A CNN-Transformer Hybrid Model Based on CSWin Transformer for UAV Image Object Detection

Wanjie Lu, Chaozhen Lan, Chaoyang Niu, Wei Liu, Liang Lyu, Qunshan Shi, and Shiju Wang

***Abstract*—The object detection of unmanned aerial vehicle (UAV) images has widespread applications in numerous fields; however, the complex background, diverse scales, and uneven distribution of objects in UAV images make object detection a challenging task. This study proposes a convolution neural network transformer hybrid model to achieve efficient object detection in UAV images, which has three advantages that contribute to improving object detection performance. First, the efficient and effective cross-shaped window (CSWin) transformer can be used as a backbone to obtain image features at different levels, and the obtained features can be input into the feature pyramid network to achieve multiscale representation, which will contribute to multiscale object detection. Second, a hybrid patch embedding module is constructed to extract and utilize low-level information such as the edges and corners of the image. Finally, a slicing-based inference method is constructed to fuse the inference results of the original image and sliced images, which will improve the small object detection accuracy without modifying the original network. Experimental results on public datasets illustrate that the proposed method can improve performance more effectively than several popular and state-of-the-art object detection methods.**

***Index Terms*—Convolutional neural network (CNN), hybrid network, object detection, transformer, unmanned aerial vehicle (UAV) image.**

I. Introduction

With the development of remote sensing technologies, unmanned aerial vehicles (UAVs) have been widely employed in various fields, such as digital cities, smart agriculture and forestry, and disaster inspection. As one of the key technologies to realize the application of UAV images, object detection based on UAV images has been widely employed in military and civilian areas. Traditional object detection methods, such as support vector machines and AdaBoost, have problems, such as cumbersome manual feature design, poor robustness, and computational redundancy, which cannot meet the current needs of UAV image object detection tasks. Driven by breakthroughs in computer vision (CV) and deep learning in recent years, UAV image object detection methods based on deep learning, such as convolutional neural networks (CNN) and transformers, are becoming a field of rapid development and intense research [1], [2].

CNN-based object detection methods have excellent abilities of adaptive learning and feature extraction and have superior detection performance to the traditional object detection methods [3], [4]. For example, two-stage detection methods, such as R-CNN [5], Fast R-CNN [6], Faster R-CNN [7], Mask R-CNN [8], Cascade R-CNN [9], and Dynamic R-CNN [10], can classify the object regions by extracting features of several candidate regions, and then obtain the object categories and positions; one-stage detection methods, such as you only look once (YOLO) series [11], single-shot multibox detector (SSD) [12], RetinaNet [13], and fully convolutional one-stage object detection (FCOS) [14], can achieve end-to-end object location and category prediction directly through the initial anchor box without region proposals. However, CNN-based object detection methods mainly use large-scale proposals, anchors, or window centers for category predictions. Furthermore, repeated prediction boxes, anchor box design, and assignment between objects and anchor boxes seriously affect the model's performance in postprocessing, and global features, such as long-distance dependencies in the processing, cannot be effectively obtained.

With the development of the attention mechanism, CV researchers have gradually attached importance to transformer-based object detection methods [15], which have achieved competitive performance in multiple CV tasks. A transformer is an encoder-decoder sequence transformation model that enables long-range interactions between different encoded elements in a sequence using the self-attention mechanism. With self-attention, long-distance dependency modeling capabilities can be achieved to enable data processing in various downstream tasks. Given these advantages, the transformer, which was originally mainly used in natural language processing, has been promptly introduced into CV tasks, remote sensing, and related fields. For example, by applying the encoder in the transformer directly to the image patches, Dosovitskiy et al. [16] proposed a convolution-free method called vision transformer (ViT), which improves the object detection performance without major modifications to the original model architecture. Bazi et al. [17] built a remote sensing image classification method based on ViT, which exhibits superior performance on four public datasets.

Manuscript received 26 September 2022; revised 1 December 2022; accepted 1 January 2023. Date of publication 4 January 2023; date of current version 16 January 2023. This work was supported by the National Natural Science Foundation of China under Grant 42201472, Grant 41901378, and Grant 42001338. *(Corresponding author: Chaozhen Lan.)*

Wanjie Lu, Chaoyang Niu, Wei Liu, and Shiju Wang are with the Institute of Data and Target Engineering, PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, China (e-mail: lwj285149763@163.com; niucy2017@outlook.com; greatliuliu@163.com; 13733150660@139.com).

Chaozhen Lan, Liang Lyu, and Qunshan Shi are with the Institute of Geospatial Information, PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, China (e-mail: lan_cz@163.com; lvliangvip@163.com; hills1@163.com).

Digital Object Identifier 10.1109/JSTARS.2023.3234161

One key factor for the success of transformer-based methods lies in large-scale training datasets and high-performance computing resources. However, in conditions of small data volume or insufficient training time, CNN-based models with the same number of parameters can achieve better performance, whereas transformer-based models require enough training data and time to achieve similar performance. The main reason is that CNN-based models have inductive biases properties, such as locality and translation equivariance [18], which limit the application of transformer-based models in areas with limited data or computing resources. In addition, the perception fields of the CNN and transformer-based models are considerably different. Based on the advantages of the self-attention mechanism, transformer-based models can perform better in capturing the relationships among distant pixels in images. However, because of the lack of capturing the internal spatial information of image patches, the local information, such as texture and corner, is lost in transformer-based models, whereas CNN-based models have advantages in capturing relevant information [19].

To make up for the deficiencies of CNN-based and transformer-based models, CNN-transformer hybrid methods [20], [21], [22] are used to effectively integrate CNN and transformer to improve the overall performance. However, for UAV image object detection, effectively fusing the local and global features in the CNNs and transformers requires further research; in addition, the CNN-transformer hybrid models still need relatively large data and a long time, otherwise good performance cannot be achieved.

Meanwhile, because of the characteristics of UAV images, deep learning models face challenges in object detection. The candidate objects in UAV images have the characteristics of different luminance, complex backgrounds, scale diversity, etc. For example, the diversity in camera angles leads to various characteristics of the same category [23]; the objects in the UAV images under the large field of view show an uneven distribution, such as a dense aggregation of objects in the city center [24], whereas various objects in the suburbs are sparsely distributed [25]. In addition, natural factors, such as clouds, rain, fog, and snow, can lead to the failure of object detection in UAV images [26]. The above factors make it difficult for the current image object detection methods in general scenarios to achieve ideal performance for UAV images.

Considering the pros and cons of CNNs and transformers, and according to the characteristics of UAV images and object detection requirements, a hybrid object detection method for UAV images that combines CNNs and transformers is proposed. In summary, the main contributions are as follows.

1) A method of using an efficient transformer, cross-shaped window (CSWin) transformer, as the backbone of the Mask R-CNN is proposed to effectively achieve multiscale UAV image object detection. The CSWin Transformer is used to establish the dependence of long-distance features in the input image and obtain features at different levels. Combined with the feature pyramid network (FPN), the multiscale representations of the obtained features at different levels are realized to meet the need for effective detection of multiscale objects in images.
2) A hybrid patch embedding module (HPEM) to process the input image is constructed. Using convolution to process the input image into low-dimensional features, and then generating sequence token embeddings, low-level information, such as edges and corners, can be extracted and utilized without hardly increasing the number of parameters in the model.
3) Given the high-resolution characteristics of UAV images, a slicing-based inference (SI) method is constructed. While using the trained model to infer the original input UAV image, the input image is sliced, and it is inferred using the trained model after amplification and enhancement. The inference results based on slicing are fused with the original image inference results to realize further detection of small objects in the input image, which can improve the object detection accuracy without modifying the original model.

The remainder of this article is organized as follows. Section II describes the related works and Section III describes the proposed method in detail. The experiments and results of the proposed and compared methods are presented in Section IV. Finally, in Section V, we discuss and conclude this study.

II. Related Works

This section will discuss related work from three aspects: CNN-based object detection for UAV images, ViT, and CNN-transformer hybrid methods.

A. CNN-Based Object Detection

CNN-based object detection models can be roughly divided into two categories: two-stage detectors and one-stage detectors.

1) *Two-Stage Detectors:* Based on the powerful feature representation ability of CNN, R-CNN [5], a typical two-stage detector, was proposed and has considerably improved the performance of object detection. Based on R-CNN, Fast R-CNN [6] was proposed by combining the spatial pyramid pooling network (SPP-Net) [27] and using region of interest (ROI) pooling. Ren et al. [7] proposed Faster R-CNN, which used the region proposal network (RPN) instead of the selective search algorithm to realize sharing of convolutional features, thereby further improving the detection speed. By adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition, He et al. [8] proposed Mask R-CNN, in which a small overhead is added to Faster R-CNN. In addition, two-stage detectors also include Cascade R-CNN [9], HTC [28], and Dynamic R-CNN [10].
2) *One-Stage Detector:* Redmon et al. [29] proposed YOLO, which just used a single neural network to complete the process from image input to output of object location and category information. Based on YOLO, a series of improved algorithms, such as YOLO9000 [30], YOLOv3 [31], and YOLOv4 [32], have been proposed. Liu et al. [12] proposed SSD to extract multiscale features. To solve the problem of object detection in a per-pixel prediction fashion, analog to semantic segmentation, FCOS [14],

a fully convolutional and anchor box-free one-stage object detector, was proposed, which completely avoids the complicated computation related to anchor boxes such as calculating overlapping during training. Lin et al. [13] discovered that the extreme foreground-background class imbalance encountered during the training of dense detectors affected the performance of one-stage detectors and proposed RetinaNet using focal loss to prevent the vast number of easy negatives from overwhelming the detector during training.

Both one and two-stage detectors can be applied to object detection tasks. However, due to the characteristics of UAV images, the above algorithms cannot fully exploit the performance of UAV image object detection. For UAV image object detection, multiscale detection is a frequent and common scenario, which is characterized by the simultaneous existence of multiscale object instances. For example, in the Visdrone2021-DET dataset, the size of the persons occupies less than 6% of the image, whereas some vehicles can occupy more than 20% of the image [33].

Currently, the most common method to achieve multiscale detection involves constructing multiscale feature maps [34] and obtaining the output results by carrying out multilayer filtering [35], [36]. In [37], the FPN used a top-down feature fusion method to achieve multiscale object detection by fusing the low-level features with more details and the top-level features with rich semantic information. Through research focused on multiscale object proposal networks, multiscale object detection was achieved by generating candidate regions with different intermediate layer features [35]. Zhang et al. [38] proposed a dual multiscale FPN framework and studied multiple training and inference strategies of multiscale object detection. Using a multiscale information preservation module, Han et al. [39] constructed multiscale pyramid images and features for each image to retain as much multiscale information of the input data as possible, which is helpful to achieve better performance of object detection. To adaptively combine multiscale feature information on different channels and spatial positions, FPN was used to obtain more discriminative features and achieve an efficient fusion of multiscale features [40]. In addition, the dilated/deformable convolution kernel [41], [42] was used to expand the receptive field of algorithms without loss of resolution to achieve multiscale object detection.

Otherwise, for detecting small and dense objects in UAV images, better performance can be achieved by improving feature maps of small objects, incorporating context information of small objects, and using data enhancement methods [43], [44]. Based on the above ideas, researchers have constructed a series of small object detection networks including RRNet [45], FSSSD [44], Cascade network [46], HRDNet [47], UAV-YOLO [48], MPFPN [49], and GANet [50]. To solve the problems of small instances, complex backgrounds, and difficult feature extractions in UAV images, because FPN can effectively integrate multiscale features to achieve small object detection and context information can provide more powerful information, DBNet [51] and SINet [52] obtained better small object detection results by providing global contextual information.

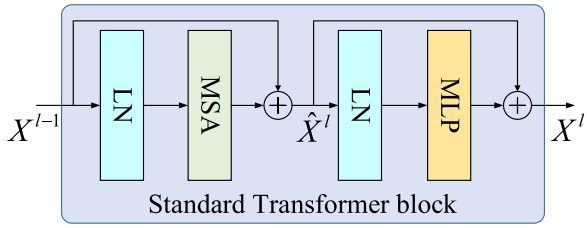


Fig. 1. Standard architecture of the transformer bock.

B. Transformer-Based Object Detection

Nowadays, transformer-based models, such as DETR [53] and ViT [16], have been widely used in CV tasks. The standard transformer block generally includes a multihead self-attention, a multilayer perceptron, and multiple layer normalizations [54], as shown in Fig. 1.

To simplify the work pipelines of current object detectors, Carion et al. [53] optimized the training process by transforming the object detection task into a direct set prediction problem and constructed end-to-end object detection with transformers (DETR), which models the interactions among different elements in a sequence using the encoder-decoder structure in the transformer. Subsequently, researchers carried out optimizations and improvements around DETR and proposed models such as Deformable DETR [55], Conditional DETR [56], DN-DETR [57], DAB-DETR [58], and DETR with improved denoising anchor boxes (DINO) [59].

Similar to CNN-based models, ViT [16], a two-branch transformer structure, was designed to learn features at different scales and demonstrated that multiscale feature representation is effective. To solve the problems of requiring a large amount of data and high-performance computing for training in ViT, DeiT [60], a data-efficient transformer-based model that is similar to ViT, was proposed. However, different from ViT, DeiT mainly achieves efficient training and better results on small datasets (e.g., ImageNet1K) through the self-attention mechanism and through the knowledge distillation method. Bashmal et al. [61] proposed a multilabel classification method based on a data-efficient transformer, which achieved the multilabel efficient classification of high-resolution UAV remote sensing images. Ranftl et al. [62] constructed a dense prediction transformer to perform dense object predictions by upsampling the low-resolution images to obtain high-resolution images. Based on the design philosophy of depthwise separable convolution, separable vision transformer (SepViT) [63] was designed to realize the information interaction within and among windows through depthwise separable attention, which effectively improves the computational efficiency of ViT.

However, the transformer has inherent defects. During the training process, the transformer produces a quadratic computational complexity related to the image resolution or the number of tokens, resulting in a huge amount of attention calculation when dealing with long sequence tokens. Therefore, computational complexity is a key consideration when the transformer is applied to the field of object detection. To effectively improve the

computational performance of the transformer, a typical solution involves reducing the scope of the attention mechanism from global to local or a certain window. Swin Transformer [64] is a hierarchical transformer whose representation was computed with shifted windows. Swin Transformer divides the image into nonoverlapping local windows and uses the shift window mechanism to limit the self-attention computation to the segmented local windows, which effectively improves computing efficiency. However, the receptive field in Swin Transformer expands slowly and requires numerous computational blocks to finally obtain global attention. To solve the problem of the limited token interaction field in the local self-attention, and effectively reduce the amount of computation while obtaining a wide range of attention simultaneously, the CSWin Transformer [65], which achieved good performance on common CV tasks by computing self-attention in the horizontal and vertical stripes in parallel that form a CSWin, was proposed. Beyond those, PVT [66] was proposed to build an attention layer with linear complexity to achieve improved computational efficiency using down-sampled keys and values, and PVT v2 [67] was presented by improving PVT by adding a linear complexity attention layer, overlapping patch embedding, and using a convolutional feed-forward network. Different from the above methods, Tang et al. [68] designed an efficient visual transformer using quadtree attention to divide the input images into multiple patches and evaluate the attention score to achieve quadratic complexity reduction, which could reduce the quadratic computational complexity to linear complexity and simultaneously obtain better-detailed information and long-distance dependencies to achieve better results in CV tasks.

Although a series of transformer-based methods can achieve good performance by combining the hierarchical structure and can be used as a general backbone in CV tasks, such as object detection [69], [70], instance segmentation [72], and other fields, their actual performance varies considerably. Among them, CSWin Transformer has a relatively simple self-attention mechanism, which makes CSWin Transformer much more effective for general vision tasks compared with the Swin Transformer, PVT, and other methods. Therefore, the CSWin Transformer was used as the backbone of this study.

C. CNN-Transformer Hybrid Methods for Object Detection

By constructing CNN-transformer hybrid models, the advantages of CNNs and transformers can be effectively integrated. Zheng et al. [68] proposed an adaptive and dynamic one-stage detector based on the feature-pyramid transformer, which enhanced the feature fusion ability of the model by embedding a transformer in the FPN. Xu et al. [70] constructed a local-perception backbone based on Swin Transformer to enhance the local-perception capability and improve the detection accuracy for small objects. By replacing the original prediction heads with transformer prediction heads (TPH), TPH-YOLOv5 [71] achieved good performance with impressive interpretability on drone-captured scenarios. Feng et al. [73] used YOLO as the baseline network and used a cross-stage partial (CSP) bottleneck transformer module as the backbone to implement the transitivity of the global spatial dependencies, which exhibit superior performance in different application environments and have high generality. Inspired by DETR, Li et al. [74] constructed a novel transformer-based remote sensing object detection framework, TRD, in which a modified transformer was designed to aggregate features of global spatial positions on multiple scales and the interactions between pairwise instances were modeled. Using the transformer as a branch network of the one-stage detection network, Zhang et al. [75] constructed GANsformer to utilize the feature information in the entire region and improved the model's ability to detect objects in aerial images. The above studies indicate that the transformer mainly functions in various CNN-based detection frameworks as a feature interaction module.

However, the characteristics of UAV images make current CNN-based, transformer-based, and CNN-transformer hybrid methods face challenges in object detection. First, the object scale in UAV images varies significantly because of the considerable change in flight altitude. Second, the objects in UAV images have different luminance and complex backgrounds. Third, the distribution of various objects in UAV images is uneven and varies dramatically. These above challenges make it difficult for the current methods to achieve the desired detection performance. Therefore, inspired by these excellent works, given the characteristics of UAV images, this study combines a transformer with CNN to build an object detection model for UAV images. Through improvement and optimization of the proposed model, the detection performance is improved to better meet the requirements of object detection in UAV images.

III. Proposed Method

In this section, the overall architecture of the proposed method is introduced first, and then, the relevant key components and modules are described in detail.

A. Overview Framework

The pipeline of the proposed CNN-transformer hybrid network model for UAV image object detection is shown in Fig. 2, which mainly includes the following modules.

1) The object detection network, Mask R-CNN, is used as the pipeline network, in which a bottom-up hierarchical structure is used to extract feature maps. By combining with FPN, a top-down hierarchical structure with lateral connections will fuse features, thereby obtaining high-level semantic information of different scales and realizing multiscale object detection through the RPN and ROI head.
2) An efficient and effective transformer-based backbone, CSWin Transformer, is used in each stage to extract features. The CNN-based backbone cannot effectively acquire long-distance interactions and dependencies, while the global self-attention mechanism requires a significant amount of computation, and the interaction field between different tokens in the local self-attention mechanism is

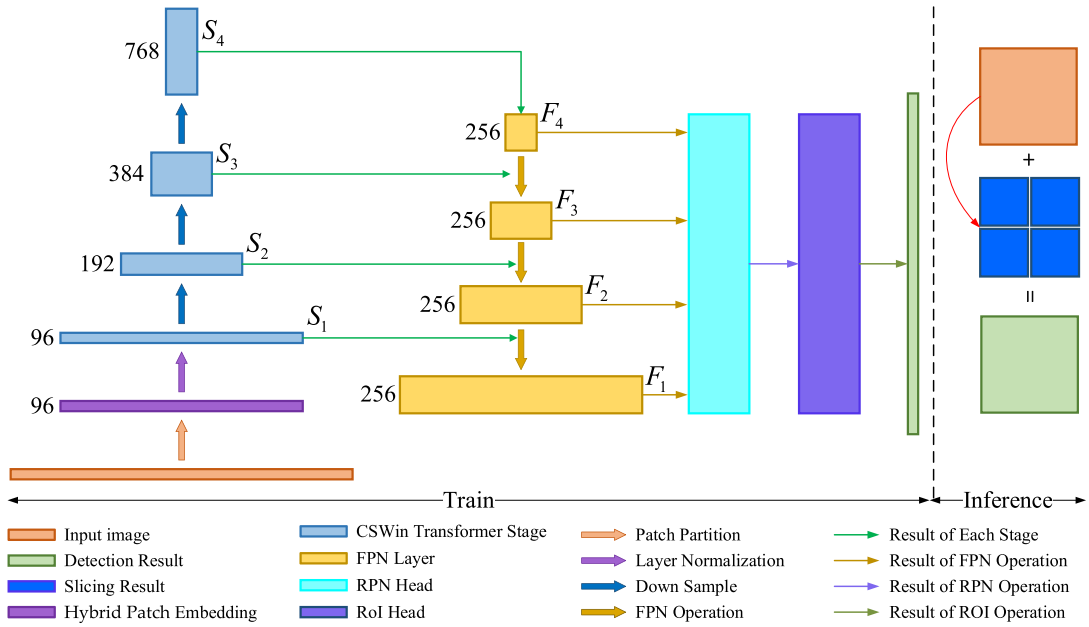


Fig. 2. Pipeline of the proposed CNN-transformer hybrid model for UAV image object detection. The left and right parts of the dotted line are the training and inference stages, respectively.

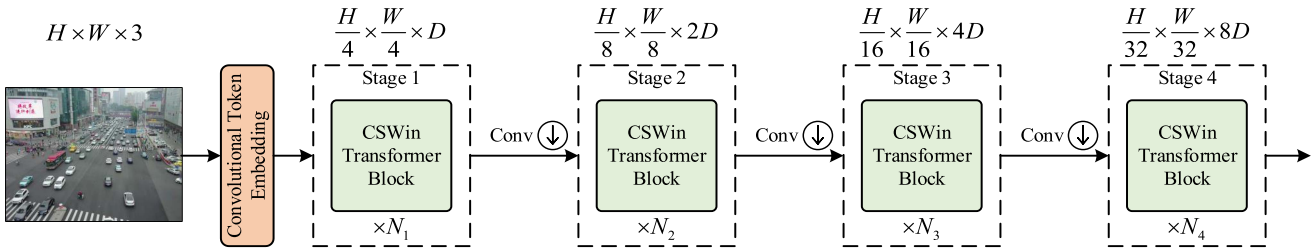


Fig. 3. Architecture of the CSWin transformer-based encoder.

limited. CSWin Transformer effectively solves the above problems by computing self-attention in the horizontal and vertical stripes in parallel.

3) An HPEM is proposed to process the input image. Through convolution, low-level feature maps are obtained. Then, these feature maps are flattened into a sequence of patches using a patch embedding module, which can effectively obtain and utilize low-level information such as edges and corners in the image while barely increasing the training data and iterations.
4) A SI method is constructed in the inference stage after completing model training to further improve the detection performance of small objects in high-resolution UAV images. The SI method divides the input image into overlapping sliced images with a fixed size, which will be reidentified. Subsequently, the reidentification results are fused with the original image object detection results through nonmaximum suppression (NMS) to further improve the object detection performance of UAV images.

B. CSWin Transformer–Based Backbone

In this study, the CSWin Transformer is used as the encoder backbone in the bottom-up hierarchical structure to extract feature maps, and the encoded results obtained are input into the top-down FPN structure for decoding to achieve multiscale object detection.

1) CSWin Transformer–Based Network Structure: The hierarchical network structure constructed with the CSWin Transformer as the backbone is shown in Fig. 3. First, the input image with size $H \times W \times C$ (H, W, and C represent the height, width, and the number of channels, respectively; for the input image, $C = 3$) is processed to obtain patch tokens, which are used as the input of the subsequent hierarchical structure. Then, to produce a hierarchical feature representation, the hierarchical network consists of four stages based on the CSWin Transformer, and the output of each stage is down-sampled using a convolution with fixed kernel size and stride, so that H and W are reduced by half, while the number of channels is doubled. The output feature map of the ith stage has $(H/2^{i+1}) \times (W/2^{i+1})$ tokens,

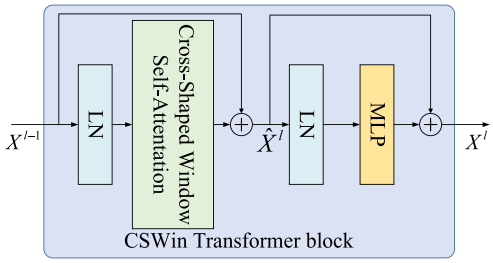


Fig. 4. CSWin transformer block.

which is similar to those in the Swin Transformer. However, different from Swin Transformer, CSWin Transformer replaces the patchily stem with a convolutional stem, which achieves better training efficiency while ensuring overall stability [76]. Finally, the output of the hierarchical network is processed by feature fusion and prediction in the RPN layer and ROI layer to achieve object detection.

2) CSWin Transformer Block: The global attention mechanism is computationally expensive, whereas the local attention mechanism limits the interaction between different tokens, and requires more computing blocks to achieve global attention. Therefore, unlike the vanilla transformer, CSWin Transformer adopts CSWin self-attention to obtain global attention more effectively, as shown in Fig. 4.

Specifically, CSWin self-attention divides the input data vertically and horizontally according to a given size to obtain horizontal and vertical stripes and performs self-attention calculation in horizontal and vertical stripes in parallel, as shown in Fig. 5. The main processing flow is as follows.

First, the input data X with size $H \times W \times C$ is linearly projected to K heads, which are divided into two parts equally, such as *Part A* and *Part B* in Fig. 5. Each part has $K/2$ heads.

Then, *Part A* and *Part B* are segmented vertically and horizontally, respectively, to obtain horizontal and vertical stripes, and self-attention calculation is performed in the horizontal and vertical stripes in parallel. We take *Part A* in Fig. 5 as an example. *Part* A is evenly divided into nonoverlapping horizontal stripes $[X^1, \ldots, X^M]$ with width SW along the vertical direction, where $M = H/SW$, and the dimension of each strip X^i is $SW \times W \times C$, where $i = 1, 2, \ldots, M$.

After that, multihead attention is computed for each stripe X^i, where the attention Y_k^i of the kth head is defined as

$$Y_k^i = \text{Attention}(X^i W_k^Q, X^i W_k^K, X^i W_k^V) \tag{1}$$

where $W_k^Q \in \mathbf{R}^{C \times d_k}$, $W_k^K \in \mathbf{R}^{C \times d_k}$, $W_k^V \in \mathbf{R}^{C \times d_k}$, and $d_k = C/K$. The overall horizontal stripes self-attention for kth head can be defined as

$$H\text{−Attention}_k(X) = [Y_k^1, Y_k^2, \ldots, Y_k^M]. \tag{2}$$

The overall vertical stripes' self-attention $V\text{−Attention}_k(X)$ for the kth head can be performed with reference to the horizontal stripes self-attention. After concatenating the self-attention in horizontal and vertical stripes, the final result can be defined as

$$\begin{aligned} &\text{CSWin} - \text{Attention}(X) \\ &= \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_K) W^O \end{aligned} \tag{3}$$

where $W^O \in \mathbf{R}^{C \times C}$, and

$$\text{head}_k = \begin{cases} H\text{−Attention}_k(X) & k = 1, \ldots, K/2 \\ V\text{−Attention}_k(X) & k = K/2+1, \ldots, K. \end{cases} \tag{4}$$

Using $\text{CSWin} - \text{Attention}(X)$, the attention area of each token within one transformer block can be enlarged. In addition, as the stage increases, CSWin Transformer associates more regions by increasing the strip width SW.

Finally, the processing of the CSWin Transformer block can be formally expressed as

$$\hat{X}^l = \text{CSWin} - \text{Attentation}(LN(X^{l-1})) + X^{l-1} \tag{5}$$

$$X^l = \text{MLP}(LN(\hat{X}^l)) + \hat{X}^l \tag{6}$$

where X^l represents the output of the lth CSWin Transformer block or the precedent convolutional layer of each stage.

C. FPN-Based Decoder

Multiscale object detection is a basic challenge in UAV image object detection, and constructing a feature pyramid is an effective technique. In CNN-based models, different network depths correspond to different levels of semantic features. The low-level features have high resolution and rich detailed information, whereas the high-level features have low resolution and rich semantic information. However, in high-level features, small objects have a higher probability of being ignored. To make full use of the different level features extracted, FPN uses the multiscale and hierarchical structure of deep convolutional networks to construct feature pyramids, which will be fused through a top-down and laterally connected structure to obtain high-level semantic features at different scales, as shown in Fig. 6. FPN can be used as a general feature extractor and can be combined with different backbones to achieve performance improvement.

The bottom-up pathway of FPN is implemented by the feed-forward calculation of CNN. By referring to the idea of a hierarchical structure, data processing is divided into different stages according to the size of the feature map, and the scale of the feature map of each stage is half of the previous stage. Using this structure, FPN can obtain more abundant feature information, and the output of each stage can be used as part of the input features of the corresponding level in the top-down pathway in the FPN.

The top-down pathway of FPN provides higher resolution features by up-sampling spatially coarser, but semantically stronger feature maps from higher pyramid levels, which will make it the same size as the feature map output from the corresponding stage in the bottom-up pathway. Then, lateral connections are employed to enhance these features using information from the bottom-up pathway, and the output features serve as the input of the next layer in the top-down pathway. To combine the semantic

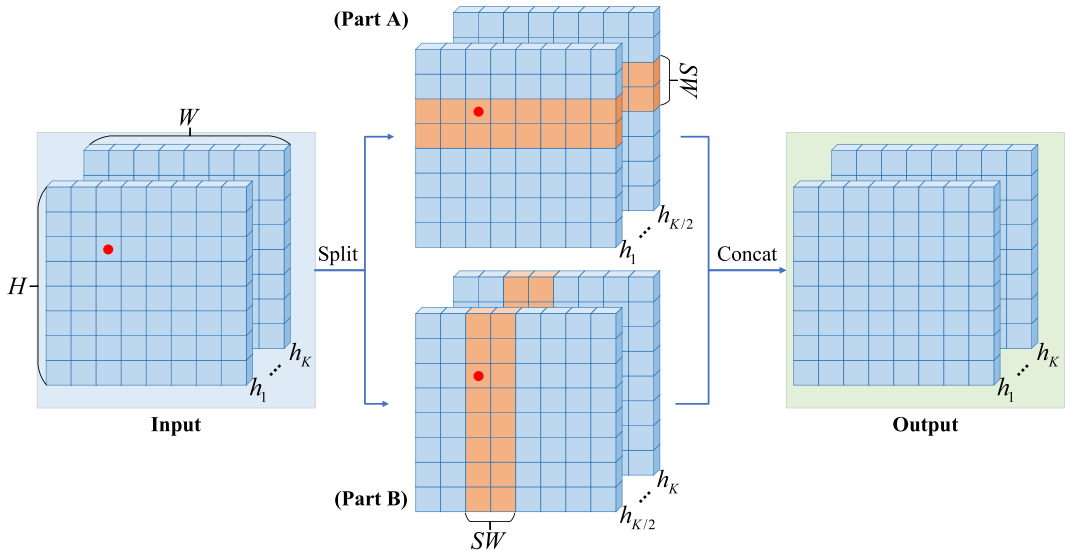


Fig. 5. Cross-shaped window self-attention.

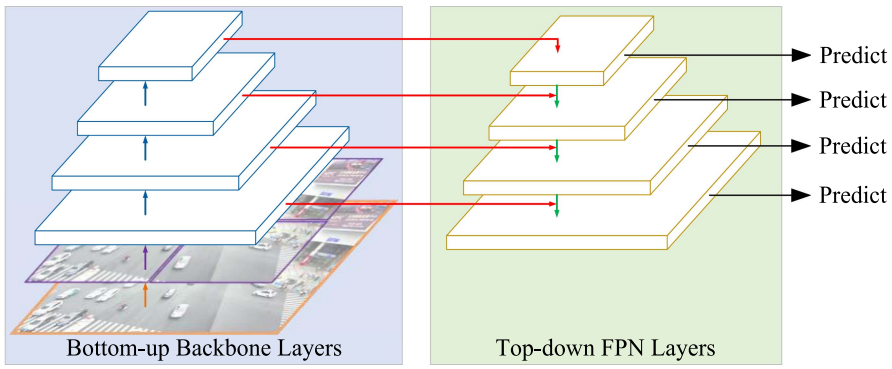


Fig. 6. Architecture of the FPN.

information of the high-level features with precise positioning information of the low-level features, each lateral connection adopts a structure similar to a residual network; to correct the number of channels, the feature map output by the corresponding level stage in the bottom-up pathway needs to be processed by a 1×1 convolution.

Based on the above characteristics, FPN utilizes not only high-level strong semantic features for classification but also low-level high-resolution information for localization.

D. Hybrid Patch Embedding Module (HPEM)

Typically, before being input into the transformer-based backbone, the raw image must be processed to generate a sequence of token embeddings. For example, ViT splits the input image with a patch size of 16×16 or 32×32, and the Swin Transformer or CSWin Transformer splits each image with a patch size of $H/4 \times W/4$ (H and W represent the height and width of the input image), which makes it difficult to capture low-level information (such as edges and corners), and requires much more training data or training iterations. Given the above situation, for capturing more detailed low-level information and enriching the features extracted by the backbone, this study proposes an HPEM, as shown in Fig. 7.

Before the original patch embedding module, HPEM adds a convolution layer to generate low-level features; simultaneously, to better facilitate the training process, batch normalization is added after the convolution layer. Subsequently, by flattening these low-level features into sequence tokens using the original patch embedding module, the low-level information in the image can be exploited to the fullest while hardly increasing the computational cost. The entire process of HPEM can be expressed as follows:

$$\mathbf{x}' = \mathrm{HPEM}(\mathbf{x}) = \mathrm{PatchEmbedding}(\mathrm{BN}(\mathrm{Conv}(\mathrm{x}))) \quad (7)$$

where $\mathbf{x}' \in \mathbb{R}^{\frac{H}{S} \times \frac{W}{S} \times C}$, and S denotes the stride in the input image. HPEM makes full use of the advantages of CNN in extracting low-level features and reduces the training difficulty of implantation by reducing the patch size.

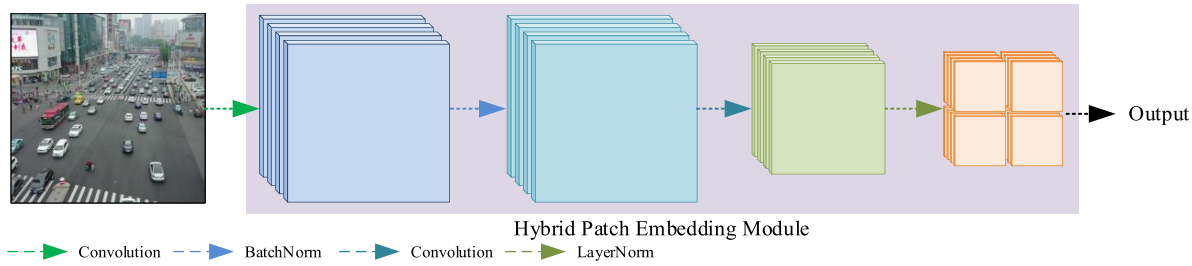


Fig. 7. Hybrid patch embedding module.

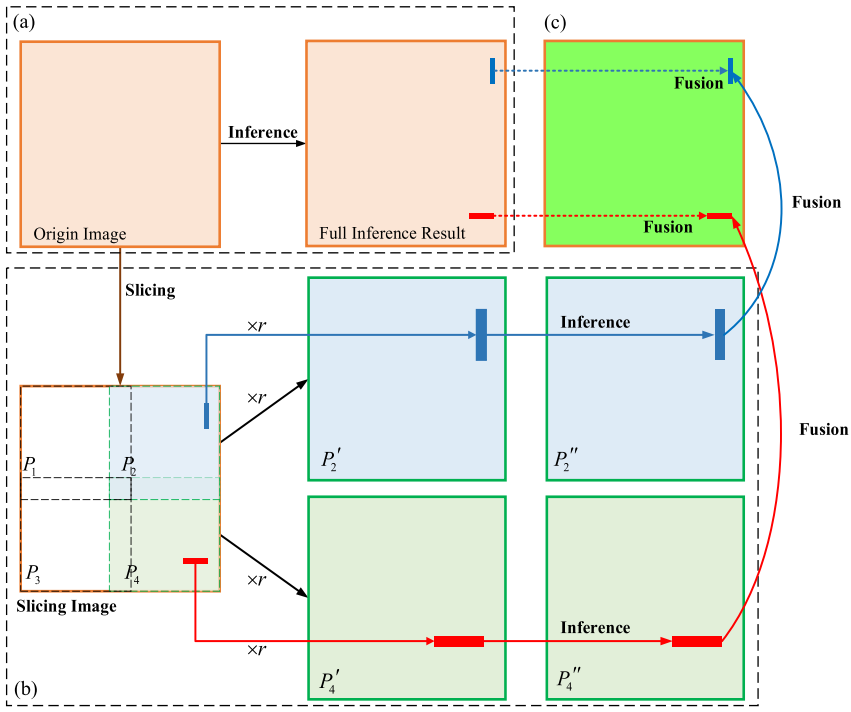


Fig. 8. Basic principles and steps of slicing-based inference.

E. Slicing-Based Inference

Influenced by the flying height and the shooting angle of UAVs, the obtained images contain numerous small objects at long distances or oblique angles. These small objects occupy fewer pixels in images, and details are insufficient, which affects the object detection performance. Although the current object detection algorithm achieves multiscale object detection by constructing feature pyramids, the performance of small object detection is expected to be improved further.

Based on the slicing-aided hyper inference mechanism [77], this study constructs the SI to further improve the performance of small object detection in UAV images. The mechanism of the SI is shown in Fig. 8, whose basic principles and steps are as follows:

Step 1: use the model trained by the proposed method in this article to perform inference on the original UAV images to achieve normal inference results, as shown in Fig. 8(a);

Step 2: according to the given size (such as 512×512), the original UAV image is divided into overlap slices, which will be proportionally enlarged. For example, as shown in Fig. 8(b), the original UAV image is divided into overlap slices P_1, P_2, P_3, and P_4. The enlarged results of slices P_2 and P_4 by 2 times are $P_2{}'$ and $P_4{}'$;

Step 3: use the model trained by the proposed method to perform inference on the enlarged slices (such as $P_2{}'$ and $P_4{}'$ in Fig. 8(b)), and obtain the inference results [such as $P_2{}''$ and $P_4{}''$ in Fig. 8(b)];

Step 4: the original UAV image inference results in Step 1 [see Fig. 8(a)] and the sliced image inference results in Step 3 [see Fig. 8(b)] are fused through NMS to obtain the final inference results, which are shown in Fig. 8(c).

IV. Experiments and Results

A. Datasets and Pretrained Models

1) Datasets: We selected the popular public dataset VisDrone2021-DET and UAVDT to train, test, and evaluate the proposed model.

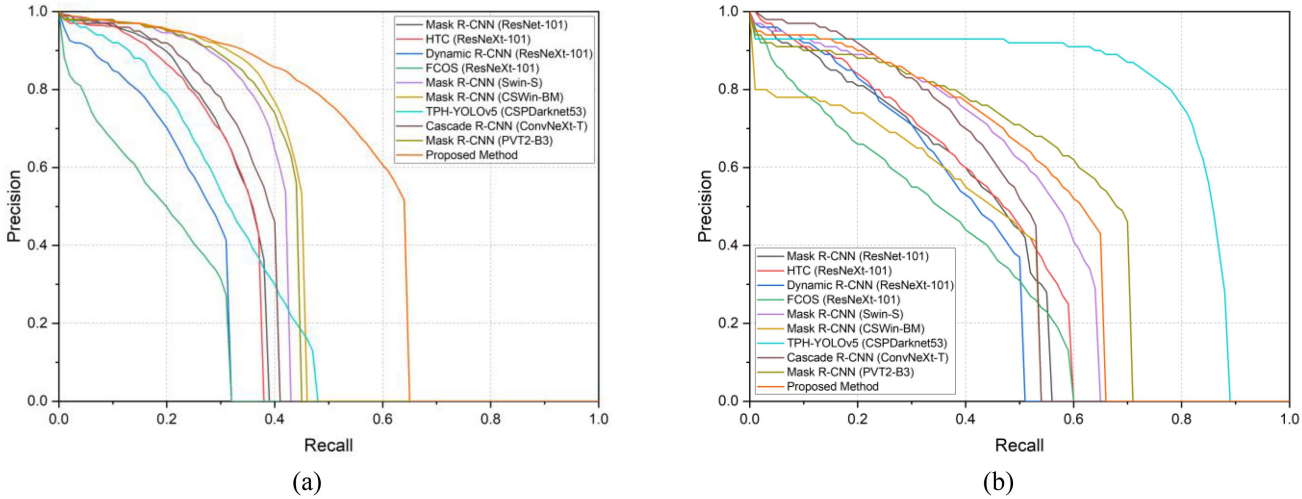


Fig. 9. PR curve (IoU = 0.5) of different methods without pretrained models. on the visdrone2021-DET dataset. (a) PR curve on the visdrone2021-DET dataset. (b) PR curve on the sparse UAVDT dataset.

TABLE I
STATISTICS DATA OF ALL OBJECT BOUNDING BOXES IN VISDRONE2021-DET TRAIN AND VAL DATASET

Size	$< 200^2$	$200^2 \sim 400^2$	$> 400^2$	$< 32^2$	$32^2 \sim 64^2$	$> 96^2$
number	487887	2035	42	306262	159999	23703

The VisDrone2021-DET [78] dataset is a mainstream UAV image dataset, which is used for object detection and has a total of 11 categories including pedestrian, people, bicycle, car, van, truck, tricycle, awning-tricycle, bus, motor, and others. The numbers of images in the training, test, and evaluation sets are 6471, 548, and 1610, respectively. In the VisDrone2021-DET dataset, the shooting angle of the UAV significantly changes, the scale of various objects is diverse, and most sizes of the objects are less than 32 pixels. Taking train and test data as examples for analysis, the results are shown in Table I [79], and the first row represents the object size range and the second row contains the number of corresponding objects. In addition, the object categories present in the VisDrone2021-DET dataset are fine-grained and classified, which makes it very challenging in the field of UAV image object detection task. Simultaneously, the number of images in the VisDrone2021-DET dataset is moderate, which can better test the performance of various models. In addition, the data size of the VisDrone2021-DET dataset is resized to an integral multiple of 224.

The UAVDT [80] dataset is a UAV benchmark captured by a UAV platform at a number of locations in urban areas; it represents various common scenes including squares, arterial streets, toll stations, highways, crossings, and T-junctions, and includes about 80 000 representative frames from 100 video sequences. UAVDT dataset has a total of 3 categories: car, truck, and bus. Because the UAVDT dataset is taken from video sequences, the images in the same video sequence contain a lot of similar content, resulting in redundancy, which makes it challenging to distinguish the performance of different models. Therefore, in this study, a sparse UAVDT dataset was constructed based on the original UAVDT dataset by extracting one in every 20 images. In the sparse UAVDT dataset, the number of images in the training, test, and evaluation sets were 1442, 206, and 412, respectively. In addition, the data size of the sparse UAVDT dataset is scaled to 448×448.

2) Pretrained Models: In addition various deep learning models to directly train on the given dataset, following-up the existing SOTA networks can better improve the performance of the proposed model. When comparing related methods, some pretrained results given by the existing SOTA networks will also be used in the experiments. These pretrained models are trained on large object detection datasets such as Pascal VOC, COCO, and FAIR1M. Although these models cannot be directly used in UAV image object detection, they can help with weight initialization.

B. Experimental Implementation Details

1) Evaluation Metrics: We used precision, a common evaluation metric that measures the percentage of the correct prediction results, as the basic experimental evaluation criteria and obtained the precision results of the bounding boxes, including average precision (AP), AP50 (AP test results when the IoU threshold is greater than 0.5), AP75 (AP test results when the IoU threshold is greater than 0.75), APS (AP test results with object frame size less than 32×32 pixels), APM (AP test results with object frame sizes between 32×32 pixels and 96×96 pixels), and APL (AP test results with object frame sizes larger than 96×96 pixels). AP is usually computed for each class separately, therefore, the mean AP, which is the average of

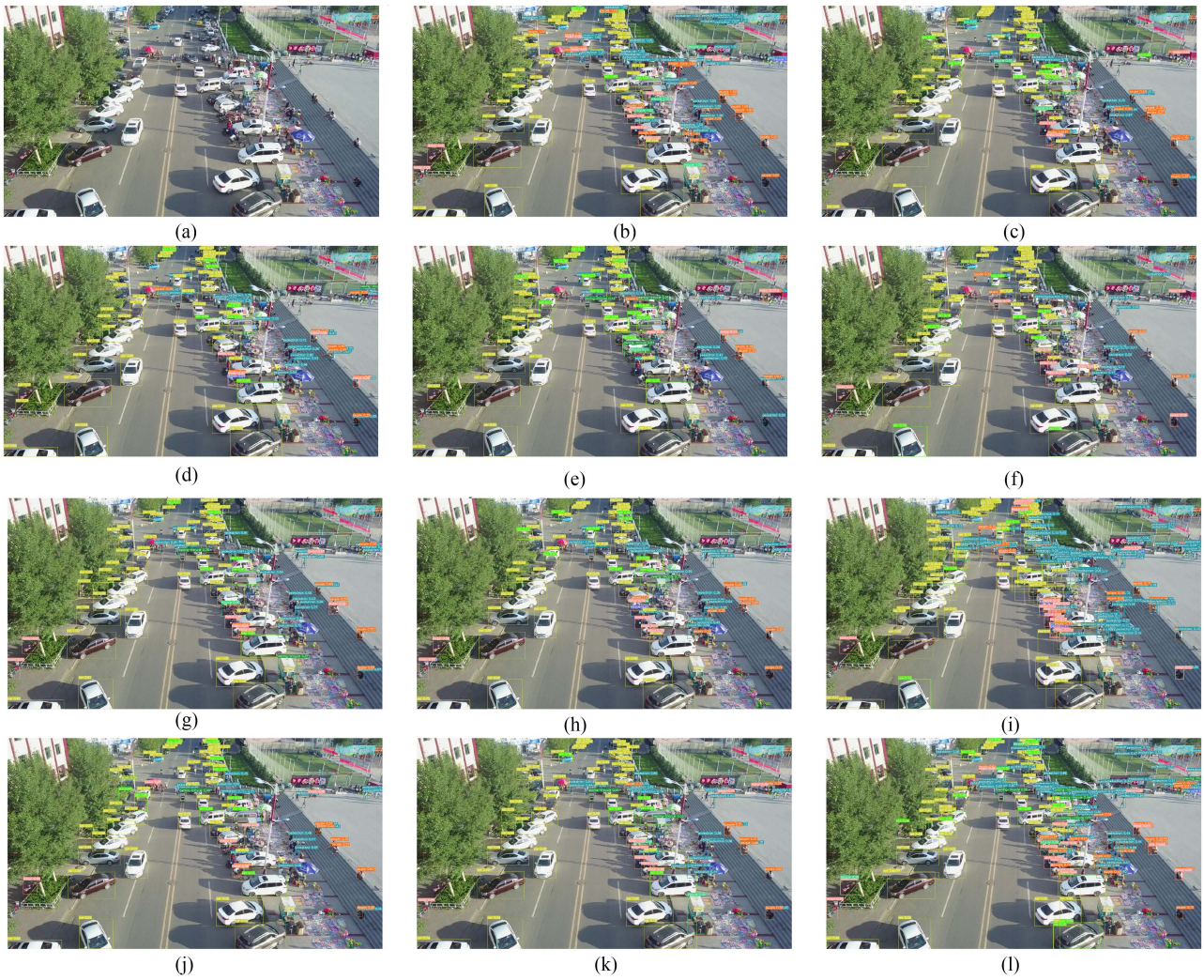


Fig. 10. Comparison of the qualitative inference results of the different methods without pretrained models in the first scenario from the visdrone2021-DET dataset. (a) Original image. (b) Ground truth. (c) Mask R-CNN (Resnet-101). (d) HTC (ResNeXt-101). (e) Dynamic R-CNN (ResNeXt-101). (f) FCOS (ResNeXt-101). (g) Mask R-CNN (Swin-S). (h) Mask R-CNN (CSWin-BM). (i) TPH-YOLOv5 (CSPDarknet53). (j) Cascade R-CNN (ConvNeXt-T). (k) Mask R-CNN (PVT2-B3). (l) Proposed method.

the AP values of all object categories, was adopted as the final measure for evaluating overall accuracy.

2) Experimental Setup: All experiments in this article were carried out on a workbench equipped with an Intel CPU i7 9700k and one NVIDIA GeForce RTX 3090 (24G). The operating system was Ubuntu 20.04 LTS, and all deep learning models were constructed based on the open-source object detection toolbox MMDetection and PyTorch framework.

3) Comparative Methods: For a fair comparison, we only considered methods providing source codes and selected some classic and advanced networks, such as Mask R-CNN, HTC, FCOS, Dynamic R-CNN, TPH-YOLOv5, and Cascade R-CNN, and the backbone networks mainly included ResNet-101 [81], ResNeXt-101 [82], ConvNeXt-T [83], CSPDarknet53 [32], Swin-S [64], and PVT2-B3 [67]. For CSWin Transformer, CSWin-BM was constructed based on CSWin-B [65] by changing the stages to 2, 2, 18, and 2 blocks. In addition, the performance of each backbone network with the corresponding pretrained models was also presented.

4) Data Processing: For the training and testing datasets, we applied a series of standard data augmentation strategies. Horizontal random flip with a flipping probability of 0.5 was used to meet the requirements of image size for data processing; the height and width of the original input data were expanded to integer multiples of 224; band normalization operations using regularization with a mean of [123.675, 116.28, 103.53] and standard deviation of [58.395, 57.12, 57.375] was used.

5) Experimental Hyperparameter: We set the same training hyperparameters to maintain the consistency and comparability of the training results. For the experiments, we used the adaptive moment estimation with decoupled weight decay (AdamW) optimizer, where the initial learning rate and the weight decay were 0.0001 and 0.05, respectively. Considering the hardware limitations, the batch size was set to 1, and except for the

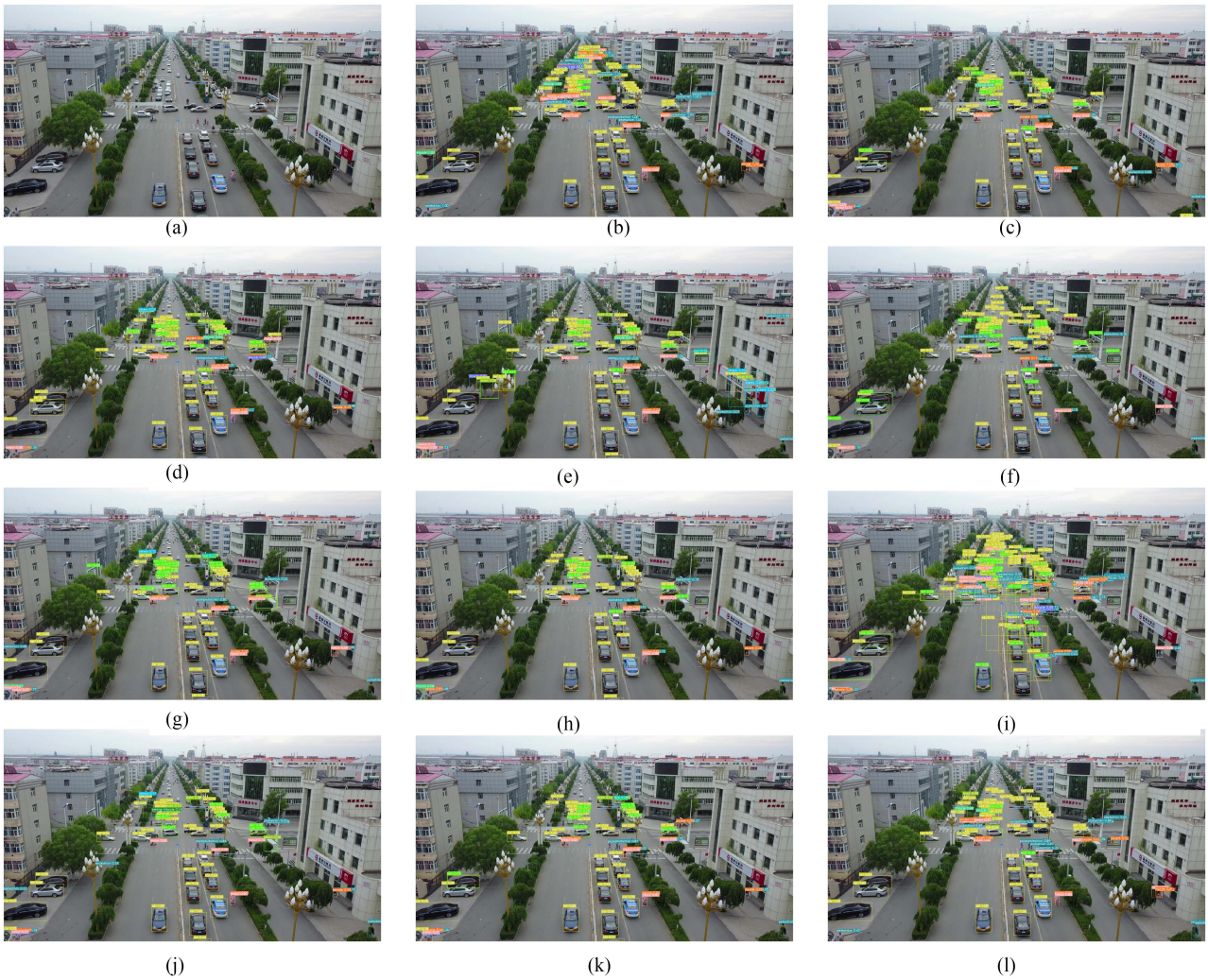


Fig. 11. Comparison of the qualitative inference results of the different methods without pretrained models in the second scenario from the visdrone2021-DET dataset. (a) Original image. (b) Ground truth. (c) Mask R-CNN (Resnet-101). (d) HTC (ResNeXt-101). (e) Dynamic R-CNN (ResNeXt-101). (f) FCOS (ResNeXt-101). (g) Mask R-CNN (Swin-S). (h) Mask R-CNN (CSWin-BM). (i) TPH-YOLOv5 (CSPDarknet53). (j) Cascade R-CNN (ConvNeXt-T). (k) Mask R-CNN (PVT2-B3). (l) Proposed method.

vanilla Mask R-CNN with ResNet-101 whose schedule is set to 200 epochs and Dynamic R-CNN with ResNeXt-101 whose schedule is set to 50 epochs, the schedule of all other methods was set to 3×, which means the entire training processing had 36 epochs.

C. Ablation Study

In this section, an ablation study was designed to analyze how much each component contributed to the accuracy of the overall performance on different datasets, including without (w/o) and with (w) pretrained models, as shown in Tables II and III. Mask R-CNN was set as the baseline method, and ResNet-101 served as the backbone network of vanilla Mask R-CNN.

Table II shows that Mask R-CNN with the CSWin Transformer as the backbone network achieved APs of 24.0 and 25.4 without and with the pretrained model, respectively, and the results of APs in Table III are 12.3 and 26.0 without and with the pretrained model, respectively, which are better than those of vanilla Mask R-CNN with ResNet-101. In addition, the detection performance showed improvements of 0.6 and 0.4 AP in Table II and 1.3 and 0.2 AP in Table III when the HPEM was added. Using SI, the performance of the proposed method can be further improved, especially in terms of APS, APM, and APL. The ablation study results proved that the various components of the proposed method were effective at improving the object detection accuracy in UAV images. The comparison shows that the performance of the proposed method with pretrained models is much better than that without pretrained models, which proves the contribution of the pretrained models.

To further verify the effectiveness of HPEM, experiments on the performance of different backbones with HPEM were conducted, and the results are shown in Table IV. Notably, to obtain more reliable comparison results, the pretrained models

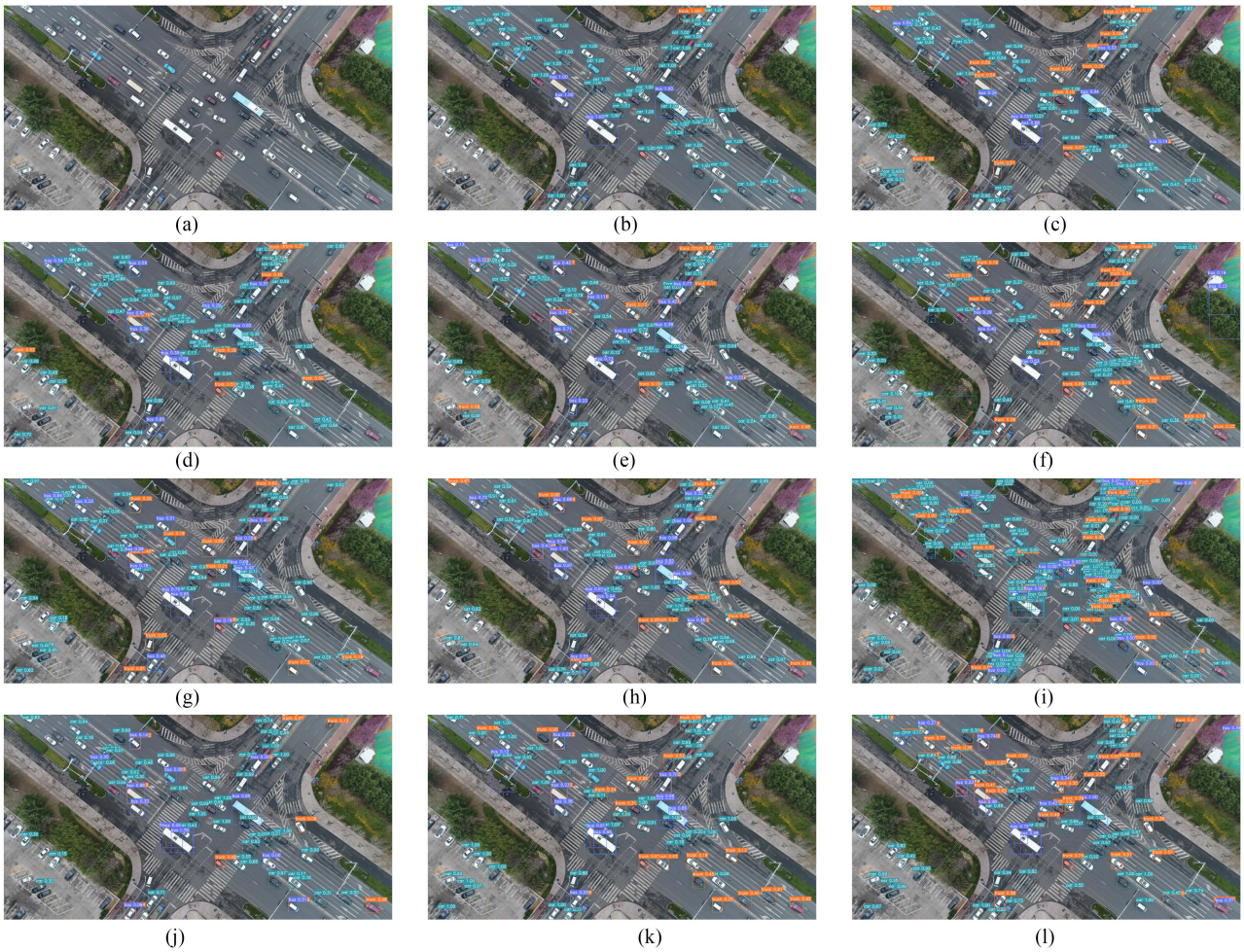


Fig. 12. Comparison of the qualitative inference results of the different methods without pretrained models in the third scenario from the sparse UAVDT dataset. (a) Original image. (b) Ground truth. (c) Mask R-CNN (Resnet-101). (d) HTC (ResNeXt-101). (e) Dynamic R-CNN (ResNeXt-101). (f) FCOS (ResNeXt-101). (g) Mask R-CNN (Swin-S). (h) Mask R-CNN (CSWin-BM). (i) TPH-YOLOv5 (CSPDarknet53). (j) Cascade R-CNN (ConvNeXt-T). (k) Mask R-CNN (PVT2-B3). (l) Proposed method.

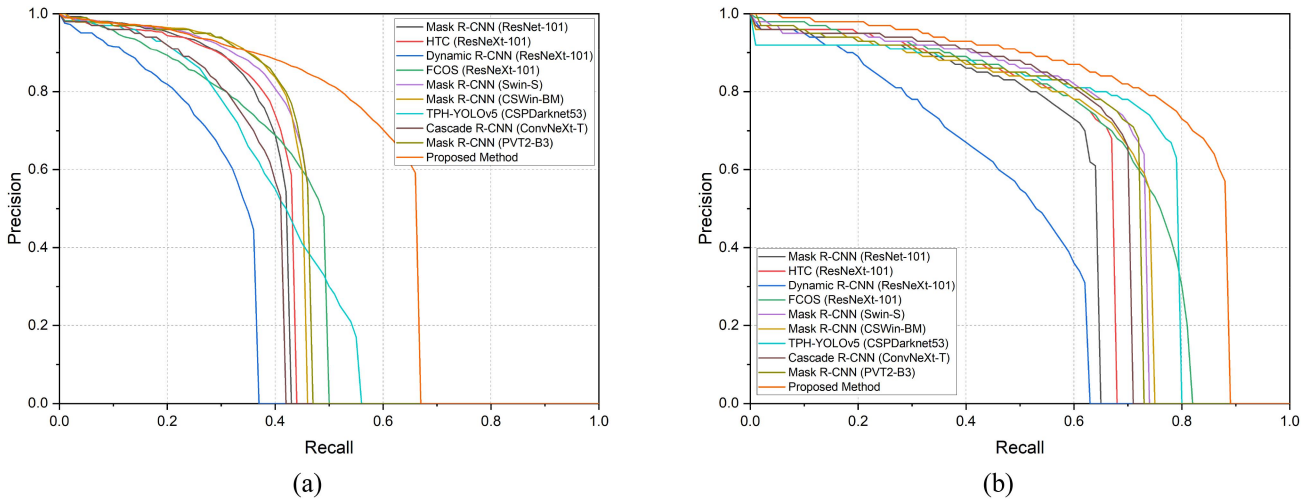


Fig. 13. PR curve (IoU=0.5) of different methods with pretrained models. on the visdrone2021-DET dataset. (a) PR curve on the visdrone2021-DET dataset. (b) PR curve on the sparse UAVDT dataset.

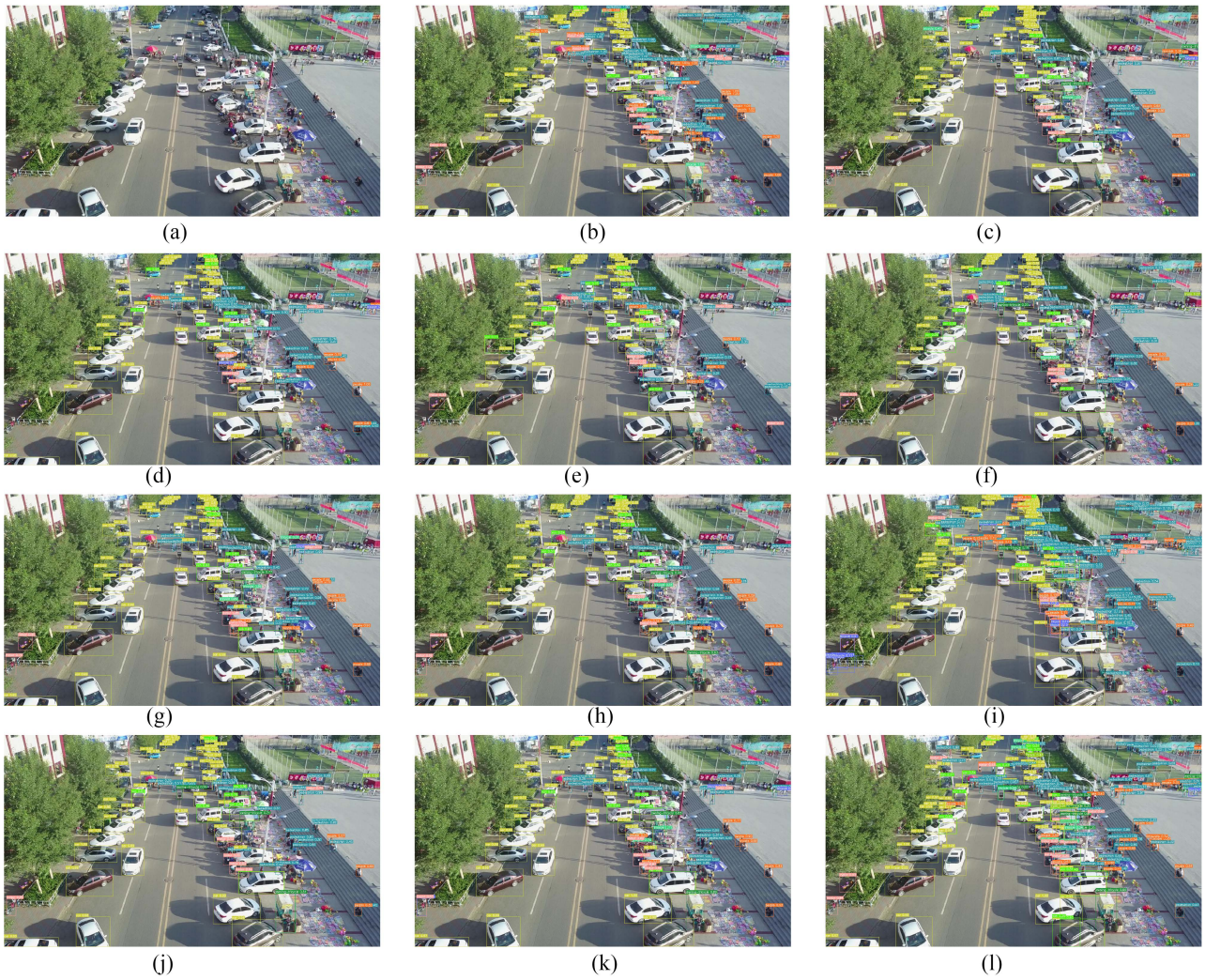


Fig. 14. Comparison of the qualitative inference results of the different methods with pretrained models in the first scenario from the visdrone2021-DET dataset. (a) Original image. (b) Ground truth. (c) Mask R-CNN (Resnet-101). (d) HTC (ResNeXt-101). (e) Dynamic R-CNN (ResNeXt-101). (f) FCOS (ResNeXt-101). (g) Mask R-CNN (Swin-S). (h) Mask R-CNN (CSWin-BM). (i) TPH-YOLOv5 (CSPDarknet53). (j) Cascade R-CNN (ConvNeXt-T). (k) Mask R-CNN (PVT2-B3). (l) Proposed method.

TABLE II
ABLATION STUDY RESULTS OF THE PROPOSED METHOD ON THE VISDRONE2021-DET DATASET

Method	CSWin-BM	HPEM	SI	Params(M)	AP	AP50	AP75	APS	APM	APL
Mask R-CNN (w/o weighted)				60.17	18.0	30.3	18.8	11.2	26.7	31.9
	√			68.27	24.0	39.2	25.4	15.5	36.0	38.4
	√	√		68.54	24.6	40.0	26.6	16.3	35.8	44.4
	√	√	√	68.54	**25.8**	**43.6**	**26.7**	**36.1**	**54.7**	**58.2**
Mask R-CNN (w weighted)				60.17	24.1	39.1	25.8	15.2	35.7	42.6
	√			68.27	25.4	41.1	27.3	16.9	37.5	42.2
	√	√		68.54	25.8	41.5	**27.7**	17.1	37.6	45.1
	√	√	√	68.54	**26.5**	**45.2**	27.1	**36.9**	**58.0**	**58.0**

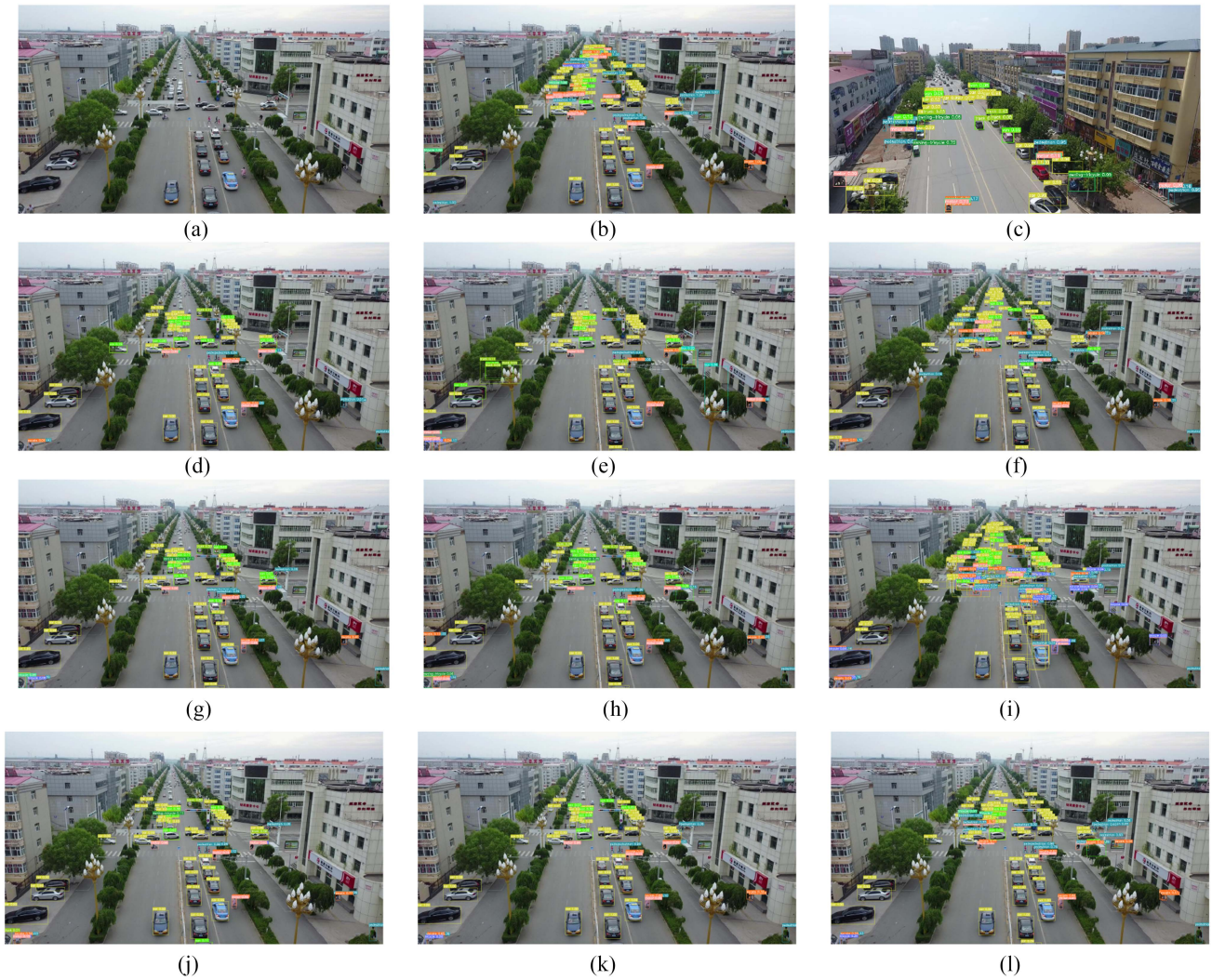


Fig. 15. Comparison of the qualitative inference results of the different methods with pretrained models in the second scenario from the visdrone2021-DET dataset. (a) Original image. (b) Ground truth. (c) Mask R-CNN (Resnet-101). (d) HTC (ResNeXt-101). (e) Dynamic R-CNN (ResNeXt-101). (f) FCOS (ResNeXt-101). (g) Mask R-CNN (Swin-S). (h) Mask R-CNN (CSWin-BM). (i) TPH-YOLOv5 (CSPDarknet53). (j) Cascade R-CNN (ConvNeXt-T). (k) Mask R-CNN (PVT2-B3). (l) Proposed method.

TABLE III
ABLATION STUDY RESULTS OF THE PROPOSED METHOD ON THE SPARSE UAVDT DATASET

Method	CSWin-BM	HPEM	SI	Params(M)	AP	AP50	AP75	APS	APM	APL
Mask R-CNN (w/o weighted)				60.17	8.4	19.4	6.2	5.5	13.4	7.5
	√			68.27	12.3	29.2	8.2	6.3	17.9	12.7
	√	√		68.54	13.6	32.7	8.8	7.8	19.8	20.6
	√	√	√	68.54	**14.4**	**33.1**	**11.0**	**23.6**	**42.1**	**46.3**
Mask R-CNN (w weighted)				60.17	21.1	40.0	20.1	13.1	29.3	16.1
	√			68.27	26.0	56.8	20.2	17.5	37.1	20.6
	√	√		68.54	26.2	**59.7**	20.0	19.8	36.0	32.0
	√	√	√	68.54	**26.3**	57.1	**20.4**	**44.9**	**65.5**	**61.2**

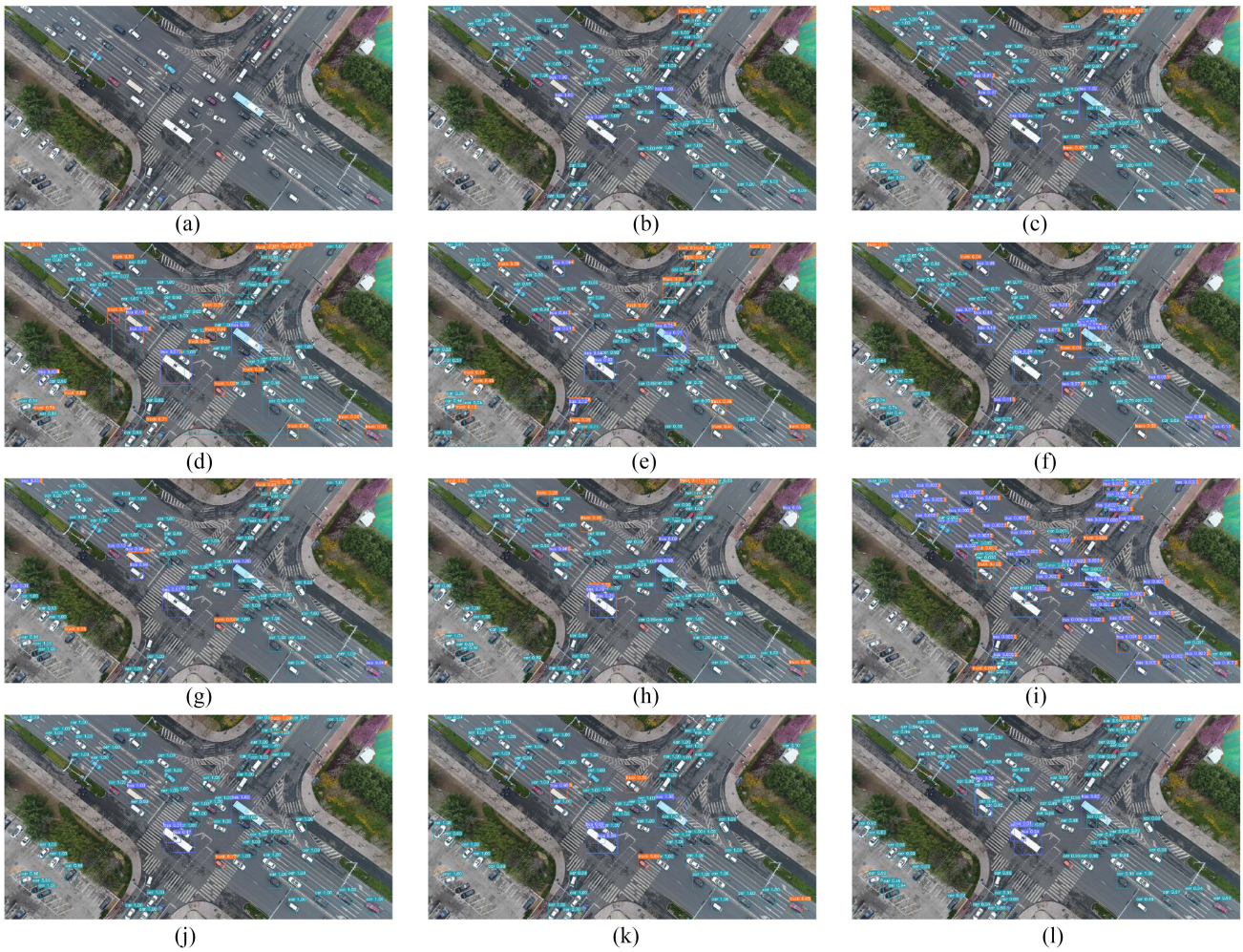


Fig. 16. Comparison of the qualitative inference results of the different methods with pretrained models in the second scenario from the sparse UAVDT dataset. (a) Original image. (b) Ground truth. (c) Mask R-CNN (Resnet-101). (d) HTC (ResNeXt-101). (e) Dynamic R-CNN (ResNeXt-101). (f) FCOS (ResNeXt-101). (g) Mask R-CNN (Swin-S). (h) Mask R-CNN (CSWin-BM). (i) TPH-YOLOv5 (CSPDarknet53). (j) Cascade R-CNN (ConvNeXt-T). (k) Mask R-CNN (PVT2-B3). (l) Proposed method.

were not used. The comparison results indicate that although HPEM is not universal, it can effectively improve performance of the proposed methods.

D. Evaluation and Comparisons

Based on the above datasets, pretrained models, and experimental implementation details, we performed an evaluation and comparisons from two aspects: without and with pretrained models.

1) Without Pretrained Models: The comparison results between different methods combined with the corresponding backbones on the Visdrone2021-DET dataset are shown in Table V. The comparison results show that Mask R-CNN with transformer as the backbone network has better performance than methods such as vanilla Mask R-CNN, HTC, FCOS, and Dynamic R-CNN with a similar count of parameters. Between the different transformer backbone networks, the method with CSWin Transformer as the backbone network achieves better performance, and the experimental result is 2.7 AP higher than the Swin Transformer. The proposed method proposed showed a further improvement in object detection accuracy, which was higher than those of vanilla Mask R-CNN (200 epochs) by 7.8 AP and Mask R-CNN with CSWin Transformer by 1.8 AP. Notably, the method proposed achieves a conspicuous improvement for small, medium, and large object detection compared to the Mask R-CNN with CSWin Transformer as the backbone network by 20.6, 18.7, and 19.8 AP, respectively.

The comparison results between different methods combined with the corresponding backbones on the sparse UAVDT dataset are shown in Table VI. The proposed method has a further improvement in object detection accuracy, which is higher than that of vanilla Mask R-CNN (200 epochs) by 6.0 AP and that of Mask R-CNN with CSWin Transformer by 2.1 AP. Notably, the proposed method achieves a conspicuous improvement for small, medium, and large object detection compared to the Mask R-CNN with CSWin Transformer as the backbone network by 17.3, 24.2, and 33.6 AP, respectively.

The precision-recall curves of the different methods without pretrained models on the Visdrone2021-DET and sparse UAVDT dataset are provided in Fig. 9, which intuitively show the detailed relationship between precision and recall. The proposed

TABLE IV
ABLATION STUDY RESULTS OF THE EFFECTIVENESS OF HPEM ON THE VISDRONE2021-DET DATASET AND SPARSE UAVDT DATASET

Dataset	Backbone	HPEM	Params(M)	AP	AP50	AP75	APS	APM	APL
Visdrone2021-DET	Swin-S		66.12	21.3	35.7	22.6	14.1	31.3	34.6
		√	66.53	18.5	31.8	19.3	12.6	25.9	28.8
	PVT2-B3		61.90	23.8	38.9	25.2	15.3	35.2	40.2
		√	65.52	22.2	36.7	23.4	15.1	31.9	35.7
	CSWin-BM		68.27	24.0	39.2	25.4	15.5	36.0	38.4
		√	68.54	24.6	40.0	26.6	16.3	35.8	44.4
Sparse UAVDT	Swin-S		66.12	11.2	26.8	7.4	7.9	16.8	8.6
		√	66.53	11.7	27.3	7.9	7.7	16.9	7.5
	PVT2-B3		61.90	13.9	29.2	11.5	8.9	19.7	17.1
		√	65.52	15.9	33.3	13.4	13.1	22.2	12.2
	CSWin-BM		68.27	12.3	29.2	8.2	6.3	17.9	12.7
		√	68.54	13.6	32.7	8.8	7.8	19.8	20.6

Mask R-CNN was set as the baseline method, and all backbones were initialized without pretrained models.

TABLE V
OBJECT DETECTION PERFORMANCE OF DIFFERENT METHODS WITH PRETRAINED MODELS ON THE VISDRONE2021-DET DATASET

Method	Epoch	Params(M)	AP	AP50	AP75	APS	APM	APL
Mask R-CNN (ResNet-101)	200	60.17	18.0	30.3	18.8	11.2	26.7	31.9
HTC (ResNeXt-101)	36	98.90	16.9	29.0	17.3	10.2	25.4	28.8
Dynamic R-CNN (ResNeXt-101)	50	60.16	14.7	23.6	15.5	8.5	21.6	30.4
FCOS (ResNeXt-101)	36	89.63	8.9	15.5	9.1	4.4	13.5	20.2
Mask R-CNN (Swin-S)	36	66.12	21.3	35.7	22.6	14.1	31.3	34.6
Mask R-CNN (CSWin-BM)	36	68.27	24.0	39.2	25.4	15.5	36.0	38.4
TPH-YOLOv5 (CSPDarknet53)	50	76.18	9.0	19.2	8.5	6.0	14.6	14.8
Cascade R-CNN (ConvNeXt-T)	36	76.94	23.4	36.3	25.6	15.4	33.6	33.4
Mask R-CNN (PVT2-B3)	36	61.90	23.8	38.9	25.2	15.3	35.2	40.2
Proposed Method	36	68.54	**25.8**	**43.6**	**26.7**	**36.1**	**54.7**	**58.2**

Parentheses indicate the backbone network (without pretrained models).

method exhibits better performance on the Visdrone2021-DET dataset in Fig. 9(a). However, the performance exhibited in Fig. 9(b) is not particularly ideal, which may because of the larger amount of training data needed by the transformer.

In addition to the quantitative comparisons, the qualitative inference results by different methods of the three scenarios selected from the Visdrone2021-DET and sparse UAVDT dataset are shown in Figs. 10–12, respectively, which show that the proposed method exhibits a conspicuous detection capability on the Visdrone2021-DET and sparse UAVDT dataset. In particular, it pays more attention to distant small objects and can achieve better detection performance for small objects.

2) With Pretrained Models: For downstream tasks, pretrained models on large-scale image datasets, such as ImageNet [84], and COCO 2017 [85], are usually used as the initial

TABLE VI
OBJECT DETECTION PERFORMANCE OF DIFFERENT METHODS WITH PRETRAINED MODELS ON THE SPARSE UAVDT DATASET

Method	Epoch	Params(M)	AP	AP50	AP75	APS	APM	APL
Mask R-CNN (ResNet-101)	200	60.17	8.4	19.4	6.2	5.5	13.4	7.5
HTC (ResNeXt-101)	36	98.90	8.0	18.9	5.3	5.4	13.4	6.9
Dynamic R-CNN (ResNeXt-101)	50	60.16	7.7	16.5	6.2	5.0	13.2	2.9
FCOS (ResNeXt-101)	36	89.63	6.6	17.1	4.2	3.9	9.1	8.2
Mask R-CNN (Swin-S)	36	66.12	11.2	26.8	7.4	7.9	16.8	8.6
Mask R-CNN (CSWin-BM)	36	68.27	12.3	29.2	8.2	6.3	17.9	12.7
TPH-YOLOv5 (CSPDarknet53)	50	76.18	14.3	28.2	**13.3**	10.5	19.7	27.6
Cascade R-CNN (ConvNeXt-T)	36	76.94	13.6	29.4	9.2	7.1	21.0	7.4
Mask R-CNN (PVT2-B3)	36	61.90	13.9	29.2	11.5	8.9	19.7	17.1
Proposed Method	36	68.54	**14.4**	**33.1**	11.0	**23.6**	**42.1**	**46.3**

Parentheses indicate the backbone network (without pretrained models).

TABLE VII
OBJECT DETECTION PERFORMANCE OF DIFFERENT METHODS WITH PRETRAINED MODELS ON THE VISDRONE2021-DET DATASET

Method	Epoch	Params(M)	AP	AP50	AP75	APS	APM	APL
Mask R-CNN (ResNet-101)	200	60.17	24.4	39.5	26.3	15.4	36.1	42.0
HTC (ResNeXt-101)	36	98.90	22.9	37.4	24.7	14.0	34.6	38.1
Dynamic R-CNN (ResNeXt-101)	50	60.16	17.9	28.4	19.2	11.2	26.8	33.4
FCOS (ResNeXt-101)	36	89.63	19.2	30.7	20.1	11.4	28.7	34.9
Mask R-CNN (Swin-S)	36	66.12	24.6	41.0	25.8	15.3	36.8	44.2
Mask R-CNN (CSWin-BM)	36	68.27	25.4	41.1	**27.3**	16.9	37.5	42.2
TPH-YOLOv5 (CSPDarknet53)	50	76.18	17.1	30.5	16.6	10.1	25.6	30.8
Cascade R-CNN (ConvNeXt-T)	36	76.94	25.9	39.7	28.5	17.8	36.1	36.4
Mask R-CNN (PVT2-B3)	36	61.90	26.1	41.9	28.1	16.9	38.8	40.8
Proposed Method	36	68.54	**26.5**	**45.2**	27.1	**36.9**	**58.0**	**58.0**

Parentheses indicate the backbone network (with pretrained models).

weights of the backbone network. Therefore, in this study, we used these pretrained models to initialize the weights of the corresponding backbone networks in comparative models and conducted subsequent comparisons. The results on the Visdrone2021-DET dataset in Table IV show that with the pretrained models, the performance of each method is improved. The proposed method has a further improvement in object detection accuracy, which is higher than that of vanilla Mask R-CNN (200 epochs) by 2.1 AP and Mask R-CNN with CSWin Transformer by 1.1 AP. Furthermore, the proposed method achieved a conspicuous improvement in small, medium, and large objects detection compared to the Mask R-CNN with CSWin Transformer as the backbone network by 20.0, 20.5, and 15.8 AP, respectively.

The results on the sparse UAVDT dataset in Table VIII show that with pretrained models, the performance of each method is improved. The proposed method led to a further improvement in object detection accuracy, which is higher than that of vanilla Mask R-CNN (200 epochs) by 5.2 AP and Mask R-CNN with CSWin Transformer by 0.3 AP, and the model achieved a

TABLE VIII
OBJECT DETECTION PERFORMANCE OF DIFFERENT METHODS WITH PRETRAINED MODELS ON THE SPARSE UAVDT DATASET

Method	Epoch	Params(M)	AP	AP50	AP75	APS	APM	APL
Mask R-CNN (ResNet-101)	200	60.17	21.1	40.0	20.1	13.1	29.3	16.1
HTC (ResNeXt-101)	36	98.90	22.5	42.5	22.7	11.9	34.1	18.1
Dynamic R-CNN (ResNeXt-101)	50	60.16	10.4	20.2	9.5	6.6	15.5	7.1
FCOS (ResNeXt-101)	36	89.63	23.3	45.1	**23.4**	17.0	33.8	18.4
Mask R-CNN (Swin-S)	36	66.12	25.7	55.8	20.0	15.1	38.5	21.2
Mask R-CNN (CSWin-BM)	36	68.27	26.0	56.8	20.2	17.5	37.1	20.6
TPH-YOLOv5 (CSPDarknet53)	50	76.18	18.8	31.9	19.9	14.8	23.3	27.9
Cascade R-CNN (ConvNeXt-T)	36	76.94	22.6	52.4	14.0	11.5	34.5	29.9
Mask R-CNN (PVT2-B3)	36	61.90	25.3	49.9	23.3	17.5	36.5	23.9
Proposed Method	36	68.54	**26.3**	**57.1**	20.4	**44.9**	**65.5**	**61.2**

Parentheses indicate the backbone network (with pretrained models).

conspicuous improvement for small, medium, and large object detection compared to the Mask R-CNN with CSWin Transformer as the backbone network by 27.4, 28.4, and 40.6 AP, respectively.

The precision-recall curves of the different methods with pretrained models on the Visdrone2021-DET and sparse UAVDT dataset are provided in Fig. 13, and the proposed method still exhibits superior performance. The comparison between the PR curve in Figs. 9 and 13 indicate that an adequate pretraining model can further improve the performance of the transformer.

The qualitative inference results by different methods of the three scenarios selected from the Visdrone2021-DET and sparse UAVDT dataset with pretrained models are shown in Figs. 14–16. The proposed method exhibited a conspicuous detection capability on both the Visdrone2021-DET and sparse UAVDT datasets.

V. DISCUSSION AND CONCLUSION

The experimental results in Tables V–VIII and the qualitative inference results in Figs. 10–16 show that the CNN-transformer hybrid model proposed in this study can achieve better results compared with the current classic and popular models. By constructing a feature pyramid structure and using CSWin Transformer as the backbone, the proposed method can obtain better high-level semantic features of different scales while effectively establishing long-distance dependencies, which is helpful to achieve multiscale object detection, and the ablation study results in Tables II and III indicate that the AP test results of small, middle and large objects have a significant improvement. Combined with HPEM, low-dimensional information such as edges and corners in the image can be further extracted and utilized to enhance and enrich the feature information. In addition, according to the high-resolution characteristics of UAV images, SI is used to fuse the normal and SI results, which can improve the accuracy of object detection, especially that of detecting small objects, without modifying the original model, and the results in Tables II and III indicate that the AP test results of small objects are improved greatly, even doubled, with the SI method. The SI method indicates that the subsequent inference process also has a large impact on the overall performance. Furthermore, the comparison of the PR curves in Figs. 9 and 13 indicate that the transformer needs large amounts of training data or a better weight initialization to perform better.

However, deficiencies are also observed in experimental comparisons. Although pretrained models can effectively improve the performance of networks, the contributions of various pretrained models are considerably different. In addition, the scenarios of pretrained models are considerably different from those of the training dataset, which leads to inadequate use of the pretrained models. The training time required by the proposed method is low, but each epoch is time-consuming, and the training process has a high hardware requirement. In addition, this study verifies the effectiveness of the CNN-transformer hybrid model and the proposed module (such as the HEPM and SI), and there is still a certain gap from SOTA methods. Meanwhile, notably, in the comparison results, although some methods have demonstrated excellent performance, ideal results under the training dataset in this study are not achieved with the official codes, which is a problem that requires further in-depth analysis.

There is still room for improving the proposed method, and follow-up work can be carried out around model transfer, lightweight models, and performance improvement, which can likely be solved by transfer learning and the use of efficient transformers.

REFERENCES

[1] J. Bo, R. Qu, Y. Li, and C. Li, "Object detection in UAV imagery based on deep learning: Review," *Acta Aeronauticaet Astronautica Sinica*, vol. 42, no. 4, Aug. 2021, Art. no. 524519.

[2] K. Nguyen et al., "The state of aerial surveillance: A survey," 2022, *arXiv:2201.03080v2*.

[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] G. Mao, T. Deng, and N. Yu, "Object detection in UAV images based on multi-scale split attention," *Acta Aeronautica et Astronautica Sinica*, vol. 43, pp. 1–12, doi: 10.7527/S1000-6893.2021.26738. [Online]. Available: http://hkxb.buaa.edu.cn/CN/10.7527/S1000-6893.2021.26738

[5] 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.*, 2013, pp. 580–587.

[6] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis* ., 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," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit* ., 2017, pp. 2980–2988.

[9] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6154–6162.

[10] H. Zhang, H. Chang, B. Ma, N. Wang, and X. Chen, "Dynamic R-CNN: Towards high quality object detection via dynamic training," in *Proc. Eur. Conf. Comput. Vis.*, Aug. 2020, pp. 260–275.

[11] S. Wang, "Research towards YOLO-series algorithms: Comparison and analysis of object detection models for real-time UAV applications," in *Proc. 2nd Int. Conf. Internet Things, Artif. Intell. Mech. Autom.*, May 2021, Paper 012021.

[12] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. 14th Eur. Conf.*, Oct. 2015, pp. 21–37.

[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

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

[15] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, Dec. 2017, pp. 6000–6010.

[16] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. 9th Int. Conf. Learn. Representations*, May 2021.

[17] Y. Bazi, L. Bashmal, M. M. A. Rahhal, R. A. Dayil, and N. A. Ajlan, "Vision transformers for remote sensing image classification," *Remote Sens.*, vol. 13, no. 3, Feb. 2021, Art. no. 516.

[18] L. Bashmal, Y. Bazi, M. M. Al Rahhal, H. Alhichri, and N. Al Ajlan, "UAV image multi-labeling with data-efficient transformers," *Appl. Sci.*, vol. 11, no. 9, 2021, Art. no. 3974.

[19] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "ImageNet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," in *Proc. 7th Int. Conf. Learn. Representations*, May 2019.

[20] Z. Li, G. Chen, and T. Zhang, "A CNN-transformer hybrid approach for crop classification using multitemporal multisensor images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 847–858, 2020.

[21] W. F. Hendria, Q. T. Phan, F. Adzaka, and C. Jeong, "Combining transformer and CNN for object detection in UAV imagery," *ICT Express*, 2022, doi: 10.1016/j.icte.2021.12.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959521001715

[22] C. Zhang, W. Jiang, Y. Zhang, W. Wang, Q. Zhao, and C. Wang, "Transformer and CNN hybrid deep neural network for semantic segmentation of very-high-resolution remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4408820.

[23] S. Deng et al., "A global-local self-adaptive network for drone-view object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 1556–1569, 2021.

[24] F. Yang, H. Fan, P. Chu, E. Blasch, and H. Ling, "Clustered object detection in aerial images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8310–8319.

[25] Y. Wang, Y. Yang, and X. Zhao, "Object detection using clustering algorithm adaptive searching regions in aerial images," in *Proc. Eur. Conf. Comput. Vis.*, Aug. 2020, pp. 651–664.

[26] Z. Wu, K. Suresh, P. Narayanan, H. Xu, H. Kwon, and Z. Wang, "Delving into robust object detection from unmanned aerial vehicles: A deep nuisance disentanglement approach," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1201–1210.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[28] K. Chen et al., "Hybrid task cascade for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4969–4978.

[29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[30] J. Redmon and A. Farhadi, "YOLO9000: Better faster stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6517–6525.

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

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

[33] P. Zhu, L. Wen, D. Du, X. Bian, Q. Hu, and H. Ling, "Vision meets drones: Past, present and future," 2020, *arXiv:2001.06303v3*.

[34] X. Wu, W. Li, D. Hong, R. Tao, and Q. Du, "Deep learning for unmanned aerial vehicle-based object detection and tracking: A survey," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 1, pp. 91–124, Mar. 2022.

[35] Z. Deng, H. Sun, S. Zhou, J. Zhao, L. Lei, and H. Zou, "Multi-scale object detection in remote sensing imagery with convolutional neural networks," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, no. Part A, pp. 3–22, Nov. 2018.

[36] W. Zhang, C. Liu, F. Chang, and Y. Song, "Multi-scale and occlusion aware network for vehicle detection and segmentation on UAV aerial images," *Remote Sens.*, vol. 12, no. 11, May 2020, Art. no. 1760.

[37] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 936–944.

[38] X. Zhang, K. Zhu, G. Chen, X. Tan, and Y. Gong, "Geospatial object detection on high resolution remote sensing imagery based on double multi-scale feature pyramid network," *Remote Sens.*, vol. 11, no. 7, Mar. 2019, Art. no. 755.

[39] W. Han et al., "Improving training instance quality in aerial image object detection with a sampling-balance-based multistage network," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 12, pp. 10575–10589, Dec. 2021.

[40] Y. Liu, Q. Li, Y. Yuan, Q. Du, and Q. Wang, "ABNet: Adaptive balanced network for multiscale object detection in remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5614914.

[41] C. Chen, W. Gong, Y. Chen, and W. Li, "Object detection in remote sensing images based on a scene-contextual feature pyramid network," *Remote Sens.*, vol. 11, no. 3, Feb. 2019, Art. no. 339.

[42] R. Zhang, Z. Shao, X. Huang, J. Wang, and D. Li, "Object detection in UAV images via global density fused convolutional network," *Remote Sens.*, vol. 12, no. 19, Sep. 2020, Art. no. 3140.

[43] R. Mottaghi et al., "The role of context for object detection and semantic segmentation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 891–898.

[44] X. Liang, J. Zhang, L. Zhuo, Y. Li, and Q. Tian, "Small object detection in unmanned aerial vehicle images using feature fusion and scaling-based single shot detector with spatial context analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1758–1770, Jun. 2020.

[45] C. Chen et al., "RRNet: A hybrid detector for object detection in drone-captured images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop*, 2019, pp. 100–108.

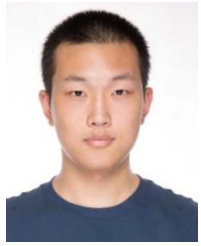
[46] X. Zhang, E. Izquierdo, and K. Chandramouli, "Dense and small object detection in UAV Vision based on cascade network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop*, 2019, pp. 118–126.

[47] Z. Liu, G. Gao, L. Sun, and Z. Fang, "HRDNet: High-resolution detection network for small objects," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2021, pp. 1–6.

[48] M. Liu, X. Wang, A. Zhou, X. Fu, and C. Piao, "UAV-YOLO: Small object detection on unmanned aerial vehicle perspective," *Sensors*, vol. 20, no. 8, Apr. 2020, Art. no. 2238.

[49] Y. Liu, F. Yang, and P. Hu, "Small-object detection in UAV-captured images via multi-branch parallel feature pyramid networks," *IEEE Access*, vol. 8, pp. 145740–145750, 2020.

[50] Y. Q. Cai et al., "Guided attention network for object detection and counting on drones," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 709–717.
[51] D. Du et al., "VisDrone-DET2020: The vision meets drone object detection in image challenge results," in *Proc. Eur. Conf. Comput. Vis.*, Aug. 2020, pp. 692–712.
[52] X. Hu et al., "SINet: A scale-insensitive convolutional neural network for fast vehicle detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 1010–1019, Mar. 2019.
[53] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.
[54] X. He, Y. Zhou, J. Zhao, D. Zhang, R. Yao, and Y. Xue, "Swin transformer embedding UNet for remote sensing image semantic segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4408715.
[55] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *Proc. 9th Int. Conf. Learn. Representations*, May 2021.
[56] D. Meng et al., "Conditional DETR for fast training convergence," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3631–3640.
[57] 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.*, 2022, pp. 13609–13617.
[58] S. Liu et al., "DAB-DETR: Dynamic anchor boxes are better queries for DETR," in *Proc. 10th Int. Conf. Learn. Representations*, Apr. 2022.
[59] H. Zhang et al., "DINO: DETR with improved denoising anchor boxes for end-to-end object detection," 2022, *arXiv:2203.03605v4*.
[60] H. Touvron, M. Cord, M. Douze, F. Massa, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 10347–10357.
[61] L. Bashmal, Y. Bazi, M. Rahhal, H. Alhichri, and N. A. Ajlan, "UAV image multi-labeling with data-efficient transformers," *Appl. Sci.*, vol. 11, no. 9, Apr. 2021, Art. no. 3974.
[62] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 12159–12168.
[63] W. Li et al., "SepViT: Separable vision transformer," 2022, *arXiv:2203.15380*.
[64] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9992–10002.
[65] X. Dong et al., "CSWin transformer: A general vision transformer backbone with cross-shaped windows," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12114–12124.
[66] W. Wang et al., "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis* ., 2021, pp. 548–558.
[67] W. Wang et al., "PVT v2: Improved baselines with pyramid vision transformer," *Comp. Visual Media*, vol. 8, pp. 415–424, 2022.
[68] S. Tang, J. Zhang, S. Zhu, and P. Tan, "Quadtree attention for vision transformers," in *Proc. 10th Int. Conf. Learn. Representations*, Apr. 2022.
[69] W. Yuan and W. Xu, "MSST-Net: A multi-scale adaptive network for building extraction from remote sensing images based on swin transformer," *Remote Sens.*, vol. 13, no. 23, Nov. 2021, Art. no. 4743.
[70] X. Xu et al., "An improved swin transformer-based model for remote sensing object detection and instance segmentation," *Remote Sens.*, vol. 13, no. 23, Nov. 2021, Art. no. 4779.
[71] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2021, pp. 2778–2788.
[72] F. Fan et al., "Efficient instance segmentation paradigm for interpreting SAR and optical images," *Remote Sens.*, vol. 14, no. 43, Jan. 2022, Art. no. 531.
[73] J. Feng and C. Yi, "Lightweight detection network for arbitrary-oriented vehicles in UAV imagery via global attentive relation and multi-path fusion," *Drones*, vol. 6, no. 5, Apr. 2022, Art. no. 108.
[74] Q. Li, Y. Chen, and Y. Zeng, "Transformer with transfer CNN for remote-sensing-image object detection," *Remote Sens.*, vol. 14, no. 4, Feb. 2022, Art. no. 984.
[75] Y. Zhang, X. Liu, S. Wa, S. Chen, and Q. Ma, "GANsformer: A detection network for aerial images with high performance combining convolutional network and transformer," *Remote Sens.*, vol. 14, no. 4, Feb. 2022, Art. no. 923.
[76] W. Maciej, P. Hyunsoo, M. Jacek, and K. Kyung-Joong, "Recent advances in general game playing," *Sci. World J.*, vol. 2015, Aug. 2015, Art. no. 986262.
[77] F. C. Akyon, S. O. Altinuc, and A. Temizel, "Slicing aided hyper inference and fine-tuning for small object detection," in *Proc. IEEE Int. Conf. Image Process.*, 2022, pp. 966–970.
[78] Y. Cao et al., "VisDrone-DET2021: The vision meets drone object detection challenge results," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2021, pp. 2847–2854.
[79] J. Wan, B. Zhang, Y. Zhao, Y. Du, and Z. Tong, "VistrongerDet: Stronger visual information for object detection in VisDrone images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2021, pp. 2820–2829.
[80] D. Du, Y. Qi, H. Yu, Y. Yang, and K. Duan, "The unmanned aerial vehicle benchmark: Object detection and tracking," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 375–391.
[81] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[82] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5987–5995.
[83] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11966–11976.
[84] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, pp. 211–252, 2014.
[85] T. Y. Lin, M. Maire, S. Belongie, J. Hays, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2014, pp. 740–755.



Wanjie Lu received the B.S. degree in photogrammetry and remote sensing and the Ph.D. degree in surveying and mapping from the Information Engineering University, Zhengzhou, China, in 2016 and 2020, respectively.

He is currently a Lecturer with the Data and Target Engineering Institute, Information Engineering University, Zhengzhou, China. His research interests include unmanned aerial vehicle remote sensing image processing, deep learning algorithm, and spatial information service.

Chaozhen Lan received the B.S. and M.S. degrees in photogrammetry and remote sensing and the Ph.D. degree in surveying and mapping from the Zhengzhou Institute of Surveying and Mapping, Zhengzhou, China, in 2002, 2005, and 2009, respectively.

He is currently an Associate Professor and a Master's Supervisor with the Information Engineering University, Zhengzhou, China. His research interests include photogrammetry and unmanned aerial vehicle remote sensing.

Chaoyang Niu received the B.S. and M.S. degrees in information engineering from Zhengzhou Information Technology Institute, Zhengzhou, China, in 2003 and 2006, respectively, and the Ph.D. degree in signal and information processing from Zhengzhou Institute of Surveying and Mapping, Zhengzhou, China, in 2011.

In 2016, he was an Associate Professor with the Data and Target Engineering Institute, Information Engineering University. His research interests include pattern recognition, unmanned aerial vehicle remote sensing, and optical and radar imagery processing.

Wei Liu received the B.S., M.S., and Ph.D. degrees in information and communication engineering from the Information Engineering University, Zhengzhou, China, in 2001, 2003, and 2016, respectively.

He is currently an Associate Professor with the Information Engineering University. His research interests include pattern recognition, remote sensing information processing, and deep learning.

Liang Lyu received the B.S. degree in measurement and control engineering and the M.S. degree in photogrammetry and remote sensing from the Zhengzhou Institute of Surveying and Mapping, Zhengzhou, China, in 2011 and 2014, respectively, and the Ph.D. degree in surveying and mapping from the Information Engineering University, Zhengzhou, China, in 2019.

He is currently an Associate Professor with the Information Engineering University. His research interests include photogrammetry, remote sensing, digital Earth information resources, space situational awareness, and visualization.

Qunshan Shi received the B.S. and M.S. degrees in photogrammetry and remote sensing from Zhengzhou Institute of Surveying and Mapping, Zhengzhou, China, in 2008 and 2011, respectively, and the Ph.D. degree in surveying and mapping from the Information Engineering University, Zhengzhou, China, in 2015.

He is currently an Associate Professor and a Master's Supervisor with the Information Engineering University. His research interests include photogrammetry, remote sensing, and virtual reality.

Shiju Wang received the B.S. degree in electronic science and technology and the M.S. degree in optical engineering from Zhengzhou University, Zhengzhou, China, in 2013 and 2016, respectively.

He is currently an Assistant Professor with the Information Engineering University, Zhengzhou, China. His research interests include remote sensing, image processing, and unmanned aerial vehicle technology.