tang.*)

Xiaoyu Tang, Xingming Chen, and Jintao Cheng are with the School of Electronic and Information Engineering, Faculty of Engineering, South China Normal University, Foshan, Guangdong 528225, China, and also with the School of Physics, South China Normal University, Guangzhou, Guangdong 510000, China (e-mail: tangxy@sclu.edu.cn).

Jin Wu is with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong.

Rui Fan is with the College of Electronics and Information Engineering, Shanghai Research Institute for Intelligent Autonomous Systems, the State Key Laboratory of Intelligent Autonomous Systems, and the Frontiers Science Center for Intelligent Autonomous Systems, Tongji University, Shanghai 201804, China (e-mail: rui.fan@ieee.org).

Chengxi Zhang is with the School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China.

Zebo Zhou is with the School of Aeronautics and Astronautics, University of Electronic Science and Technology of China, Chengdu 611731, China, also with the Aircraft Swarm Intelligent Sensing and Cooperative Control Key Laboratory of Sichuan Province, Chengdu 610097, China, and also with the National Laboratory on Adaptive Optics, Chengdu 610209, China.

Digital Object Identifier 10.1109/TIM.2024.3379397

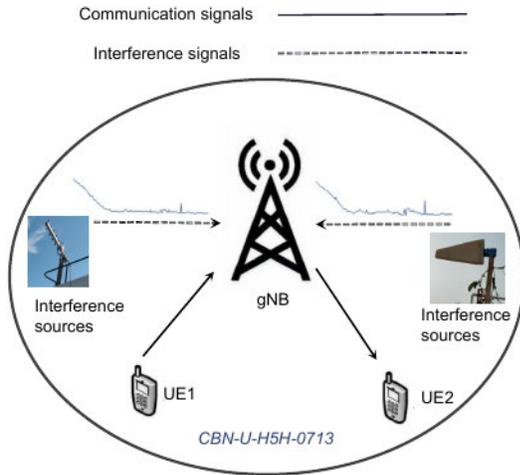


Fig. 1. Process of 5G communication in the CBN-U-H5H-0713 area is shown in the figure. Two antenna interference source signals appear in it. The gNodeB (gNB) denotes a 5G base station. The user equipment (UE) denotes the terminal equipment that users use to access the wireless network.

the development of efficient convolutional neural networks (CNNs) in deep learning. When addressing various tasks and technical challenges, corresponding enhancements to these architectures are necessary. The antenna interference source object detection task presents three main challenges. The first issue pertains to the lightweight nature and low computational complexity of a detection model; consequently, object detectors can be deployed on lightweight computing devices, enabling real-time object detection via UAVs. Previous research, exemplified by GhostNet [4] and EfficientNet [5], has focused on designing lightweight networks as potential backbones for different detection models to achieve an overall lightweight solution. However, these networks are susceptible to information feature loss. The second difficulty in the antenna interference source object detection task lies in the differences arising from the different angles and heights from which the UAV captures the antennas. These variations result in a nonuniform distribution of target sizes within the images, most of which are extremely small in size. In addition, there is a significant interclass dissimilarity issue, wherein antennas of the same type exhibit markedly different morphologies in different images. To address these issues, researchers have explored two aspects: multiscale feature learning and attention mechanisms [6], [7], [8], [9]. However, despite improvements in small object detection accuracy, these methods encounter challenges related to the model's weak generalizability and robustness. As a result, the overall detection accuracy for the targets was compromised. Moreover, the computational complexity of such models is higher. The third difficulty is complex target backgrounds, which cause serious false and missed detections. Given that antennas are commonly installed on tall buildings or fenced balconies in practical scenarios, the resulting complex and mutually obscuring environment between the target and the background significantly hinders detection. Researchers have suggested using attention or self-attention mechanisms to address this difficulty. In [10], [11], and [12], a squeeze-and-excitation (SE) attention module

was proposed or a self-attention structure was used to build the whole network for object detection. The advantage of these models lies in their ability to effectively capture the spatial relationship between the target and the background. This capability significantly enhances the object detection performance on complex backgrounds. However, these mechanisms tend to consume considerable computational resources and memory, which is not consistent with the original lightweight design intention. In addition, networks built solely on self-attention mechanisms also suffer from long training times and poor detection accuracy for small objects.

In response to the aforementioned limitations, we propose YOLO-Ant, a lightweight one-stage detector designed for detecting antenna interference sources with small targets and complex backgrounds. Initially, we analyze the scale and number of channels in each feature layer of the model; subsequently, we design the network's width and depth to ensure that the entire detection process is performed within a lightweight framework. Our design considerations aim to balance detection accuracy with the reduction of model parameters and computational complexity. To address the issues of small target size and large interclass variation, we implement an efficient feature extraction module based on depthwise separable convolution, DSLK-Block, which is applied to each feature layer in the model. This method effectively enhances the network's feature learning and fusion capabilities, leading to a significant improvement in detection accuracy for all types of targets, particularly small targets. In addition, this approach contributes to reducing the model's overall weight. Finally, to address the problem of complex backgrounds, YOLO-Ant uses an innovative CNN and transformer hybrid structure to act on the neck of the model. This process enables us to fully utilize both local and global feature learning to address the challenges posed by complex backgrounds while still accounting for small object detection. This approach significantly improves all the detection accuracy indicators while only slightly increasing the model's number of parameters and computational complexity. To demonstrate the model's generalizability and robustness, we also tested YOLO-Ant on public datasets and achieved highly competitive results. In conclusion, the main contributions of this article are as follows.

- 1) In response to the lack of learning samples for antenna object detection schemes, we conducted image acquisition and manual annotation of the three most common types of antennas encountered in real-world interference source investigation tasks. This dataset is pioneering and establishes the foundation for subsequent work.
- 2) We initially pruned YOLOv5-s, obtaining a lightweight detection framework. Within this framework, the following conditions hold.
 - a) A lightweight plug-and-play module based on depthwise separable convolution combined with large convolutional kernels was proposed to effectively improve the feature extraction and detection capabilities of the network for small targets.
 - b) The innovative use of a transformer module to construct the neck structure of the detection model

improved the detection capability of the network without increasing the model's parameter count or computational complexity, effectively solving the problem of dealing with complex backgrounds.

- 3) Our proposed method achieves state-of-the-art (SOTA) performance on the antenna dataset, striking a balance between lightweight design and detection accuracy. Moreover, YOLO-Ant yields competitive results on public datasets such as COCO, validating its robustness and superior performance. The source code is released in <https://github.com/SCNU-RISLAB/YOLO-Ant>.

The remainder of this article is structured as follows. Section II presents the related work, briefly introducing the improvement points of the model proposed in this article and the related work involved, including CNN network development, the emergence of the transformer detection model, and the crossover development between the CNN and the transformer. Section III discusses the proposed lightweight detection framework based on the YOLOv5-s improvement, including pruning the baseline model, the design of the DSLK-Block module, and the neck structure built based on DSLKvit-Block. The experimental results are given in Section IV, and Section V concludes this article.

II. RELATED WORK

A. CNN-Based Object Detection

The development of object detection in the computer vision field has been greatly influenced by CNN-based methods. Traditional approaches using hand-designed features and classifiers have been shown to be inadequate, leading to the dominance of CNN-based methods. The initial CNN model, LeNet-5 [13], was limited by computational resources and model size. However, with advancements in computational power and larger datasets, deeper and more complex CNN models, such as AlexNet [14], VGGNet [15], GoogLeNet [16], and ResNet [17], have emerged. These models have improved network accuracy, reduced parameters, and addressed network degradation issues, laying a solid foundation for 2-D object detection.

Two distinct methods have emerged from the convoluted development of 2-D object detection: two- and one-stage detectors. Two-stage detectors, such as R-CNN [18] and Fast R-CNN [19], generate candidate frames using algorithms and perform classification and regression on each candidate frame. Faster R-CNN [20] introduces the region proposal network (RPN) for candidate frame generation. In contrast, one-stage detectors, such as YOLO [21] and SSD [22], perform classification and regression directly on each location in the input image. YOLOv2 [23] and YOLOv3 [24] improved detection accuracy through methods such as multiscale prediction, batch normalization, and feature pyramid networks (FPNs). SSD introduces multiscale detection using multiple-scale feature maps, while RetinaNet [25] focuses on addressing the category imbalance problem. For the aforementioned model, one-stage detectors are more suitable for real-time detection tasks on UAVs than two-stage detectors because they do not require additional networks or algorithms for fine-tuning. However,

to compensate for the deficiency in accuracy resulting from the pursuit of detection speed, improvements need to be made to the backbone and neck of the one-stage detector by developing various efficient feature extraction modules or structures. The backbone and neck are the basic components of object detection models. The backbone is a CNN trained on image classification datasets such as ImageNet [26], in which the input image is transformed into a high-dimensional feature representation. The neck module further processes the feature map, changing the scale and resolution to extract different levels of feature information. Numerous object detection models, such as NAS-FPN [27], EfficientDet [28], YOLOv4 [29], and YOLOv7 [30], have been developed based on these concepts, incorporating various improvements and techniques to enhance accuracy and performance. However, these general models are often designed with modules that consider various common tasks, exhibiting generalizability but not effectively addressing specific challenges in particular scenarios. For instance, there are several challenges, such as small object detection and complex backgrounds, in our task. Therefore, making task-specific modifications is crucial when contemplating different tasks.

B. Developing an Attention Mechanism in the CV Domain

Attention mechanisms, initially utilized in natural language processing, have gained significant traction in computer vision [31], [32], particularly in the field of object detection. Attention mechanisms, such as channel attention, spatial attention, and their combinations, have been introduced [33], [34], [35]. They effectively utilize global and local information in feature maps, improving feature representation and attention weighting, thereby enhancing model accuracy and efficiency. However, for these conventional attention mechanisms, a fixed window size or other constraints are typically employed to regulate the correlation between each position and others. In contrast, self-attention mechanisms can extract information from different positions in the information sequence more flexibly, enabling the extraction of global information. This flexibility has contributed to the widespread application of transformer [36] models based on self-attention mechanisms, including in various domains such as computer vision. For example, the Vision Transformer (ViT) [37] splits images into patches for self-attention computations. The swin transformer [38] improves local information processing by using a window-based partitioning approach. Detection with transformers (DETR) [11] adopts a global self-attention mechanism, allowing each position to obtain contextual information from the entire image. Naturally, transformers incur substantial computational costs and training time, posing challenges for model convergence. To address these challenges, researchers have introduced lightweight transformer object detectors, including MobileViT [39] and EdgeViT [40]. Moreover, innovative approaches, such as conditional DETR [41] and DN-DETR [42], have been developed to address the crucial issue of slow training convergence. However, due to their simplified design, the majority of current lightweight transformer structures are applicable only to classification tasks involving small-sized image inputs and are not suitable

for detection tasks. The proposed detection methods aimed at addressing slow convergence have made transformer models more complex. Therefore, achieving a balance between the lightweight nature of transformer models and detection accuracy remains a crucial research scope in the current field of computer vision.

C. Combination of CNN and Transformer

In object detection, CNNs and transformers have distinct applications and advantages. CNNs are known for their strong image feature extraction abilities, ability to perform multi-channel processing, and ability to learn spatial correlations. However, CNN-based models have limitations in handling objects of different sizes and proportions due to fixed window sizes and strides. On the other hand, transformers exhibit excellent performance in capturing long-range dependencies within input sequences without prior knowledge, albeit at a slower speed and requiring substantial amounts of training data. Evidently, the amalgamation of CNNs and transformers offers complementarity across various dimensions, and researchers have already delved into numerous methodologies to explore this synergy.

The pioneering DETR model replaces fully connected and convolutional layers with transformers while using ResNet as the feature extractor, improving accuracy and efficiency. Huawei's CMTBlock combines depthwise separable convolution and the transformer's multihead self-attention module for local and global information fusion. The CMT model [43] stacks the CMTBlock in a hybrid CNN-transformer structure. The Conformer [44] adopts a dual-network structure, where the CNN branch enhances the local perception of the transformer branch. The mobile-former [45] features parallel CNN and transformer modules with bidirectional bridges, leveraging MobileNet [46] for local processing and the transformer for global interaction. However, networks or models employing such hybrid structures face challenges in effectively balancing accuracy and lightweight design. For instance, detectors, such as DETR and lacking FPN structures, exhibit suboptimal performance in small object detection. While the CMT and Conformer networks have proven effective in classification tasks, their application to downstream tasks such as object detection deviates from the realm of lightweight design. In contrast to the aforementioned models, which concatenate both structures, an alternative approach involves making transformer-style improvements directly on the CNN network. ConvNeXt [47] implements novel architectures and optimization strategies similar to those of transformers, achieving competitive results without attention structures. RepLKNet [48] employs large convolutional kernels to widen the receptive field, thus emulating the transformer-like capability for global feature extraction. By investigating the computational principles of transformers, ACMix [49] maps their operation process onto convolutional operators, thereby combining them with traditional convolution operations to construct a novel CNN architecture. Parc-Net [50] introduces circular convolution for global information extraction within a pure convolutional structure. Although these innovative networks

may not achieve the SOTA performance, their greater significance lies in exploring the factors contributing to the success of transformers from a CNN perspective, providing inspiration for subsequent research endeavors. The fusion of transformers and CNNs offers a flexible and diverse range of integration methods. Future research should strive to deepen the understanding of their interactions to improve design and optimization.

D. Object Detection of Antenna Interference Sources

Regularly monitoring and mitigating antenna interference sources has become one of the most critical tasks in the wireless communication field. In the past, detecting antenna interference sources mainly relied on traditional techniques such as spectrum analysis, signal recognition, and positioning. However, these methods have many limitations. For example, when detection personnel identify a radio interference signal through a spectrum analyzer, they can determine only the approximate direction of the interference source based on the strength of the received signal and cannot accurately determine its position.

The rapid advancement of deep learning and computer vision has facilitated the successful application of object detection-assisted tasks in various industries. Examples include defect detection in industrial settings, pest/weed detection in agriculture, and vehicle and pedestrian detection in transportation [51], [52], [53], [54], [55]. These solutions provide effective ideas for our antenna interference source detection task. When investigators confirm the approximate direction of the interference source antenna through a signal receiver and spectrum analyzer, they can use drones with cameras and related object detection algorithms to replace manual accurate positioning work. Unfortunately, the field of antenna interference source detection based on object detection tasks has largely not been explored. Due to the lack of learning samples and models for related antenna interference source detection, existing detection methods are not suitable for antenna detection. Therefore, it is urgent and meaningful to create a professional dataset and train a model suitable for this detection task to address the difficulty of locating antenna interference sources in the wireless communication field.

III. PROPOSED DETECTION FRAMEWORK

A. Overall Model Structure

The overall idea for the network (Fig. 2) lies in the combination of a CNN and transformer, both the inductive bias ability of the convolutional operation and the ability of the transformer to extract global information, while also meeting the needs of a lightweight model with low computational complexity. YOLO-Ant adopts DSLKNet, which is composed of DSLK-Blocks, as the backbone for downsampling and feature extraction in images. In DSLKNet, four DSLK-Layers employ convolutional kernels of varying sizes to sequentially extract rich features from different receptive fields of the image. To address the challenge of detecting small objects, we incorporate the neck structures of the FPN and PAN for multiscale feature learning. On the neck component,

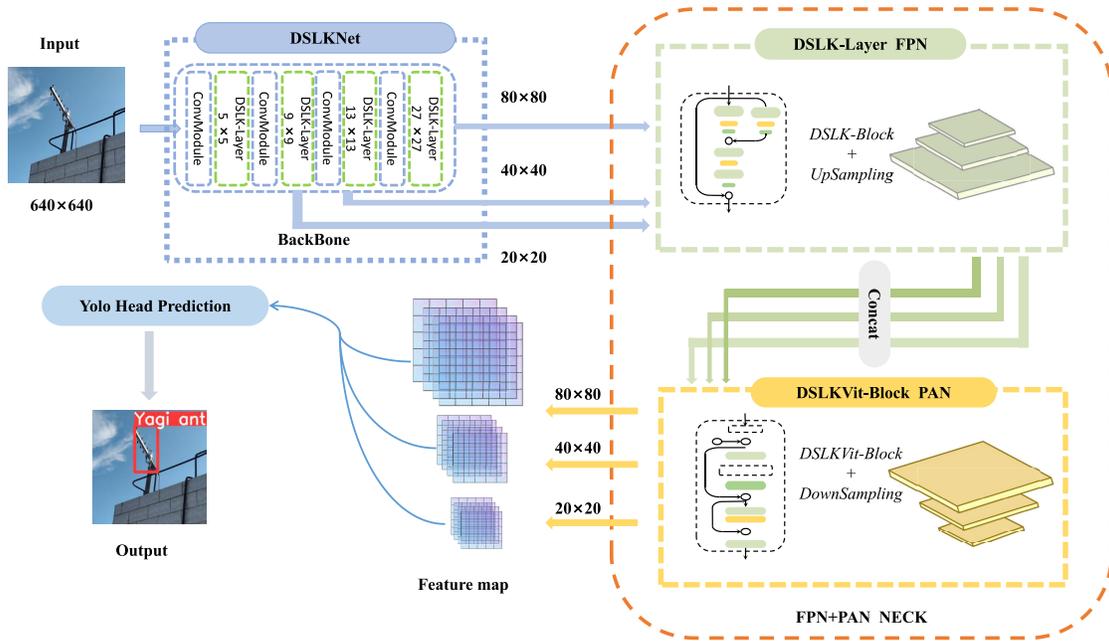


Fig. 2. YOLO-Ant model can be roughly divided into three parts: the backbone consisting of DSLKNet, the FPN + PAN structure consisting of DSLK-Layer, and DSLKVit-Block forming the neck.

we conducted pruning based on YOLOv5-s (the detailed data provided in Section IV). In comparison to the baseline model, the pruned neck model features an increased number of module stacks and a reduced number of channels in each module. This structural modification effectively alleviates redundancy in feature extraction, resulting in an overall model that is not only more lightweight but also attains higher detection accuracy. In this lightweight framework, we introduce the DSLKVit-Block, which is a combination of transformer and convolutional modules. Even though the transformer module has a larger parameter count and computational complexity than the convolutional module, the final YOLO-Ant model remains lighter than the baseline model. Overall, the integration of CNN and transformer in the model is manifested as follows: 1) a transformer-like pure CNN structure is proposed using depthwise separable convolution and a large convolution kernel, effectively expanding the perceptual field of the convolution operation and extracting the target context information; and 2) the FPN structure of the pure CNN and the PAN structure designed with the transformer module is combined in a parallel structure to complement each other and thus improve the feature processing capability of the model. Sections III-B and III-C provide a detailed description of the working principles of the DSLK-Block and DSLKVit-Block within the YOLO-Ant.

B. More Efficient Feature Extraction Module, DSLK-Block

The DSLK-Block structure built with depthwise separable convolution was introduced using large convolutional kernels. The design of this structure is based on several starting points.

- 1) Depthwise separable convolution is used instead of conventional convolution operations to achieve model lightweighting.

- 2) Large convolutional kernels are used to increase the receptive fields to extract a greater amount of contextual information. Models, such as RepLKNet, have shown that pure convolutional networks can achieve performance comparable to that of transformer-style networks in this way.
- 3) Inspired by the ConvNeXt approach, DSLKBlock uses fewer normalization and activation functions, replacing the rectified linear unit (ReLU) with a Gaussian error linear unit (GELU).

However, unlike other feature extraction blocks, DSLK-Block has three places that change correspondingly with the network location of the DSLK-Layer to balance the relationships between parameter volume, computational complexity, and accuracy.

- 1) The size of the large convolutional kernel in the backbone changes according to the location of the DSLKBlock. The rationale behind this design primarily stems from several considerations. First, if all DSLK-Layers adopt excessively large convolutional kernels, the model will be greatly burdened in terms of parameters and detection speed. Second, small objects typically have pixels in the range of 32×32 , roughly equivalent to $1/40$ of the original image size. When the network input size is set to 640×640 , the corresponding size of the small objects is approximately 16×16 . To ensure that the early layers of the network can sufficiently extract feature information from small objects and prevent the loss of small object details caused by large convolutions, we employ 5×5 and 9×9 convolutional kernels in the first two layers. Finally, as the downsampling rate increases, the feature map sizes decrease. We employ larger convolutional kernels to handle larger sized objects, while these

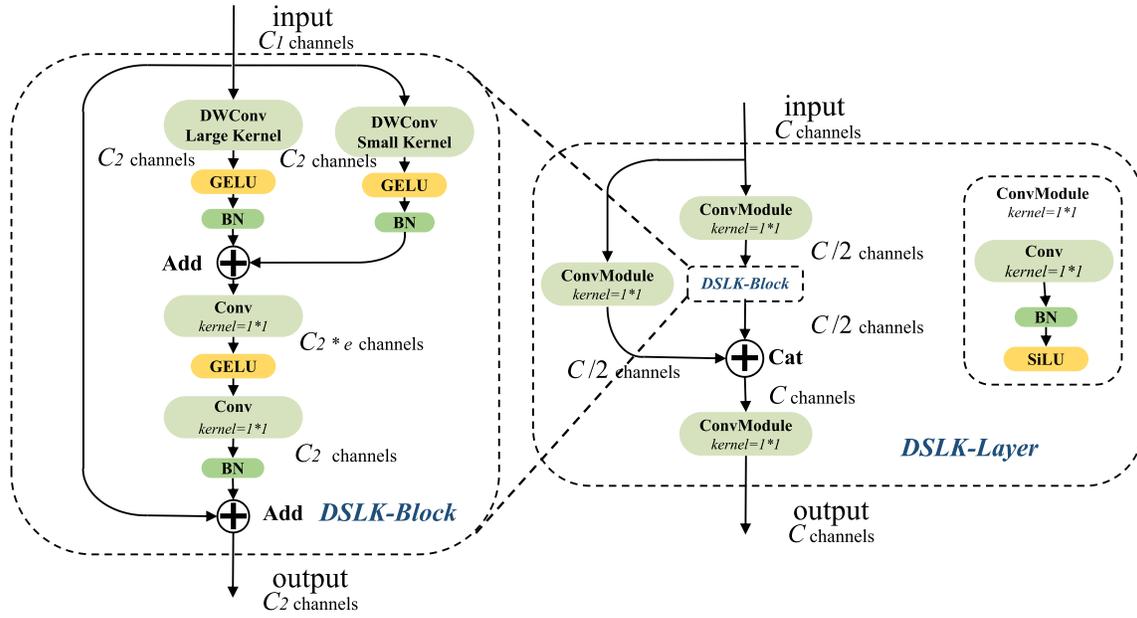


Fig. 3. DSLK-Block and DSLK-Layer.

larger kernels also provide more comprehensive contextual information for small feature maps (derived from the relationships between adequately extracted small objects and their surrounding environments). In the final model backbone, the sizes of the large convolutional kernels are set to 5×5 , 9×9 , 13×13 , and 27×27 , thereby achieving efficient detection, particularly for objects of different sizes, especially small objects.

- 2) For the DSLK-Layer within the model backbone, to prevent the potential loss of important information as the convolutional kernel size increases, we incorporated a parallel pathway using 3×3 convolutional kernels within the DSLK-Block. The output from the small kernel pathway is fused with that of the large kernel pathway using an addition operation. To ensure model lightweightness and expedite model convergence, within the neck structure, all the large convolutional kernels within the DSLK-Block were resized to 3×3 dimensions, while the small convolutional kernel pathway was modified to follow a conventional shortcut form.
- 3) Depthwise separable convolution first uses depthwise convolution to convolve each feature point within the same channel and then extracts information between different channels of the same feature point through pointwise convolution. After decomposing conventional convolution operations into these two steps, the computational cost of the convolution operation is effectively reduced. The pointwise convolution of the DSLK-Block adopts a variable factor to control the number of channels. To further balance the relationships between the model parameters, computational complexity, and accuracy, the DSLK-Block uses different pointwise convolution depths at different positions in the network and achieves good results. The final structures of DSLK-Block and DSLK-Layer are shown in Fig. 3. The

DSLK-Block is part of the DSLK-Layer, as illustrated in the right of Fig. 3, and forms each CNN feature processing layer within the backbone and neck of the model. Assuming that the input feature map is denoted as $X_i \in \mathcal{R}^{C_i \times H_i \times W_i}$ (where C , H , and W denote the number of channels, spatial height, and width, respectively), its workflow can be represented as follows:

$$F_{\text{DSLK-Layer}}(X_i) = f_{\text{CBS}}(1, C_i, f_{\text{CBS}}(1, C_i/2, X_i)) \otimes F_{\text{DSLK-Block}}(f_{\text{CBS}}(1, C_i/2, X_i)) \quad (1)$$

where \otimes represents the Concat operation and f_{CBS} represents a module that sequentially undergoes convolution, normalization, and activation functions, which can be expressed using the following formula:

$$f_{\text{CBS}}(k, c, X_i) = \rho(\sigma(f_{\text{conv}}(k, c, 1, X_i))) \quad (2)$$

where $\rho(x)$ represents the batch normalization, $\sigma(x)$ represents the SiLU activation function, $f_{\text{conv}}(k, c, g, X_i)$ represents the convolution operation, k is the kernel size, c is the output channel number, and g represents the number of groups ($g = 1$ in regular convolution and $g = C_i$ in depthwise separable convolution).

We represent the workflow of the DSLK-Block using formula $F_{\text{DSLK-Block}}$

$$F_{\text{DSLK-Block}}(X_i) = X_i + f_{\text{pw}}(e, f_{\text{dw}}(K_L, X_i) + f_{\text{dw}}(K_S, X_i)) \quad (3)$$

where K_L and K_S represent the large and small convolution kernels used in the two depthwise convolution paths, respectively (K_L takes values of 3, 5, 9, 13, and 27 in the model and K_S takes a value of 0 or 3; if $K_L = 0$, then the path becomes a normal shortcut operation); and f_{dw} represents the substitution

of conventional convolutions with depthwise convolution in f_{CBS} , as expressed by the following formula:

$$f_{dw}(K, X_i) = \rho(\sigma(f_{conv}(K, C_i, C_i, X_i))) \quad (4)$$

while f_{pw} represents the pointwise convolution block, as expressed by the following formula:

$$f_{pw}(e, X_i) = \rho(f_{conv}(1, 1, C_i, \sigma(f_{conv}(1, 1, e \times C_i, X_i)))) \quad (5)$$

where e represents the variable expansion coefficient, which is used to control the channel expansion and scaling factor in the pointwise convolution process.

C. DSLKViT-Neck Structure for Efficient Feature Fusion

The efficiency of transformer models relies heavily on their global attention mechanism, which differs from convolution operations in that information between feature points is calculated only within the size of the convolution kernel. Instead, transformer models consider the interactions between each feature point and all other points in the feature map. Using the transformer's self-attention mechanism to enhance the contextual information extraction of feature points is useful for dealing with problems such as target shape differences caused by multiangle shots from UAVs and interference from complex environmental backgrounds in antenna detection tasks. However, this design comes at the cost of consuming a significant amount of computational resources, making it unacceptable for lightweight models. Therefore, finding methods to efficiently utilize transformer structures while conserving resources remains a significant challenge.

In MobileViT, the transformer calculates information only between feature points at the same position in each patch of the feature map. In EdgeViT, a convolution operation is used to aggregate all the information within a local window of size $k \times k$. Then, the transformer calculates the information between all the feature points that contain the aggregated information. Due to the large amount of data redundancy inherent in image data, the difference in information between adjacent pixels is often not significant, allowing this computational savings to occur. Motivated by models such as MobileViT and EdgeViT, we introduce an innovative DSLKViT-Block module, which incorporates both CNN and transformer architectures, as shown in Fig. 4. To reduce the computational complexity of the entire model, DSLKViT-Block calculates information only between "representative feature point" within small regions of the feature map instead of computing the mutual information between every feature point.

The DSLKViT-Block initially conducts feature extraction on localized regions of the feature map through convolution operations within the DSLK-Block, resulting in a set of "representative feature points." Each of these feature points represents the aggregated features from their respective regions. The new feature map, composed of all these "representative feature points," has reduced dimensions, conserving resources for subsequent self-attention computations. Assuming that the input feature map is denoted as $X_i \in R^{C_i \times H_i \times W_i}$

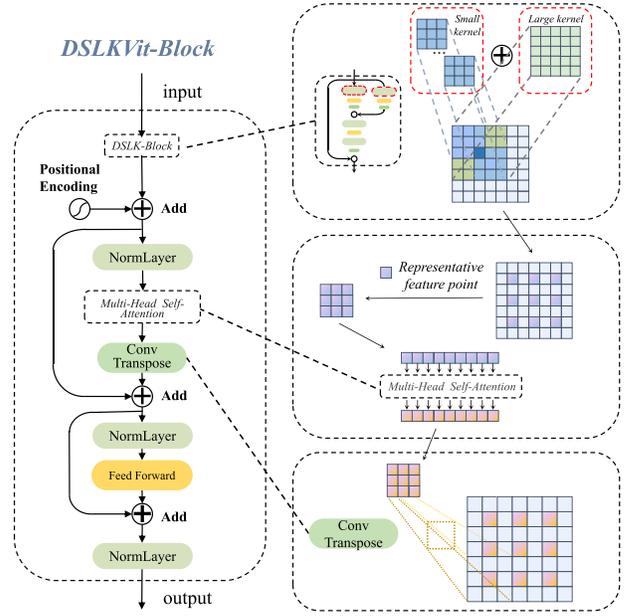


Fig. 4. DSLKViT-Block.

(where C , H , and W denote the number of channels, spatial height and width, respectively), we denote the process of aggregating local information in the DSLKViT-Block as follows using the formula f_{Local} :

$$f_{Local}(X_i) = f_{LA}(sr, \rho(f_{PE}(X_i) + F_{DSLK-Block}(X_i))) \quad (6)$$

where $\rho(x)$ represents the NormLayer operation, f_{PE} represents the positional encoding achieved through convolution operations, and $f_{LA}(sr, x)$ represents the operation of aggregating local information, which can be achieved using pooling layers. The sr parameter indicates the local range covered by the "representative feature points" and is equal to $sr \times sr$.

Subsequently, the feature map undergoes multihead attention calculations, enabling the interactions among the "representative feature points" that aggregate information within each region, thereby acquiring rich contextual information between various regions on the original feature map. Finally, the model employs deconvolution operations to map the "representative feature points" back to their respective corresponding regions and enhances the network's expressive capabilities through a feedforward network (FFN). The process of obtaining global information can be expressed using the following formula:

$$f_{Global}(sr, X_i) = X_i + \rho(f_{LD}(sr, MHSA(f_{Local}(X_i)))) \quad (7)$$

where $f_{LD}(sr, x)$ represents the operation of mapping representative feature points back to their original regions, which can be accomplished via deconvolution. MHSA represents the multihead self-attention computation

$$\begin{aligned} MHSA(X) &= \text{MultiHead}(Q, K, V) \\ &= \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h) W^O \end{aligned} \quad (8)$$

where $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$, $W_i^Q \in R^{d_{model} \times d_k}$, $W_i^K \in R^{d_{model} \times d_k}$, $W_i^V \in R^{d_{model} \times d_v}$, and $W^O \in R^{hd_v \times d_{model}}$. In this work, we employ $h = 4$ parallel attention

heads. For each of these, we use $d_k = d_v = d_{\text{model}}/4 = C$. For single-headed attention, we compute the attention function on a set of queries simultaneously, packed together into a matrix Q . The keys and values are also packed together into matrices K and V . We compute the matrix of outputs as follows:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (9)$$

where d_k is a scaling factor. We can obtain $Q \in R^{N \times C}$, $K \in R^{N \times C}$, and $V \in R^{N \times C}$ through a linear layer by input feature $X \in R^{C \times H \times W}$, where $N = H \times W$.

Ultimately, the entire workflow of DSLKVit-Block can be represented as follows:

$$F_{\text{DSLKVit}}(\text{sr}, X_i) = \text{FFN}(f_{\text{Global}}(\text{sr}, (f_{\text{Local}}(X_i)))) \quad (10)$$

$$\text{FFN}(X_i) = X_i + \rho(\text{MLP}(X_i)) \quad (11)$$

where the multilayer perceptron (MLP) function can be implemented using two fully connected layers.

After finalizing the foundational DSLKVit-Block module, we proceeded to modify the model's neck structure at the macro level, aiming to harness the complementary advantages of transformers and CNNs to their fullest extent. We observed that the neck structure of YOLOv5-s adopts an FPN with a PAN architecture. In this configuration, the FPN's top-down structure merges high-level feature maps with low-level feature maps to propagate semantic information from higher levels, enhancing the representational capacity of the low-level feature maps. Subsequently, the connection of the PAN, a bottom-up structure, aims to assist high-level feature maps in acquiring richer positional information from shallow-level feature maps. The PAN and FPN also engage in lateral connections to fuse feature maps of the same dimensions, enriching the information content. Therefore, the use of the transformer to construct PAN is logically sound. During the bottom-up process, the DSLKVit-Block gradually extracts finer features from large-sized feature maps to small-sized feature maps. Moreover, lateral connections are leveraged to incorporate the convolution results of the DSLK-Block to compensate for the insensitivity of the transformer structure to positional information. Furthermore, due to the computational complexity of the transformer being $O(N^2)$, applying it to feature maps of size 80×80 would result in a prohibitively high computational cost. Therefore, we deployed the DSLKVit-Block only in the smallest two feature layers (40×40 and 20×20).

The DSLKVit-Neck structure constructed from the above design incorporates the complementary concepts of CNNs and transformers at both the microlevel (within DSLKVit-Block) and macrolevel (the entire model's neck structure).

IV. EXPERIMENT

A. Experimental Description

1) Datasets Introduction:

1) *Antenna Interference Source Dataset*: In the antenna interference source detection task proposed in this article, the methods and datasets in the relevant field are still lacking. Therefore, our first priority is to create the first dataset specifically for antenna interference

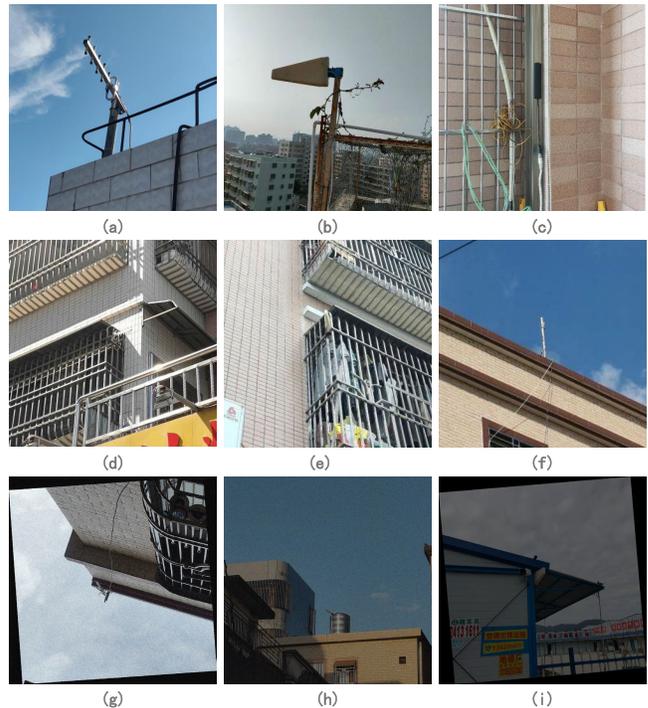


Fig. 5. Antenna interference source dataset. (a)–(c) Represent the yagi antenna, the plate log antenna, and the patch antenna, respectively. (d)–(f) Demonstrate challenges commonly encountered in practical detection such as detecting small objects and complex backgrounds. (h)–(j) Illustrate the usage of rotation, flipping, and image color space data augmentation techniques on the antenna dataset.

source detection. In the daily interference source investigation and cleaning activities of the communication bureau, we identified three common antenna interference sources that significantly impact communication: the Yagi antenna, plate log antenna, and patch antenna, as shown in Fig. 5(a)–(c). The three antennas have different shapes, bringing multiple challenges to the detection task. For example, the Yagi antenna is easily confused with a complex fence background when it is installed on a balcony, as shown in Fig. 5(d). The plate log antenna has a flat and wide shape that can be recognized from the front well. However, the angle from the side greatly reduces the recognition rate, and different angles create problems with significant interclass differences, as shown in Fig. 5(e). The patch antenna is the most difficult to recognize because it is merely a monochromatic rectangular object accompanied by a lengthy wire, resulting in a lack of intricate features available for the model to learn from. In addition, when the patch antenna is wrapped around a pole and placed high up, the object captured by aerial photography is small and difficult to distinguish in the image, as shown in Fig. 5(f). Moreover, we simulate different angles, distances, and lighting effects to enrich the dataset when drones take pictures of the antennas. Based on the collected images of the three types of antenna interference sources, we used common data augmentation techniques such as rotation, flipping, and image color space changes to expand the original images. Finally, the dataset was divided into training images (1777) and

validation images (449) at an 8:2 ratio and labeled by professionals.

- 2) *COCO*: The Common Objects in Context (COCO) dataset is widely used in the computer vision community for benchmarking object detection and segmentation algorithms. It is designed for use in object detection, segmentation, and captioning. The dataset contains more than 330 000 images with more than 2.5 million object instances labeled across 80 different object categories. In addition, the dataset includes 91 different categories of “stuff” or background categories, such as sky, grass, and water. The dataset also includes annotations for object segmentation masks, object bounding boxes, and keypoint annotations.
- 3) *VisDrone*: The VisDrone dataset is a large-scale benchmark dataset for visually understanding aerial scenes. The dataset consists of more than 6000 video clips and more than 2 million images captured by various types of drones in different locations and scenarios, covering a wide range of aspects of aerial vision. The dataset contains rich annotations for object detection, tracking, counting, and image quality assessment, which are important for both academic research and industrial applications.

2) Experimental Setting:

Experimental Configuration: All the experiments were executed on the Linux operating system, an NVIDIA GTX3060 GPU with 12 GB of memory, and data training and object detection were performed in PyTorch 1.12.0, torchvision0.13.0, and CUDA10.2 environments. On two public datasets, COCOtrain2017 and VisDrone2019-DET-train, we conducted training and testing on the COCOval2017 and VisDrone2019-DET-val validation sets, respectively. In the experiments with the COCO dataset, YOLO-Ant was trained without the use of pretrained weights, and the model was tested using the weights that achieved the highest mAP.5:.95 after training to full convergence. For the Antenna dataset and the Visdrone dataset, training was more challenging due to the smaller number of samples and the difficulty posed by small targets. To improve the model fit and enhance the accuracy, all the models were trained using pretrained weights (models pretrained on the COCO dataset). To ensure a fair comparison, all the experiments involving model training, testing, and other evaluations are conducted using images of size 640×640 as the input. All pretrained weights for comparative models in the experiments, as well as settings for hyperparameters such as learning rates and optimizers, were sourced from the official project or MMDetection’s official documentation.

3) *Evaluation Criteria*: In the object detection field, the following metrics are used to judge the detection accuracy of a model [56]: accuracy (P), recall (R), average precision (AP), mean AP (mAP), and intersection over union (IoU). The accuracy is the proportion of positive cases predicted by the model that are actually positive. Recall is the proportion of all true positive (TP) cases that the model correctly predicts as positive cases. AP is the weighted average of accuracy and recall. The mAP is the mean AP value for different categories. The IoU is the ratio of the intersection area of the predicted

frame to the true frame to the concatenated area and is usually used to evaluate the localization accuracy of an object detection model. mAP.5 indicates the mAP when the IoU threshold is 0.5; mAP.5:.95 evaluates the performance of the algorithm over a range of IoU thresholds from 0.5 to 0.95, as the detection model performance under different IoU thresholds is considered, allowing a more comprehensive assessment of the detection and localization capabilities of the model

$$P = \frac{T_P}{T_P + F_P} \quad (12)$$

$$R = \frac{T_P}{T_P + F_N} \quad (13)$$

$$AP = \int_0^1 P(R) dR \quad (14)$$

$$mAP = \frac{\sum_{i=1}^N AP(i)}{N} \quad (15)$$

where TP is the correct detection result; false positive (FP) is the wrong detection result; false negative (FN) is the wrong undetected result, which means missed detection; and N indicates the number of detection task categories.

In addition to model accuracy, the numbers of parameters and giga floating-point operations per second (GFLOPs) are crucial considerations for determining the lightweight nature of a model. FLOPs indicate the number of floating-point operations performed by a model in a single forward propagation, serving as a measure of its computational complexity. This metric enables the comparison of computational overhead across different models. GFLOPs represents the billion floating-point operations per second, with 1 GFLOPs equaling 1000 MFLOPs. In general, the more complex the model structure is, the higher the number of FLOPs. When designing an object detection model, a tradeoff between accuracy and computational complexity is needed to achieve better detection performance and higher computational efficiency by making a corresponding choice according to the specific task requirements.

B. Experimental Results and Analysis

1) *Baseline Model Selection*: Starting from the practical task of antenna interference source detection, we select classic or cutting-edge algorithms (with a focus on lightweight methods) for comparison on our antenna dataset. The selection process encompasses considerations from three main aspects: model parameter count, computational complexity, and detection accuracy. It is evident from Table I that with the increased popularity of transformers in the computer vision domain in recent years, most detection methods have become closely associated with them. However, this trend has led to increasingly larger models, rendering them unsuitable for lightweight real-world applications. In terms of detection accuracy, several cutting-edge transformer models, such as PVT-tiny [60] and DAB-DETR, not only fail to meet the lightweight criteria but also exhibit detection performances inferior to those of traditional single-stage detectors, such as FCOS [57] and YOLO. The reason behind this phenomenon lies in the inherent limitations of transformer mechanisms for small object detection.

TABLE I
PERFORMANCE OF DIFFERENT MODELS ON THE ANTENNA DATASET

model	Year	Param.	GFLOPs	mAP.5	mAP.5:.95	Yagi Antenna mAP.5	Plate Log Antenna mAP.5	Patch Antenna mAP.5
RetinaNet(ResNet50) [25]	ICCV'2017	37.97M	191.43	0.477	0.237	-	-	-
YOLOv3-tiny [24]	'2018	8.85M	13.17	0.32	0.156	0.288	0.645	0.0274
FCOS(ResNet50) [57]	ICCV'2019	32.12M	60.59	0.533	0.264	-	-	-
YOLOv4-tiny [29]	'2020	5.88M	16.18	0.28	0.134	-	-	-
YOLOx-s [58]	CVPR'2021	8.94M	26.76	0.579	0.350	0.650	0.781	0.306
YOLOv7-tiny [30]	CVPR'2023	6.23M	13.86	0.42	0.217	0.476	0.642	0.141
EfficientDet-D1 [28]	CVPR'2020	6.56M	11.51	0.267	0.156	-	-	-
TOOD(ResNet50) [59]	ICCV'2021	32.02M	60.79	0.499	0.265	-	-	-
PVT-tiny [60]	ICCV'2021	23.0M	50.94	0.472	0.246	-	-	-
DAB-DETR(ResNet50) [61]	ICLR'2022	43.70M	33.90	0.531	0.227	-	-	-
YOLOv8-s [62]	'2023	11.2M	28.4	0.548	0.316	-	-	-
DINO(ResNet50) [63]	ICLR'2023	47.54M	93.77	0.595	0.331	-	-	-
Mask R-CNN(ConvNeXt-V2-B [64])	'2023	110.59M	198.56	0.642	0.348	-	-	-
RT-DETR(ResNet18) [65]	'2023	20M	60	0.653	0.362	-	-	-
ours	'2023	6.13M	16.18	0.692	0.374	0.739	0.825	0.512

Taking a holistic view in Table I, among the group of the lightest models, there is a significant disparity in model accuracy, with most mAP.5 values falling below 0.4. In contrast, among some larger models, the accuracy on the antenna dataset is higher and more consistent, with mAP.5 values generally hovering at approximately 0.5. We observed that lightweight models, such as YOLOv4-tiny and EfficientDet-D1, are constrained by their simplistic network structures and computational limitations, making it challenging for them to achieve outstanding performance on antenna datasets characterized by small targets and complex backgrounds. On the other hand, larger models have better feature processing capabilities, resulting in overall superior accuracy. However, to meet the demand for deploying models on low-level computing platforms, lightweight models remain a crucial focus. From this perspective, even though RT-DETR [65] achieves the highest accuracy among numerous comparative models, its computational complexity remains significantly higher than that of YOLO models when using a ResNet18 backbone. In addition, due to its complex transformer architecture, the model deviates from the lightweight direction. While YOLOv7-tiny boasts lower computational complexity and model parameter count while achieving an mAP.5 above 0.4, its detection accuracy still lags behind that of YOLOv5-s. Therefore, YOLOv5-s is appropriately selected as the baseline model for subsequent experiments.

2) *Baseline Model Pruning*: In the neck structure of the YOLOv5-s model, the C3 module and convolutional layers exhibit a wide and shallow shape (deeper channel counts and fewer convolution module stackings) as the downsampling factor increases. Specifically, when the 640×640 input image is downsampled to 80×80 (P3), 40×40 (P4), and 20×20 (P5) in the neck structure, the number of stacked C3 modules is 1, and the channel numbers of the C3 modules and the convolutional layers in the three feature maps are 128, 256, and 512. When training YOLOv5-s on the antenna dataset, the visualization of the outputs of the P3, P4, and P5 feature layers revealed that increasing the number of channels in the neck section did not necessarily imply the extraction of richer features. As shown in Fig. 6(b), among

the 256 channels in the P4 feature map, there were varying numbers of highly redundant patterns, and this redundancy was even more prevalent in the 512 channels of the P5 feature map. These redundant feature maps impeded the model's detection speed and, to some extent, reduced the model's detection accuracy, particularly when these highly repetitive features were ineffective. Therefore, we conducted experiments by reducing the output channels of the C3 modules and convolutional layers corresponding to the P4 and P5 feature layers by half. The results demonstrated that this pruning operation not only made the model lighter but also improved the detection accuracy, as shown in Fig. 6(c). We reasonably conclude that widening the model by increasing only the number of channels is not effective for feature extraction and fusion on the antenna dataset; it increases the number of model parameters, reduces the computational speed, and even brings negative gains.

Based on this discovery, we reduced the channel dimensions of the P3, P4, and P5 to 128. While making the model "narrower" in terms of its width, we enhanced its learning capability and robustness by increasing the number of stacked C3 layers. After pruning the model's neck structure into a "narrow and deep" configuration, the overall model not only became more lightweight but also achieved even higher detection accuracy, as shown in Table II.

3) *Ablation Experiment*: As shown in Table III and Fig. 7(a), the baseline model was tested via ablation experiments on the COCO dataset. After pruning YOLOv5-s, both the number of parameters and the computational complexity were effectively reduced. Simultaneously, all the detection indices are comparable to or even surpass the original version, among which the mAP.5:.95 improved significantly, which shows that the structural pruning operations discussed in Section III are reasonable. On this basis, the model adds the feature processing module DSLK-Block to replace the C3 structure of the original model, which further reduces the number of parameters by 26% compared to the original YOLOv5-s. The computational complexity is also reduced by 1.52 GFLOPs. In terms of accuracy, this version of the model shows a more significant improvement in both mAP.5 and mAP.5:.95, with 11.3% and 4.26% improvements in small- and

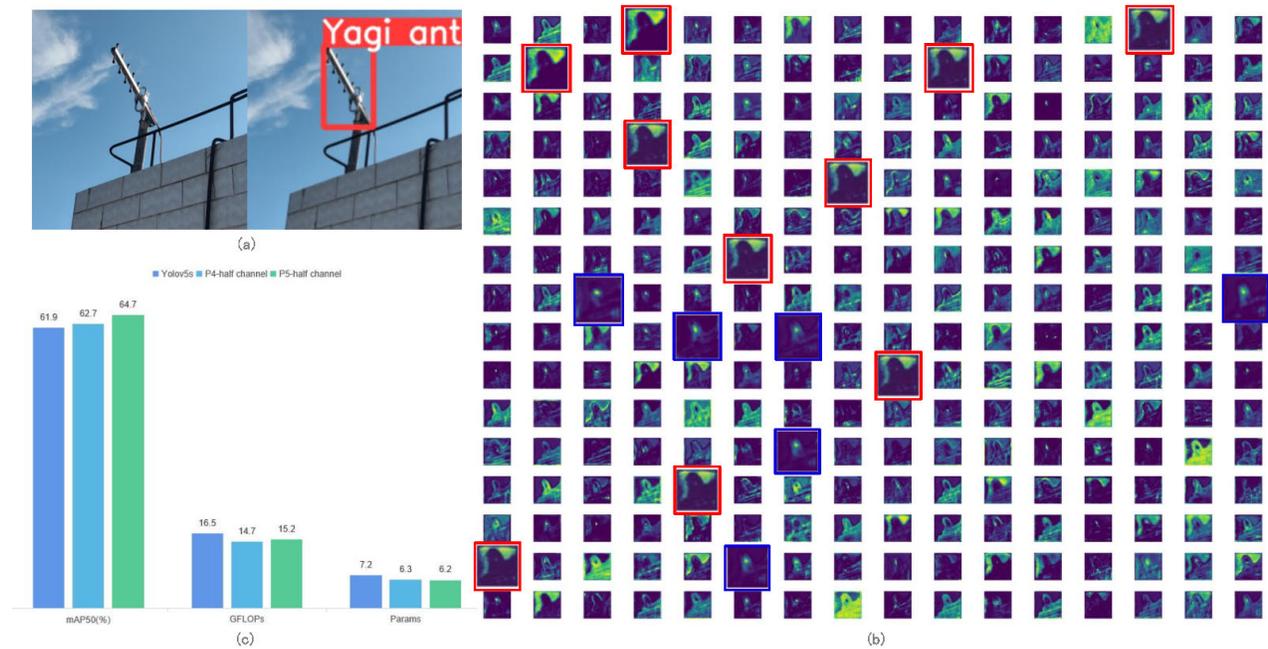


Fig. 6. (a) Example of an input antenna image and detection result. (b) Visualization results of the P4 feature layer of the baseline model were presented. These visualizations unveiled the presence of a considerable degree of information redundancy among the 256 channels. This redundancy was notably characterized by a pronounced repetition of feature maps. To illustrate this phenomenon, we have highlighted two specific groups as exemplary instances. (c) Performance results of the baseline model after channel pruning on the neck.

TABLE II
BASELINE MODEL PRUNING

Layer Name	Output size	YOLOv5-s		YOLOv5s -lite	
		modules[in channels,out channels] \times n	params	modules[in channels,out channels] \times n	params
stage 17	80 \times 80	C3[256,128] \times 1	90880	C3[256,128] \times 3	173312
stage 18	80 \times 80	Conv[128,128] \times 1	147712	Conv[128,128] \times 1	147712
stage 20	40 \times 40	C3[256,256] \times 1	296448	C3[256,128] \times 3	173312
stage 21	40 \times 40	Conv[256,256] \times 1	590336	Conv[256,128] \times 1	147712
stage 23	20 \times 20	C3[512,512] \times 1	1182720	C3[256,128] \times 3	173312
GFLOPs		16.5		15.7	
Params(all)		7235389		5398845	
mAP.5		0.619		0.640	

TABLE III
ABLATION EXPERIMENT ON THE COCO DATASET

model	Param.	GFLOPs	mAP.5	mAP.5:.95	mAP.5(small)	mAP.5(medium)	mAP.5(large)
YOLOv5s-pruning	5.39M	15.67	0.570	0.385	0.222	0.430	0.498
ours(DSLK-Block)	5.35M	14.97	0.584	0.395	0.236	0.441	0.508
ours(DSLK-Block+DSLKVit-Block)	6.13M	16.18	0.599	0.410	0.245	0.455	0.535

medium-sized targets, respectively; these findings show that DSLK-Block plays a significant role in small object detection. Furthermore, when we constructed the PAN structure in the neck using the transformer module based on DSLK-Block, the final model was obtained. Due to the balanced relationship between computing resources and accuracy improvement, DSLKVit-Block is applied to only two feature layers of smaller sizes, P4 and P5, corresponding to medium- and large-sized objects, respectively, in the detection task. The final model has an increase in the number of parameters and the

complexity of operations compared to the version using only the DSLK-Block, but both are lower than YOLOv5-s. The experimental results also show that, compared to YOLOv5-s, the detection accuracy of large targets is improved by 9.2%, which leads to a 9.6% improvement in mAP.5:.95; this is a more significant effect, while other indicators are also improved by approximately 0.03.

Table IV shows the ablation experiments of the proposed model on the antenna interference source dataset. The overall performance improvement is similar to that of the abla-

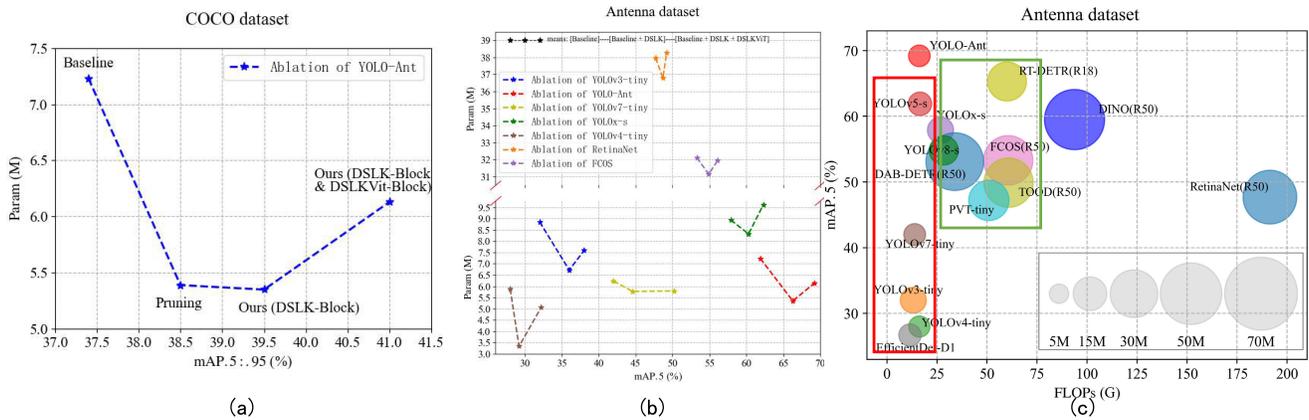


Fig. 7. (a) Corresponding changes in accuracy and number of parameters during the ablation experiment of YOLO-Ant on the COCO dataset. (b) Performance of two modules, DSLK-Block and DSLKVit-Block, retrofitting different models on the antenna dataset. (c) Performance of different models on the antenna dataset in three dimensions: accuracy, computational complexity, and number of parameters. The models enclosed by the green box generally exhibit higher accuracy compared to those enclosed by the red box but deviate from the lightweight design principle.

TABLE IV
ABLATION EXPERIMENT ON THE ANTENNA DATASET

model	Param.	GFLOPs	mAP.5	mAP.5:95	Yagi Antenna mAP.5	Plate Log Antenna mAP.5	Patch Antenna mAP.5
YOLOv5s-pruning	5.39M	15.67	0.640	0.356	0.666	0.804	0.451
ours(DSLK-Block)	5.35M	14.97	0.663	0.363	0.727	0.793	0.468
ours(DSLK-Block+DSLKVit-Block)	6.13M	16.18	0.692	0.374	0.739	0.825	0.512

TABLE V
VALIDITY OF THE PROPOSED MODULE ON OTHER MODELS (ON THE ANTENNA DATASET)

model	Year	Param.	GFLOPs	mAP.5	mAP.5:95	Yagi Antenna mAP.5	Plate Log Antenna mAP.5	Patch Antenna mAP.5
RetinaNet(ResNet50) [25]	ICCV'2017	37.97M	191.43	0.477	0.237	-	-	-
+DSLK-Block		36.80M	184.92	0.487	0.241	-	-	-
+DSLK-Block+DSLKVit-Block		38.26M	187.85	0.492	0.243	-	-	-
YOLOv3-tiny [24]	'2018	8.85M	13.17	0.32	0.156	0.288	0.645	0.0274
+DSLK-Block		6.72M	8.94	0.36	0.145	0.414	0.596	0.0682
+DSLK-Block+DSLKVit-Block		7.58M	11.45	0.38	0.17	0.37	0.609	0.16
FCOS(ResNet50) [57]	ICCV'2019	32.12M	60.59	0.533	0.264	-	-	-
+DSLK-Block		31.15M	58.49	0.549	0.268	-	-	-
+DSLK-Block+DSLKVit-Block		31.94M	58.49	0.561	0.268	-	-	-
YOLOv4-tiny [29]	'2020	5.88M	16.18	0.28	0.134	-	-	-
+DSLK-Block		3.34M	14.53	0.292	0.145	-	-	-
+DSLK-Block+DSLKVit-Block		5.06M	15.76	0.322	0.156	-	-	-
YOLOx-s [58]	CVPR'2021	8.94M	26.76	0.579	0.350	0.650	0.781	0.306
+DSLK-Block		8.32M	26.62	0.603	0.334	0.679	0.789	0.341
+DSLK-Block+DSLKVit-Block		9.61M	27.81	0.623	0.350	0.673	0.763	0.434
YOLOv7-tiny [30]	CVPR'2023	6.23M	13.86	0.42	0.217	0.476	0.642	0.141
+DSLK-Block		5.77M	12.93	0.446	0.227	0.512	0.696	0.13
+DSLK-Block+DSLKVit-Block		5.79M	12.89	0.502	0.255	0.594	0.706	0.206

tion experiments on the COCO dataset. The pruned version achieves comparable levels of all the metrics compared to the original version; both the version with only DSLK-Block and the final model show significant improvements compared to YOLOv5-s, and the final model achieves the best performance in all the metrics. Notably, the version using only the DSLK-Block has the most significant improvement in Yagi antenna detection compared to the other two antennas, while the final version using the DSLKVit-Block has a more significant improvement in the plate log antenna and patch antenna. We believe that the efficient feature extraction capability of DSLK-Block plays a crucial role due to the more

complex shape of the Yagi antenna. The plate log antenna and the patch antenna are more fixed in shape and color, and simple local convolution cannot extract more effective information; therefore, a transformer, a self-attention mechanism that focuses more on global information, can effectively extract the feature information of the target and background and combine the target with the surrounding environment information, making the improvement of detection accuracy more obvious.

In general, combining the ablation experiments performed by the model on the COCO dataset and the antenna dataset, we can conclude the following.

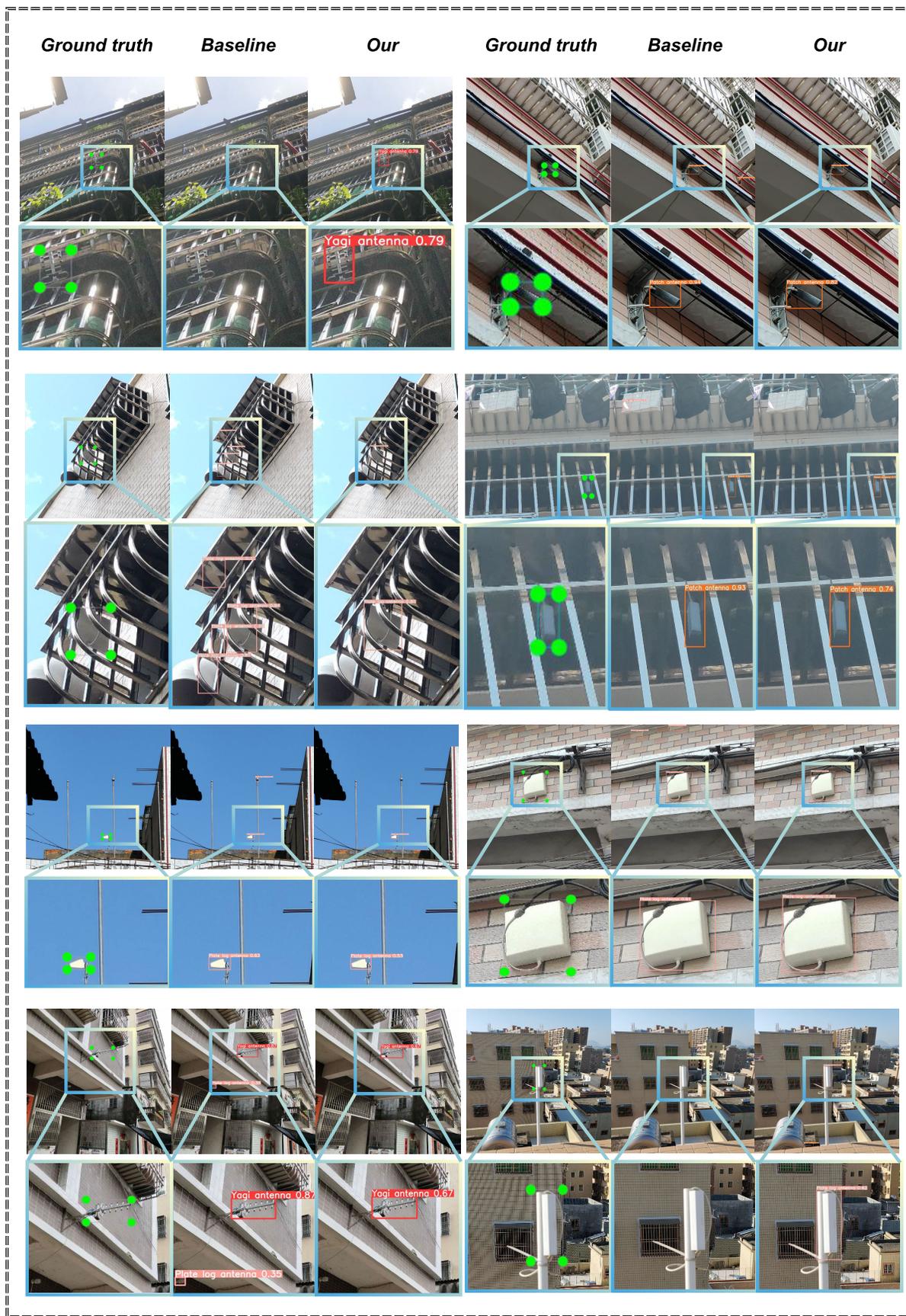


Fig. 8. Comparison of detection performance between YOLO-Ant and baseline models on the antenna interference source dataset.

TABLE VI
COMPARATIVE EXPERIMENTS ON THE COCO DATASET

model	Param.	GFLOPs	mAP.5	mAP.5:.95	mAP.5(small)	mAP.5(medium)	mAP.5(large)
EfficientDet-D1 [28]	6.6M	11.51	0.586	0.396	0.179	0.443	0.560
YOLOv6-tiny [66]	15.0M	36.7	0.566	0.403	-	-	-
DAMO-YOLO-tiny [67]	8.5M	18.1	0.580	0.418	0.230	0.461	0.585
PPYOLOE-s [68]	7.9M	17.4	0.605	0.430	0.232	0.464	0.569
YOLOv7-tiny [30]	6.2M	13.7	0.567	0.387	0.188	0.424	0.519
Mobile-Former-508M(RetinaNet) [45]	8.4M	-	0.583	0.380	0.229	0.412	0.497
EdgeViT-XXS(RetinaNet) [40]	13.1M	-	0.590	0.387	0.224	0.420	0.516
PVT-tiny(RetinaNet) [60]	23.0M	50.94	0.569	0.367	0.226	0.388	0.500
ConT-m-tiny(RetinaNet) [69]	27.0M	217.2	0.581	0.379	0.230	0.406	0.504
our	6.1M	16.2	0.599	0.410	0.245	0.455	0.535

- 1) Pruning on the neck structure based on the original version can effectively reduce the model redundancy and still ensure accuracy while effectively reducing the number of parameters and computational complexity.
- 2) Our proposed DSLK-Block can effectively improve the feature extraction capability of models for small-sized targets, as well as detect complex environments, and is lightweight.
- 3) The DSLKViT-Block employed in the final model offers a significant advantage in effectively utilizing global information, leading to improvements in accuracy across all aspects. Since this structure acts on the output layer for larger size object detection, it makes the model improvement for such objects more obvious and directly leads to a significant improvement in the mAP.5:.95 metric.

4) Validity of the Proposed Module for Other Models:

In our ablation experiment, we demonstrated that modifying the model structure and adding two modules, DSLK-Block and DSLKViT-Block, can effectively improve the baseline model in terms of parameter size, computational complexity, and accuracy. To further demonstrate the effectiveness and applicability of these two modules, we conducted various model comparisons on the antenna dataset. After each model was tested on the antenna dataset, we added the two modules in turn to the model before further comparison; the experimental results are shown in Table V and Fig. 7(b). The two modules we proposed perform well on various commonly used detectors, and the improvement effect they bring is consistent with the experimental trend on the baseline. The introduction of the DSLK-Block to replace the original convolution modules resulted in a significant improvement in the lightweight nature of all the models. However, due to differences in the complexity of their respective neck structure designs, there were substantial variations in the change in model parameter count with the introduction of the DSLKViT-Block. Nevertheless, without exception, the introduction of both modules led to a noticeable enhancement in the model's detection accuracy. Fig. 8 shows a comparison of the detection performances of our model and the baseline model on the antenna dataset. The DSLK-Block proved instrumental in detecting numerous small targets, particularly patch antennas, which the baseline model failed to identify. When faced with the common background of balcony fences, the baseline model is prone to mistaken local regions for the Yagi antenna, resulting in many false

alarms. When encountering the problem of large interclass differences presented by plate log antennas in multiangle shooting, the baseline model's detection resulted in many FPs. To address these issues, our proposed method combines a CNN and a transformer, introducing the DSLKViT-Block module to effectively resolve the problem. A comparison of the detection figures clearly reveals that our method results in fewer false alarms and misses after the correct targets are detected. Finally, we compare the performance of YOLO-Ant with that of many classical and cutting-edge detectors on the antenna dataset in three dimensions, namely, the model parameter number, accuracy, and computational complexity, as shown in Fig. 7(c), and YOLO-Ant achieves the best performance.

5) *Experiments on Public Datasets:* To demonstrate the generalizability, robustness, and capability of YOLO-Ant, which can excel not only on antenna datasets but also on public datasets, we conducted comparisons with numerous models on two common datasets, COCO and VisDrone. As shown in Tables VI and VII, we compare our model with several other recent outstanding one-stage detectors and several lightweight transformer-based detectors. The results indicate that YOLO-Ant has the fewest parameters and relatively lower computational complexity, outperforming other transformer-based detection models by a significant margin. In terms of detection accuracy, our approach slightly lags behind the top-performing PPYOLO-s in terms of mAP.5 and slightly lags behind PPYOLO-s and DAMO-YOLO-tiny in terms of mAP.5:.95. However, in small object detection, our approach achieves the highest precision among all the models. Fig. 9 shows the detection performance comparison between our model and YOLOv5-s on the COCO dataset. On the VisDrone dataset, YOLO-Ant also achieves competitive results. Overall, considering the parameter count, computational complexity, and detection accuracy, YOLO-Ant is one of the top contenders among lightweight object detectors.

6) *Model Detection Speed Test:* In this experiment, we conducted a speed comparison between YOLO-Ant and several transformer models, as well as YOLO series models known for their lightweight design. We converted all the model weights trained on the antenna dataset into ONNX format and tested them using the OPENVINO tool on a platform with low computational power (Intel Core¹ i9-11900 CPU). Testing was

¹Trademarked.



Fig. 9. Comparison of detection performance between YOLO Ant and baseline models in the COCO dataset. The three images from left to right in each group represent the ground truth, the detection effect of baseline, and YOLO-Ant.

performed on the validation set of the antenna dataset. The comparative results are presented in Table VIII.

The table shows that YOLOv7-tiny achieves a frames per second (FPS) rate exceeding 40, making it the fastest model.

However, this lightweight design comes at the cost of reduced detection accuracy. On the other hand, while YOLO-Ant exhibits a slightly slower detection speed than YOLOv7-tiny does, it outperforms the baseline model YOLOv5-s to

TABLE VII
COMPARATIVE EXPERIMENTS ON THE VISDRONE DATASET

model	Param.	GFLOPs	mAP.5	mAP.5:95
FCOS [57]	-	-	0.258	0.142
VFNet [70]	-	-	0.288	0.168
TOOD [59]	-	-	0.294	0.181
RetinaNet(ResNet50) [25]	37.97M	191.43	0.214	0.118
YOLOx-s [58]	8.97M	26.93	0.254	0.133
YOLOv7-tiny [30]	6.23M	13.86	0.298	0.153
ours	6.13M	16.18	0.295	0.163

TABLE VIII
MODEL DETECTION SPEED TEST

model	Param.(M)	GFLOPs	FPS
YOLOv7-tiny [30]	6.23	13.86	43.82
DETR(ResNet50) [11]	41.58	79.52	3.89
RT-DETR(ResNet18) [65]	20.00	60.00	10.89
ours	6.1	16.2	35.87

a small extent. In addition, the lightweight design of the proposed model is significantly superior to that of the two transformer-based models. This demonstrates the success of our structural design, both in terms of accuracy and lightweight efficiency.

Due to the extensive use of depthwise separable convolutions in our designed modules, which require more memory bandwidth, the input-output (I/O) read speed of the device became the speed bottleneck of the model. We believe that in future work, further optimizations can be made in this regard. This approach enables YOLO-Ant to achieve not only the highest detection accuracy in antenna detection but also further enhance its lightweight design.

V. CONCLUSION

Taking UAV detection of antenna interference sources as the starting point, we propose a lightweight object detector with improved YOLOv5. Initially, to ensure that the model is lightweight and adaptable to subsequent modifications, the new network is first pruned based on YOLOv5, which effectively reduces the number of model parameters and computational complexity while ensuring accuracy. To address the challenges posed by small target sizes and complex backgrounds in antenna detection tasks, we propose an efficient and lightweight convolutional module called DSLK-Block. Furthermore, we introduce a lightweight transformer structure integrated with DSLK-Block, which is applied to the network's neck. This combination significantly enhances the network's ability to extract and process features. The new model not only is effective on the antenna dataset but also achieves competitive results on public datasets such as COCO. In future work, we will further explore the integration of our current efforts with drone inspection technology. Simultaneously, we will incorporate traditional equipment such as spectrum analyzers required for conventional antenna interference source identification. The subsequent objective will be to refine the current approach to create a comprehensive intelligent unmanned detection system.

REFERENCES

- [1] X. Lu, J. Ji, Z. Xing, and Q. Miao, "Attention and feature fusion SSD for remote sensing object detection," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, Art. no. 5501309.
- [2] Z. Yang, Z. Xu, and Y. Wang, "Bidirection-fusion-YOLOv3: An improved method for insulator defect detection using UAV image," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 3521408.
- [3] K. Hao, G. Chen, L. Zhao, Z. Li, Y. Liu, and C. Wang, "An insulator defect detection model in aerial images based on multiscale feature pyramid network," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 3522412.
- [4] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1580–1589.
- [5] M. Tan and Q. V. Le, "EfficientNetv2: Smaller models and faster training," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10096–10106.
- [6] J.-S. Lim, M. Astrid, H.-J. Yoon, and S.-I. Lee, "Small object detection using context and attention," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIC)*, Apr. 2021, pp. 181–186.
- [7] X. Yang et al., "SCRDet: Towards more robust detection for small, cluttered and rotated objects," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 8232–8241.
- [8] C. Yang, Z. Huang, and N. Wang, "QueryDet: Cascaded sparse query for accelerating high-resolution small object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13668–13677.
- [9] S.-J. Ji, Q.-H. Ling, and F. Han, "An improved algorithm for small object detection based on YOLO V4 and multi-scale contextual information," *Comput. Electr. Eng.*, vol. 105, Jan. 2023, Art. no. 108490. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0045790622007054>
- [10] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local networks meet squeeze-excitation networks and beyond," 2019, *arXiv:1904.11492*.
- [11] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. 16th Eur. Conf. Comput. Vis.* Glasgow, U.K.: Springer, 2020, pp. 213–229.
- [12] S. Zheng et al., "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6881–6890.
- [13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
- [15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
- [16] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.
- [17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [18] 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.
- [19] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
- [20] R. Faster, "Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 9199, 2015, pp. 2969239–2969250.
- [21] 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.
- [22] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. 14th Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 21–37.
- [23] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.
- [24] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

- [25] 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.*, Oct. 2017, pp. 2980–2988.
- [26] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
- [27] G. Ghaisi, T.-Y. Lin, R. Pang, and Q. V. Le, "NAS-FPN: Learning scalable feature pyramid architecture for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 7036–7045.
- [28] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10781–10790.
- [29] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
- [30] C.-Y. Wang, A. Bochkovskiy, and H.-Y. Mark Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022, *arXiv:2207.02696*.
- [31] J. Li, Y. Zhang, P. Yun, G. Zhou, Q. Chen, and R. Fan, "RoadFormer: Duplex transformer for RGB-normal semantic road scene parsing," 2023, *arXiv:2309.10356*.
- [32] R. Fan, S. Guo, and M. J. Bocus, *Autonomous Driving Perception*. Springer, 2023.
- [33] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
- [34] 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.
- [35] J. Fu et al., "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3146–3154.
- [36] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
- [37] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
- [38] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.
- [39] S. Mehta and M. Rastegari, "MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer," 2021, *arXiv:2110.02178*.
- [40] J. Pan et al., "EdgeViTs: Competing light-weight CNNs on mobile devices with vision transformers," in *Proc. 17th Eur. Conf. Comput. Vis.* Tel Aviv, Israel: Springer, 2022, pp. 294–311.
- [41] D. Meng et al., "Conditional DETR for fast training convergence," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3651–3660.
- [42] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang, "DN-DETR: Accelerate DETR training by introducing query denoising," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 13619–13627.
- [43] J. Guo et al., "CMT: Convolutional neural networks meet vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12175–12185.
- [44] Z. Peng et al., "Conformer: Local features coupling global representations for visual recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 357–366.
- [45] Y. Chen et al., "Mobile-former: Bridging mobilenet and transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 5270–5279.
- [46] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
- [47] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 11976–11986.
- [48] X. Ding, X. Zhang, J. Han, and G. Ding, "Scaling up your kernels to 31×31: Revisiting large kernel design in CNNs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 11963–11975.
- [49] X. Pan et al., "On the integration of self-attention and convolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 815–825.
- [50] H. Zhang, W. Hu, and X. Wang, "ParC-Net: Position aware circular convolution with merits from ConvNets and transformer," in *Proc. 17th Eur. Conf. Comput. Vis.* Tel Aviv, Israel: Springer, 2022, pp. 613–630.
- [51] L. Guan, L. Jia, Z. Xie, and C. Yin, "A lightweight framework for obstacle detection in the railway image based on fast region proposal and improved YOLO-tiny network," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 5009116.
- [52] 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, 2022, Art. no. 5026114.
- [53] L. Wang, H. Qin, X. Zhou, X. Lu, and F. Zhang, "R-YOLO: A robust object detector in adverse weather," *IEEE Trans. Instrum. Meas.*, vol. 72, 2022, Art. no. 5000511.
- [54] M. A. A. Al-Qaness et al., "An improved YOLO-based road traffic monitoring system," *Computing*, vol. 103, pp. 211–230, Jan. 2021.
- [55] F. Dang, D. Chen, Y. Lu, and Z. Li, "YOLOWeeds: A novel benchmark of YOLO object detectors for multi-class weed detection in cotton production systems," *Comput. Electron. Agricult.*, vol. 205, Feb. 2023, Art. no. 107655, doi: [10.1016/j.compag.2023.107655](https://doi.org/10.1016/j.compag.2023.107655).
- [56] S. Guo et al., "UDTIRI: An online open-source intelligent road inspection benchmark suite," *IEEE Trans. Intell. Transp. Syst.*, early access, doi: [10.1109/TITS.2024.3351209](https://doi.org/10.1109/TITS.2024.3351209).
- [57] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," 2019, *arXiv:1904.01355*.
- [58] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.
- [59] C. Feng, Y. Zhong, Y. Gao, M. R. Scott, and W. Huang, "TOOD: Task-aligned one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 3490–3499.
- [60] W. Wang et al., "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 568–578.
- [61] S. Liu et al., "DAB-DETR: Dynamic anchor boxes are better queries for DETR," 2022, *arXiv:2201.12329*.
- [62] G. Jocher, A. Chaurasia, and J. Qiu. (Jan. 2023). *YOLO by Ultralytics*. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [63] H. Zhang et al., "DINO: DETR with improved denoising anchor boxes for end-to-end object detection," 2022, *arXiv:2203.03605*.
- [64] S. Woo et al., "ConvNeXt v2: Co-designing and scaling ConvNets with masked autoencoders," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 16133–16142.
- [65] W. Lv et al., "DETRs beat YOLOs on real-time object detection," 2023, *arXiv:2304.08069*.
- [66] C. Li et al., "YOLOv6: A single-stage object detection framework for industrial applications," 2022, *arXiv:2209.02976*.
- [67] X. Xu, Y. Jiang, W. Chen, Y. Huang, Y. Zhang, and X. Sun, "DAMO-YOLO: A report on real-time object detection design," 2022, *arXiv:2211.15444*.
- [68] S. Xu et al., "PP-YOLOE: An evolved version of YOLO," 2022, *arXiv:2203.16250*.
- [69] H. Yan, Z. Li, W. Li, C. Wang, M. Wu, and C. Zhang, "ConTNet: Why not use convolution and transformer at the same time?" 2021, *arXiv:2104.13497*.
- [70] H. Zhang, Y. Wang, F. Dayoub, and N. Sünderhauf, "VarifocalNet: An IoU-aware dense object detector," 2020, *arXiv:2008.13367*.



Mr. Tang is a member of the IEEE ICICSP Technical Committee.

Xiaoyu Tang (Member, IEEE) received the B.S. degree from South China Normal University, Foshan, China, in 2003, and the M.S. degree from Sun Yat-sen University, Guangzhou, China, in 2011. He is currently pursuing the Ph.D. degree with South China Normal University.

He is working with the School of Physics, South China Normal University, Guangzhou, China, where he is engaged in information system development. His research interests include machine vision, intelligent control, and the Internet of Things.



Xingming Chen received the B.Eng. degree from the School of Physics and Telecommunication Engineering, South China Normal University, Guangzhou, China, in 2021. He is currently pursuing the M.E. degree with the Department of Electronics and Information Engineering, South China Normal University.

His research interests include computer vision and deep learning.



Jintao Cheng received the B.Eng. degree from the School of Physics and Telecommunications Engineering, South China Normal University, Guangzhou, China, in 2021.

His research interests include computer vision, simultaneous localization and mapping (SLAM), and deep learning.



Jin Wu (Member, IEEE) was born in Zhenjiang, China, in 1994. He received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2016. He is currently pursuing the Ph.D. degree with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology (HKUST), Hong Kong.

From 2013 to 2014, he was a Visiting Student with Groep T, Katholieke Universiteit Leuven (KU Leuven), Leuven, Belgium. He has coauthored over

120 technical papers in representative journals and conference proceedings.

Mr. Wu was a Committee Member of the IEEE CoDIT conference in 2019, the Special Section Chair of the IEEE ICGNC conference in 2021, the Special Session Chair of the 2023 IEEE International Conference on Intelligent Transportation Systems (ITSC), the Track Chair of the 2024 IEEE International Conference on Consumer Electronics (ICCE), and the Chair of the 2024 IEEE CTSoc Gaming, Entertainment and Media (GEM) Conference. He was selected as the World's Top 2% Scientist by Stanford University and Elsevier in the 2020, 2021, and 2022 year round. He was awarded the outstanding reviewer of IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT in 2021. He is currently a Review Editor of *Frontiers in Aerospace Engineering* and an invited guest editor for five special issues of MDPI. He is also in the IEEE Consumer Technology Society (CTSoc) as a Committee Member and a Publication Liaison.



Rui Fan (Senior Member, IEEE) received the B.Eng. degree in automation from Harbin Institute of Technology, Harbin, China, in 2015, and the Ph.D. degree in electrical and electronic engineering from the University of Bristol, Bristol, U.K., in 2018, under the supervision of Prof. John G. Rarity and Prof. Naim Dahnoun.

He worked as a Research Associate (supervisor: Prof. Ming Liu) at The Hong Kong University of Science and Technology, Hong Kong, from 2018 to 2020, and a Post-Doctoral Scholar-Employee (supervisors: Prof. Linda M. Zangwill and Prof. David J. Kriegman) at the University of California at San Diego, La Jolla, CA, USA, from 2020 to 2021. He began his faculty career as a Full Research Professor with the College of Electronics and Information Engineering, Tongji University, Shanghai, China, in 2021, where he was then promoted to a Full Professor and the Shanghai Research Institute for Intelligent Autonomous Systems in 2022. His research interests include computer vision, deep learning, and robotics.

Dr. Fan served as a Senior Program Committee Member for Association for the Advancement of Artificial Intelligence (AAAI) from 2023 to 2024. He was honored by being included in the Stanford University List of Top 2% Scientists Worldwide in 2022 and 2023, recognized on the Forbes China List of 100 Outstanding Overseas Returnees in 2023, and acknowledged as one of the Xiaomi Young Talents in 2023. He is the General Chair of the AVVision Community and organized several impactful workshops and special sessions in conjunction with IEEE Winter Conference on Applications of Computer Vision (WACV) in 2021, IEEE International Conference on Image Processing (ICIP) from 2021 to 2023, International Conference on Computer Vision (ICCV) in 2021, and European Conference on Computer Vision (ECCV) in 2022. He served as an Associate Editor for IEEE International Conference on Robotics and Automation (ICRA) in 2023, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) in 2023, and IROS in 2024.



Chengxi Zhang (Member, IEEE) was born in Shandong, China, in February 1990. He received the B.S. and M.S. degrees in microelectronics and solid-state electronics from Harbin Institute of Technology, Harbin, China, in 2012 and 2015, respectively, and the Ph.D. degree in control science and engineering from Shanghai Jiao Tong University, Shanghai, China, in 2019.

He was a Post-Doctoral Fellow with the School of Electronic and Information Engineering, Harbin Institute of Technology, Shenzhen, China, from 2020 to 2022. His research interests include embedded system software and hardware design, information fusion, and control theory.



Zebo Zhou was born in Yongchuan, Chongqing, China, in November 1982. He received the B.Sc. and M.Sc. degrees from the School of Geodesy and Geomatics, Wuhan University, Wuhan, China, in 2004 and 2006, respectively, and the Ph.D. degree from the College of Surveying and Geoinformatics, Tongji University, Shanghai, China, in 2009.

In 2009 and 2015, he was a Visiting Fellow with the Surveying and Geospatial Engineering Group, School of Civil and Environmental Engineering, University of New South Wales, Kensington, NSW, Australia. He is currently an Associate Professor with the School of Aeronautics and Astronautics, University of Electronic Science and Technology of China, Chengdu, China. He has been in charge of projects of the National Natural Science Foundation of China and has taken part in the National 863 High-tech Founding of China. His research interests include global navigation satellite system (GNSS) navigation and positioning, GNSS/inertial navigation system (INS) integrated navigation, and multisensor fusion.

Dr. Zhou was a Guest Editor of several special issues published on *International Journal of Distributed Sensor Networks* and *Asian Journal of Control*. He has presented related works on the annual conference of the Institute of Navigation and the Annual Chinese Satellite Navigation Conference for several times and has received the best paper awards at these conferences.