Improving UAV Image Target Detection: A Novel Approach Using OptiDETR with Swin Transformer

Wenlong Ma, Weisheng Liu*

***Abstract*—In the analysis of drone aerial images, object detection tasks are particularly challenging, especially in the presence of complex terrain structures, extreme differences in target sizes, suboptimal shooting angles, and varying lighting conditions, all of which exacerbate the difficulty of recognition. In recent years, the DETR model based on the Transformer architecture has eliminated traditional post-processing steps such as NMS(Non-Maximum Suppression), thereby simplifying the object detection process and improving detection accuracy, which has garnered widespread attention in the academic community. However, DETR has limitations such as slow training convergence, difficulty in query optimization, and high computational costs, which hinder its application in practical fields. To address these issues, this paper proposes a new object detection model called OptiDETR. This model first employs a more efficient hybrid encoder to replace the traditional Transformer encoder. The new encoder significantly enhances feature processing capabilities through internal and cross-scale feature interaction and fusion logic. Secondly, an IoU (Intersection over Union) aware query selection mechanism is introduced. This mechanism adds IoU constraints during the training phase to provide higher-quality initial object queries for the decoder, significantly improving the decoding performance. Additionally, the OptiDETR model integrates SW-Block into the DETR decoder, leveraging the advantages of Swin Transformer in global context modeling and feature representation to further enhance the performance and efficiency of object detection. To tackle the problem of small object detection, this study innovatively employs the SAHI algorithm for data augmentation. Through a series of experiments, It achieved a significant performance improvement of more than two percentage points in the mAP (mean Average Precision) metric compared to current mainstream object detection models. Furthermore, there is a noticeable reduction in computation and memory consumption, demonstrating the excellent performance and practical value of OptiDETR in object detection tasks.**

***Index Terms*—Object detection, UAV Photography, Detection Transformer, SAHI**

I. Introduction

OBJECT detection in UAV imagery (Unmanned Aerial Vehicles) is a significant research direction in the field of computer vision, aiming to automatically identify and locate specific targets in images or videos captured by UAVs. This technology finds broad applications in various fields, such as military reconnaissance, environmental monitoring, agricultural management, urban planning, and disaster relief[1–3].

However, despite many researchers continually exploring new methods, achieving effective object detection still faces a series of challenges.

Firstly, aerial images typically cover large geographic areas and contain various complex ground structures and diverse backgrounds, increasing the difficulty of detecting the target [4]. Secondly, the targets in the images can vary in size, ranging from large buildings to small vehicles, and even pedestrians, all of which must be detected accurately [5]. Finally, constantly changing angles and lighting conditions in UAV-captured images place higher demands on the robustness of algorithms [6].

Traditional methods include feature extraction, classifier-based approaches, and sliding window methods. Feature extraction and classifier-based methods involve the use of SIFT [7] and SURF [8] combined with SVM (Support Vector Machine) classifiers [9]. The sliding window method progressively scans the entire image to locate targets, with region proposal methods like Selective Search used to generate potential target regions. In recent years, with the advancement of deep learning technology, the use of CNNs (Convolutional Neural Networks) such as Faster R-CNN [10], YOLO (You Only Look Once) [11], and SSD (Single Shot MultiBox Detector) [12] for end-to-end object detection has become mainstream.

Each of these methods has its advantages and disadvantages. Fast R-CNN uses region proposals to locate potential target objects, then performs precise classification and regression within these regions, enhancing accuracy. However, it is relatively slow and requires separate steps to generate regional proposals, extract features, and classify them, increasing the complexity of implementation and debugging. SSD, on the other hand, does not require region proposals, making it faster. It also uses multiple-scale feature maps for detection, which helps handle objects of different sizes. However, using multiple-scale feature maps complicates optimization and tuning, and it has lower accuracy for small object detection. YOLO series models are relatively simple and fast. However, they have lower accuracy and recall rates when dealing with small objects, making them prone to missing detections.

In recent years, the DETR model has introduced the Transformer architecture and end-to-end training to simplify traditional detection processes and offer more substantial global context modeling capabilities. However, it suffers from slow convergence, high computational costs, and a

Manuscript received August 6, 2024; revised January 9, 2025.

This work was supported by the Special Fund for Scientific Research Construction of the University of Science and Technology Liaoning.

Wenlong Ma is a Postgraduate student at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051 China. (e-mail:2098862229@qq.com).

Weisheng Liu* is a Professor at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China. (corresponding author to provide fax:0412-5929809; e-mail:succman@163.com).

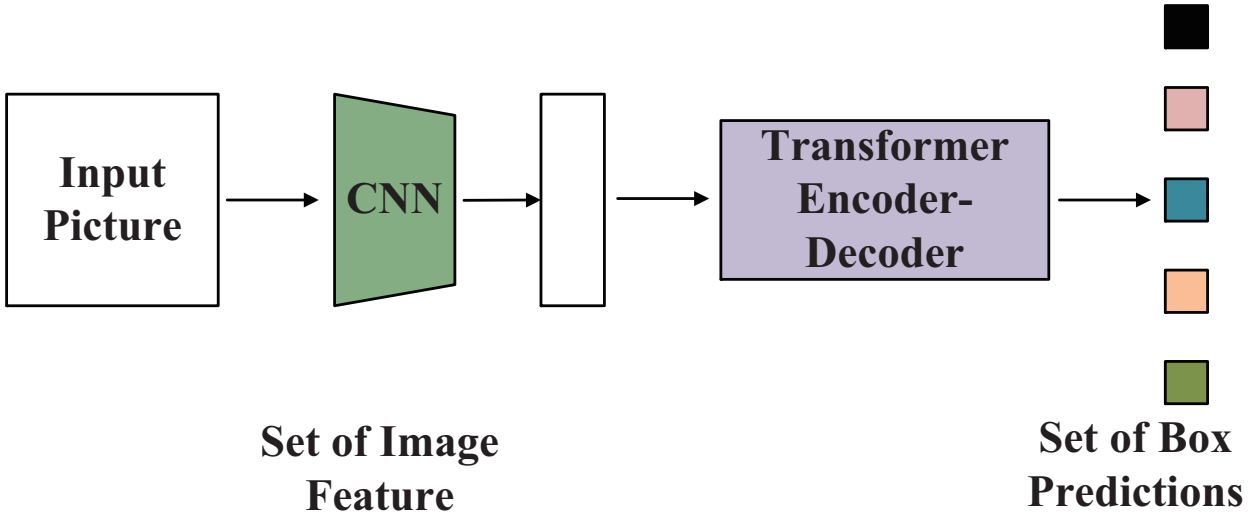

Fig. 1. The Workflow of Workflow of DETR

limited ability to detect small objects. More optimization and improvements are needed for small object detection.

In summary, object detection in UAV imagery is a challenging, yet promising research field. As technology continues to advance, it is expected to play an increasingly important role in a wide range of practical applications.

II. Related Work

A. Detection Transformer

DETR (Detection Transformer) is a deep learning model for object detection tasks, proposed by Facebook AI Research in 2020 [13]. DETR introduces the Transformer architecture to the object detection domain, achieving a direct end-to-end mapping from input images to object detection results, thereby avoiding the complex hand-crafted designs and multistage processes of traditional object detection methods.

As illustrated in Figure 1, DETR transforms the object detection problem into a sequence-to-sequence conversion task. Specifically, DETR views the object detection problem in an input image as a task of mapping an input sequence (image features) to an output sequence (object detection results). This allows DETR to leverage the powerful modeling capabilities of Transformers to perform global context modeling of both objects and backgrounds within the image.

The overall architecture of DETR comprises two key components: an Encoder and a Decoder. The encoder is responsible for converting the input image into a series of feature vectors, each representing a position in the input image. Based on these feature vectors, the Decoder incrementally generates object detection results using the Transformer's self-attention mechanism and feedforward neural networks. DETR employs a special target class, called "no-object," to represent areas in the image where no objects are present. Consequently, the Decoder can not only predict the class and location of objects but also identify background areas in the image by predicting "no-object" for specific positions.

The advantage of DETR lies in its ability to achieve end-to-end object detection, eliminating the complex processes and manual designs inherent in traditional methods. It strikes a good balance between speed and accuracy, maintaining high detection precision while achieving faster inference speeds. DETR has been widely applied in the field of object detection and has yielded significant results.

B. Swin Transformer

The Swin Transformer, introduced by Microsoft Research Asia in 2021, is a deep learning model based on the Transformer architecture that has demonstrated outstanding performance in computer vision tasks, particularly in image classification and object detection [14].

A key innovation of the Swin Transformer is introducing a "local-global" attention mechanism. The model divides the image into a series of small windows, where features are extracted within each window using a self-attention mechanism. Subsequently, interactions between windows are facilitated through cross-window attention. This design allows the model to focus on local details and global context, thereby improving the image's structural and semantic information modelling.

Additionally, the Swin Transformer incorporates a hierarchical Transformer structure, dividing the entire model into multiple stages, with each stage containing several groups of Transformer blocks. This hierarchical structure enhances the model's scalability and computational efficiency, enabling it to handle larger image sizes [15].

The Swin Transformer excels in image classification tasks, where pre-training on large-scale image datasets allows it to learn rich image feature representations. Moreover, it demonstrates strong performance in object detection tasks. When applied to object detection frameworks, the Swin Transformer achieves efficient and accurate object localization and recognition.

C. Slicing Aided Hyper Inference

The SAHI (Slicing Aided Hyper Inference) algorithm is a novel method for object detection, particularly well-suited for addressing the issue of small object detection in high-resolution images [16]. The advantage of SAHI is that it can divide large images into multiple smaller slices or fragments. This way, objects that appear very small in the large image will occupy a larger proportion and have more relative pixels in the smaller slices, thereby improving the model's ability

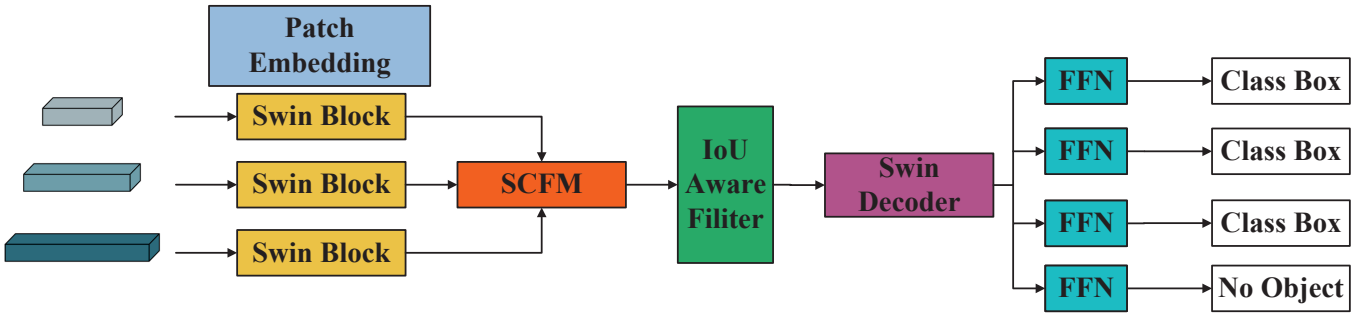

Fig. 2. The Overview Structure of OptiDETR

to detect small objects. Moreover, slicing a large image into multiple segments can reduce the background area in each segment, minimizing distractions and making it easier for the model to focus on the small objects to be detected.

When slicing the image, SAHI typically uses a sliding window strategy, resulting in overlapping regions between the windows. This prevents objects from being truncated at the edges due to slicing, thereby maintaining the completeness and accuracy of the detection results. By carefully designing the overlap ratio, it ensures that all objects appear fully in at least one slice. For objects that are divided by the slicing operation, SAHI identifies these fragments as belonging to the same object and merges them to restore the object's integrity, thus enhancing detection coverage and accuracy.

The SAHI algorithm significantly enhances the detection performance of small objects in high-resolution images through slicing and result fusion. It is applicable to various scenarios requiring high-precision object detection and can seamlessly integrate with existing detection models, making it an auspicious approach [17].

III. Method

Although the original DETR eliminates the region proposal generation and post-processing steps of traditional methods, simplifying the model workflow, it still has issues with long training times and insufficient accuracy in small object detection. Our proposed OptiDETR addresses these shortcomings by improving the backbone network encoder-decoder, introducing IoU constraints, and utilizing the SAHI algorithm. These enhancements significantly improve the accuracy of small object detection.

As shown in Figure 2, the OptiDETR model consists of four main parts: Backbone, SwinHyperEncoder, IoU Aware Filter, and SwinDecoder.

A. Backbone

The backbone is an object detection network's foundational component, extracting feature representations from the input image. It typically consists of a series of convolutional and pooling layers, which progressively reduce the size of the feature maps while increasing the number of channels. The role of the backbone is to capture low-level and mid-level features in the image, such as edges, textures, and shapes, which are crucial for tasks involving object localization and shape analysis. This study primarily utilizes CSPDarkNet and ResNet as the backbone networks.

CSPDarknet, as a lightweight neural network model, has fewer parameters compared to the ResNet network through the CSPNet (Cross Stage Partial Network) structure, and its network architecture is more advantageous, improving computational efficiency. Additionally, it exhibits better robustness when handling objects of different scales and types. The CSPNet structure within it can more effectively propagate gradient information to all layers of the network, mitigating the gradient vanishing problem and facilitating the training of deeper networks. Furthermore, CSPDarknet's modular design makes it very easy to expand and adjust. We can flexibly design and optimize the model according to actual needs by adding or reducing network layers and adjusting the feature map partition ratios. Although ResNet's residual structure also has some flexibility, its primary way of expansion is by increasing the number of residual blocks, which lacks flexibility in feature processing. Therefore, compared to ResNet, CSPDarkNet is superior as a backbone network.

B. Swin Hyper Encoder

The SwinHyperEncoder is an attention mechanism-based encoder that plays a crucial role in object detection tasks. It models long-range dependency relationships between features by adopting a hierarchical attention mechanism. The SwinHyperEncoder divides feature maps into multiple blocks and performs self-attention calculations within each block to capture global and local contextual information. This layered attention encoder enhances the semantic representation capability of features, enabling the network to better understand the target information within the image. Compared to the original DETR model's Eecoder, the Swin Hyper Encoder introduces several improvements, including hierarchical structures, local self-attention, and shifted window mechanisms. These advancements result in significant gains in computational efficiency, feature representation capability, and flexibility compared to the encoder in DETR. These improvements make the Swin Hyper Encoder particularly effective for small object detection tasks and hold promise for a wide range of applications. SwinHyperEncoder consists of three main parts: Patch Embedding, SwinBlock and SCFM. The following will introduce the composition and function of each part in detail.

1) Patch Embedding: The structure of Patch Embedding is illustrated in Figure 3. It segments the image into fixed-sized image patches using convolutional layers or linear projection layers. Each image patch is treated as a rectangular region and features are extracted through average pooling or convolutional operations. Subsequently, these extracted features are mapped to a fixed-dimensional vector space via a fully connected layer or convolutional layer.

Patch Embedding serves several key purposes. Firstly, it extracts local features, encoding local structural information in the image into vector representations. This aids in capturing fine-grained information such as textures, edges, and local shapes. Secondly, by segmenting the image into blocks, the size of each block is relatively small, significantly reducing the input dimensionality of the Transformer and thereby decreasing computational complexity. This enhances the efficiency of Transformer models in processing large-scale images. Additionally, Patch Embedding can introduce positional information. While extracting features from image blocks, positional information for each block is typically retained. Through positional encoding, the Transformer's encoder can perceive relationships between different positions.

2) SwinBlock: The SwinBlock consists of several structures, as illustrated in Figure 4. By adjusting the number of SwinBlocks, the network exhibits good scalability, and its computational process can be described by Equations (1) and (2):

$$X_l = \mathrm{W} - \mathrm{MSA}(\mathrm{LN}(X_{l-1})) + X_{l-1} \tag{1}$$

Where X_1 represents the output of the current layer l. The $\mathrm{LN}(X_{l-1})$ represents Applies Layer Normalization to the input X_1 , which is the output from the previous layer. The $W\text{-}MSA(\mathrm{LN}(X_{l-1}))+X_{l-1}$ represents uses a residual connection by adding the original input X_{l-1} to the output of the W-MSA module. This connection helps to stabilize training and retain important features from the previous layer.

$$X_{l+1} = \mathrm{MLP}(\mathrm{LN}(X_l)) + X_l \tag{2}$$

Where X_{l+1} represents the output of the current layer, which depends on the output of the previous layer,X_1. The $\mathrm{LN}(X_l)$ represents This applies Layer Normalization to X_1, normalizing its values to stabilize the input for the next operation. The $\mathrm{MLP}(\mathrm{LN}(X_l))$ represents after normalization, the result is passed through an MLP (Multilayer Perceptron). This MLP typically consists of a few linear layers with activation functions like GELU or ReLU in between. This is part of a Transformer block where the residual connections help prevent information loss across layers.

The SwinBlock consists of two sub-modules, Window-based Multi-head Self-Attention and multilayer perceptron.

Window-based W-MSA (Multi-head Self-Attention) is an attention mechanism in the Swin Transformer model, and its operation process is illustrated in Figure 5. It divides the input feature map into non-overlapping image blocks and computes self-attention within each image block to capture local dependencies within the image. W-MSA introduces window-based attention computation, performing attention calculations only within local windows around each image block. This windowed design makes attention calculation more efficient and suitable for processing large-scale images. W-MSA helps the model to model the image at a fine-grained level, capturing detailed information within the image blocks. The MLP module is another component of the Swin Transformer block. It consists of two fully connected layers used for nonlinear transformation and mapping of features within the image blocks.

3)SCFM (Scale Cross-Feature Fusion Module): As previously mentioned, the ShiftBlock aims to reduce computational complexity by dividing the input image into non-overlapping windows and performing self-attention calculations within these windows. While the ShiftBlock can reduce computational complexity, it suffers from insufficient information exchange between non-overlapping windows, thus losing the Transformer's ability to construct relationships globally using self-attention. In this context, the role of SCFM is highlighted. The structure of SCFM, as illustrated in Figure 6, improves upon the FPN-based structure, aiming to integrate the effects of features across different scales. The left side conveys strong localization features from Low-Level, while the right side conveys strong semantic features from High-Level. By combining ShiftBlock and SCFM, both computational complexity is reduced, and information exchange between different windows is enhanced.

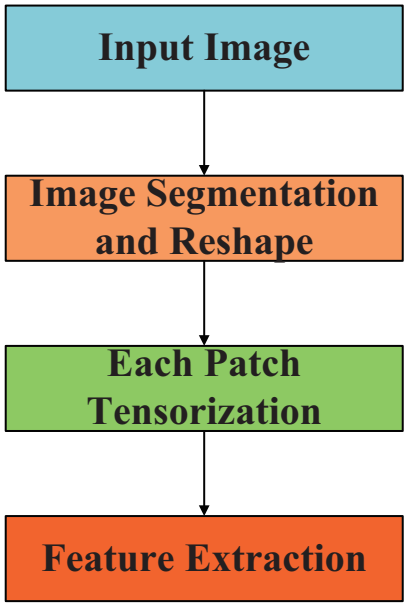

Fig. 3. The Workflow of Patch Embedding

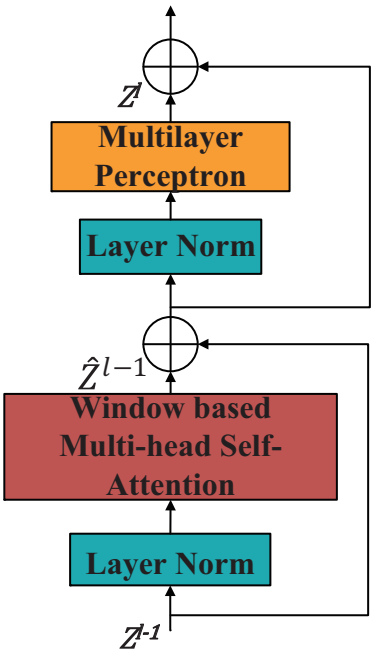

Fig. 4. The Structure of SwinBlock

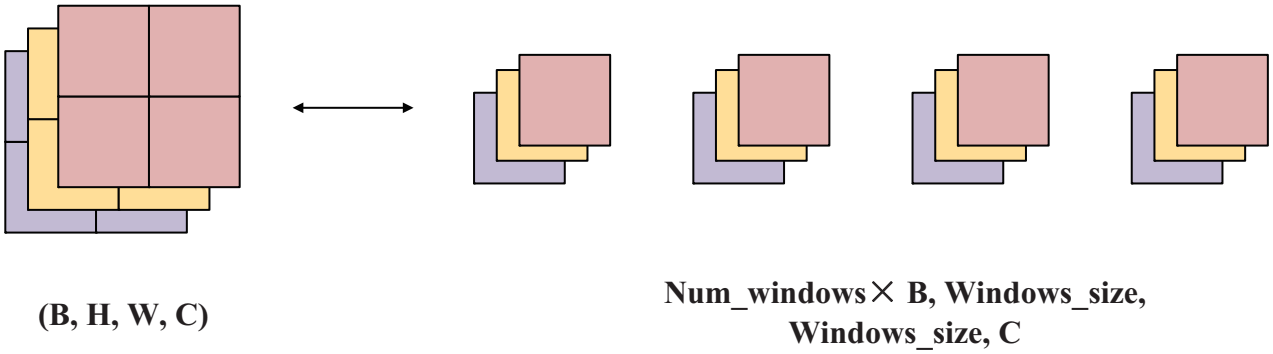

Fig. 5. The Workflow of W-MSA

C. IoU Aware Filter

The IoU Aware Filter is a filtering mechanism used for object detection aimed at screening and adjusting detection results based on the IoU of candidate boxes. This filter evaluates the matching degree between candidate boxes and real targets based on the IoU metric and filters and corrects detection results accordingly. Its function is to enhance the accuracy and robustness of object detection by filtering and adjusting candidate boxes, resulting in more precise and reliable detection results. Adding the IoU Aware Filter can incorporate IoU weights into the loss function. Compared to the original DETR, our model pays more attention to the overlap of bounding boxes. This optimization of the loss function allows it to handle overlapping and small objects better.

D. Swin-Decoder

The DETR decoder possesses a significant characteristic of modeling global context by decoding the encoder outputs through self-attention mechanisms, utilizing global context information for object detection. This enables DETR to have an advantage in handling relationships and occlusions between targets, accurately capturing the dependencies between targets. However, the DETR decoder also has some limitations. Firstly, the number of generated target boxes is fixed, which may lead to poor detection performance in scenes with a large number of targets or densely populated areas. Secondly, DETR faces challenges in detecting small targets, as it struggles to capture the detailed information of small targets, which typically require more fine-grained local information for accurate detection.

The main reason for the DETR decoder to adopt Swin-Block is to leverage the advantages of the Swin Transformer model to improve object detection performance. The DETR decoder needs to decode the encoder outputs and utilize global context information for object detection while introducing the Shifted Window Partition, which enables cross-window information exchange. The Swin-Decoder consists of several SW-Blocks, whose structure is illustrated in Figure 7, and its computational process can be described by Equations (3) and (4):

$$X_l = \text{SW-MSA}(\text{LN}(X_{l-1})) + X_{l-1} \tag{3}$$

Where SW-MSA(LN(X_{l-1})) represents SW-MSA is

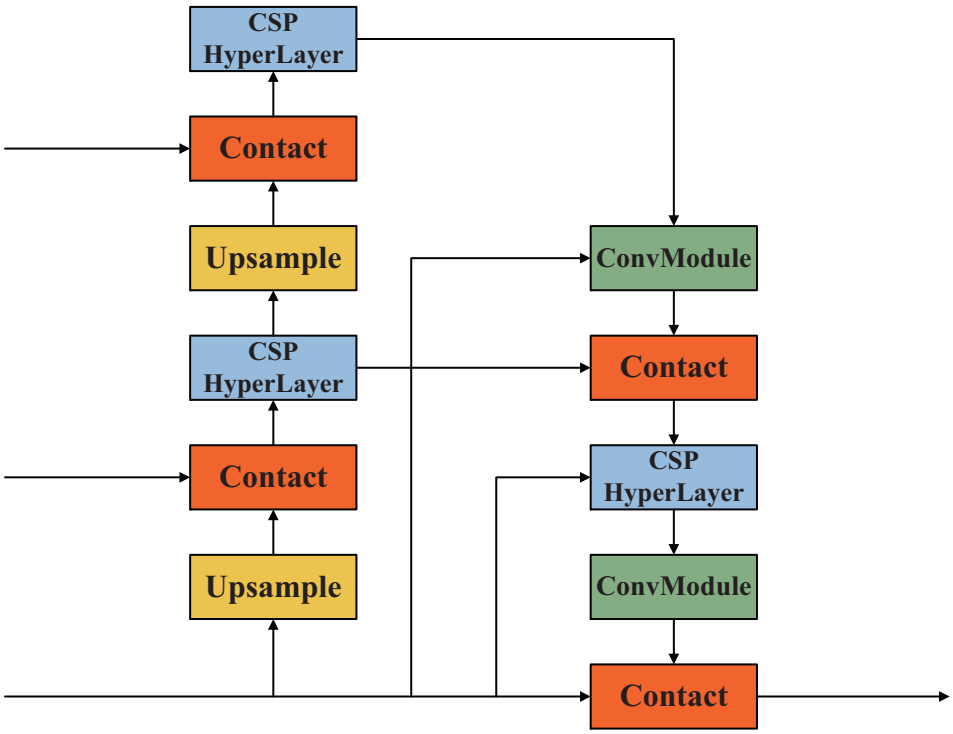

Fig. 6. The Overview of SCFM

Shifted Window Multi-Head Self-Attention, an attention mechanism applied within specific window regions. It calculates attention within shifted windows to capture local context information. This structure is used in the Swin Transformer to enhance the learning of local context through shifted window self-attention while preventing information loss across layers through residual connections.

$$X_{l+1} = \mathrm{MLP}(\mathrm{LN}(X_l)) + X_l \tag{4}$$

The most important attention mechanism module is Swin-MSA, whose operation process is illustrated in Figure 8. The previously mentioned SwinBlock can reduce computational complexity but lacks information exchange between non-overlapping windows. Therefore, Swin-MSA is introduced to facilitate information exchange across regions. By transmitting information across regions, the network can better capture global-level relationships. The key idea of Swin-MSA is to shift adjacent windows and establish connections between them, allowing information to flow between different windows. Consequently, SwinBlock can establish relationships between different windows through self-attention mechanisms, utilizing global context to optimize feature representations.

Thus, using SW-Block, the DETR decoder can fully utilize the Swin Transformer model's ability to model global context and optimize feature representations, thereby improving object detection performance and effectiveness.

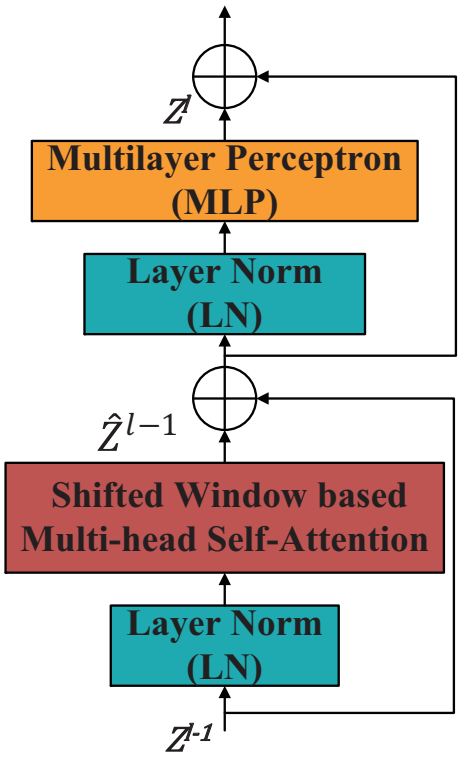

Fig. 7. The Structure of SW-Block

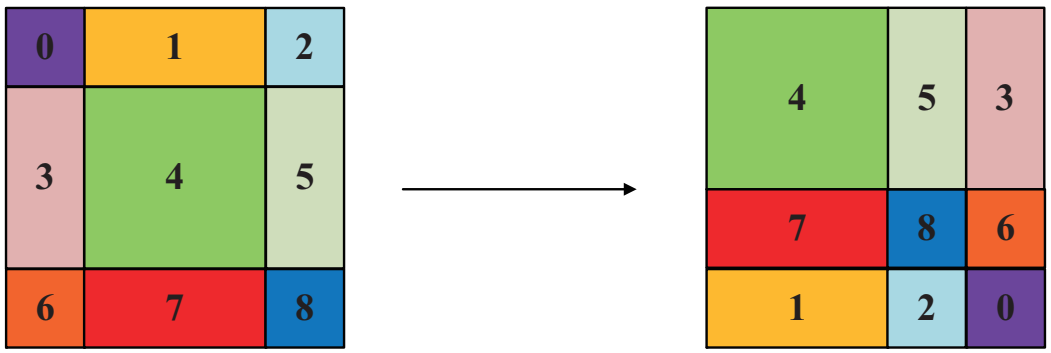

Fig. 8. The Workflow of SW-MSA

IV. Experiment

A. Dataset

The VisDrone dataset serves as a widely utilized benchmark in computer vision research, aimed at understanding visual data captured by drones equipped with cameras or general unmanned aerial vehicles. This dataset is curated by the AISKYEYE team at Tianjin University's Machine Learning and Data Mining Laboratory. The VisDrone2019 dataset comprises 288 video clips, totaling 261,908 frames and 10,209 static images. These data encompass various locations from 14 different cities across China, spanning thousands of kilometers and featuring diverse objects such as pedestrians, vehicles, and bicycles in urban and rural environments.

The images and video clips in the VisDrone dataset originate from various unmanned aerial vehicle platforms, covering a wide range of scenes, weather conditions, and lighting conditions. Annotations for over 2.6 million object bounding boxes have been manually annotated. Additionally, the dataset provides essential attribute information such as scene visibility, object categories, and occlusion status.

The VisDrone dataset offers researchers an experimental platform for conducting various critical computer vision tasks, including object detection, object tracking, and behavior analysis. By leveraging the VisDrone dataset, researchers can delve into the fusion of drone vision and computer vision, driving advancements in applications such as agriculture, aerial photography, rapid delivery, and surveillance.

B. Evaluation Merits

1) mAP (The mean Average Precision): mAP is a comprehensive metric employed in assessing the performance of object detection models. It calculates a model's average performance across various classes by computing the precision and recall for each class. Precision represents the proportion of samples predicted as positive that are indeed positive, while recall denotes the proportion of actual positive samples correctly predicted as positive. Subsequently, precision-recall

curves are plotted for each class, altering prediction confidence thresholds to obtain different precision and recall values. The AP value(the area under the precision-recall curve) is computed for each class. To derive AP, interpolation is performed among precision values at different recall points. A higher mAP value indicates the model's superior overall detection performance. The calculation formula for mAP involves averaging the AP values across all classes.

2) IoU : IoU is a metric used to evaluate the accuracy of bounding boxes in object detection. It computes the ratio of the intersection area between the predicted and actual bounding boxes to the union area. IoU50 and IoU75 evaluate object detection performance based on different IoU thresholds. IoU50 represents the evaluation metric using an IoU threshold 0.5, indicating that a prediction box is correct when IoU is greater than or equal to 0.5, reflecting a lenient threshold. IoU75, on the other hand, represents the evaluation metric using an IoU threshold of 0.75, implying that a prediction box is deemed correct when IoU is greater than or equal to 0.75, indicating a stringent threshold.

3) FLOPs (Floating Point Operations) represent the number of floating-point calculations a model requires during execution, serving as a crucial metric for assessing the complexity of algorithms or models. In deep learning, FLOPs are utilized to evaluate a model's computational demands, aiding researchers in selecting appropriate models and optimization strategies. Reducing FLOPs during model optimization helps lower computational costs and improve the model's inference speed.

Both mAP and IoU range from 0 to 1, and they are expressed in percentage form for the sake of data presentation consistency.

C. Baseline

To evaluate the performance of the OptiDETR model proposed in this paper, the experimental section selects a total of six mainstream image detection networks, namely PPYOLO, YOLO V4, YOLO V5, DETR-DC5, Anchor-DETR-DC5, and Efficient-DETR.

In addition to testing the models themselves, a comparison of the backbone components is also necessary. This paper adopts DarkNet-53, CSPDarkNet, and ResNet-50 as backbones for comparison. Below are introductions to these three backbone networks:

1) DarkNet-53: DarkNet-53 serves as the backbone network for YOLOv3, consisting of 53 convolutional layers and employing residual connections to facilitate training of deeper networks. It utilizes the Leaky ReLU activation function to enhance the model's nonlinear expressive capability. DarkNet-53 achieves a good balance between computational efficiency and detection accuracy, suitable for real-time object detection tasks.

2) CSPDarkNet: CSPDarkNet, the backbone network for YOLOv4, is an improvement upon DarkNet-53. Employing CSP (Cross Stage Partial) connections reduces redundant gradient information flow, enhancing computational efficiency and accuracy. It strengthens feature extraction capabilities while maintaining high detection speed, making it suitable for real-time application scenarios.

3) ResNet-50[18]: ResNet-50 is a residual network comprising 50 layers, utilizing skip connections to address the problem of gradient disappearance in deep networks. It employs Batch Normalization after each convolutional layer to stabilize the training process. ResNet-50 is widely applied in image classification and object detection tasks due to its efficient feature extraction capability and stable training performance.

All experimental results are the averages of five trials, with the optimal results displayed in bold and the next best results underscored.

D. Training Process

The training process can be divided into two main stages. Firstly, in Stage 1, the backbone network (ResNet or CSPNet) is pretrained using the ImageNet dataset. This pretraining aims to extract generic features from images through a large-scale image classification task. Secondly, in Stage 2, end-to-end joint training is conducted, including both Encoder and Decoder. Encoder training is the primary task of this stage, followed by IoU-Aware Filter, which selects the top 100 features with the highest classification scores

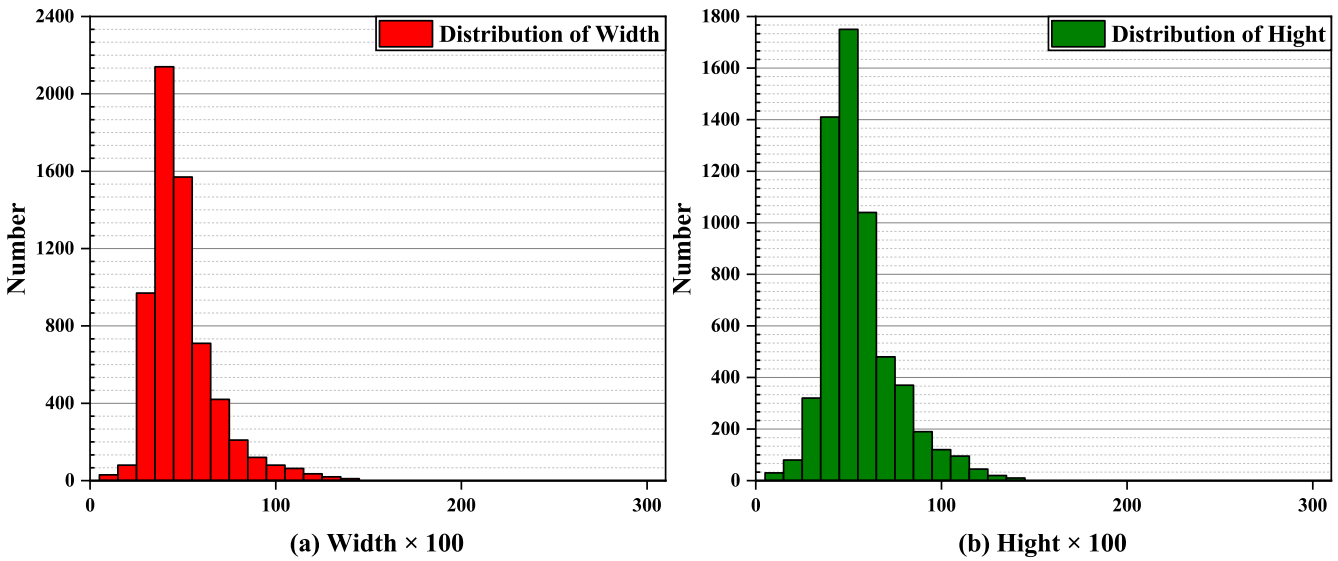

Fig. 9. The Statistics of Image Size in VisDrone

from the encoder output to initialize the target queries for the decoder. Decoder training is the final step of this stage.

Specific training details include the utilization of the AdamW optimizer, setting the number of epochs to 80, with a base learning rate of 0.0001, weight decay of 0.0001, global gradient clipping norm of 0.1, and linear warmup steps of 2000. The learning rate scheduling for the backbone network follows the DETR method. Additionally, the EMA (Exponential Moving Average) is employed for model parameter updates with a decay rate of 0.9999. Throughout the entire training process, various data augmentation techniques are employed, including random color distortion, expansion, cropping, flipping, scaling, and SAHI operations, to enhance the diversity of training data and the robustness of the model.

E. SAHI Feature Enhancement

The SAHI algorithm mentioned earlier belongs to the category of data augmentation methods. To address the issue of detecting small objects, the SAHI algorithm proposes a slice-based framework during the fine-tuning and inference stages. This involves segmenting the input image into overlapping blocks, thereby enlarging the pixel regions of small target objects. It is applicable to high-quality images such as remote sensing images and 4K UAV aerial images.

A detailed analysis of the VisDrone dataset, as depicted in Figure 9, reveals that the dataset comprises 6471 images and 34320 annotated bounding boxes, with an average image size of (1020, 1496). The median aspect ratio is 0.0246, and the median aspect ratio is 0.038. The dataset contains images with high resolutions and a significant number of small target objects.

Therefore, this study improves the data preprocessing method by employing the SAHI algorithm to address the issue of small object detection. The SAHI algorithm segments the input image into overlapping blocks to expand the pixel regions of small target objects, enhancing the model's capability to detect small objects and improving overall detection performance.

Furthermore, the SAHI algorithm performs well in handling high-quality images. VisDrone dataset images typically possess high resolutions and abundant detailed information. The SAHI algorithm effectively utilizes this information to provide more contextual information, enhancing the accuracy and robustness of object detection.

Finally, as a data augmentation method, the SAHI algorithm introduces diversity and richness, thereby increasing the diversity of training data. By segmenting the input image into multiple overlapping blocks, more training samples can be generated, augmenting the dataset and enhancing the model's generalization ability and robustness, which is particularly crucial for addressing the issue of small object detection in the VisDrone dataset.

V. Result and Analysis

A. Object Detection Performance Comparison

When conducting object detection using the VisDrone dataset, this paper compared with previously mentioned benchmark models, and the specific results are presented in Table I. The results demonstrate that the OptiDETR model, whether paired with ResNet-50 or CSPDarkNet, consistently achieved optimal or near-optimal results, thus fully showcasing its superiority. Specifically, when the OptiDETR model is paired with CSPDarkNet as the backbone, compared to other non-OptiDETR models, it exhibited improvements of 2.89%, 1.67%, and 1.64% in mAP, IoU50, and IoU75, respectively. Similarly, when the OptiDETR model is paired with ResNet-50 as the backbone, it demonstrated improvements of 1.16% and 0.71% in mAP and IoU50, respectively, compared to other non-OptiDETR models.

Furthermore, the OptiDETR model showed advantages in terms of parameter count and computational complexity. When using ResNet-50 as the backbone, the parameter count was 39.6, and the FLOPs were 128; while when using CSP-DarkNet as the backbone, the parameter count was 41.5, and the FLOPs were 130. This indicates that while maintaining high performance, OptiDETR incurred less computational and memory overhead.

Among other models, PPYOLO,YOLOv4 and YOLOv5 exhibited relatively good performance but had slightly higher parameter counts and computational complexities compared to OptiDETR. DETR-DC5, Anchor-DETR-DC5, and Efficient-DETR models performed well under specific network and input size configurations but still could not surpass the performance of the OptiDETR model.

B. Quantity Experiments of SwinBlock and SW-Block

The experiment involved training for 80 epochs, utilizing CSPDarkNet as the backbone network, and setting the image size to 640×640. Such training settings contribute to better learning of features relevant to object detection tasks and enhance the model's performance.

TABLE I
Comparison with Other Object Detection Networks

Model	Backbone	mAP	IoU 50	IoU 75	Params(M)	FLOPs
PPYOLO	DarkNet-53	47.10	68.60	58.31	38.09	**109.10**
YOLOv4	CSPDarkNet	51.40	69.20	<u>61.00</u>	40.15	123.05
YOLOv5	CSPDarkNet	51.50	71.30	60.61	42.05	<u>118.11</u>
DETR-DC5	ResNet-50	45.40	65.00	55.25	39.12	152.04
Anchor-DETR-DC5	ResNet-50	48.60	67.90	56.35	**34.61**	136.08
Efficient-DETR	ResNet-50	49.90	68.20	58.00	<u>38.21</u>	216.22
OptiDETR	ResNet-50	<u>52.10</u>	<u>71.80</u>	59.35	39.60	128.15
OptiDETR	CSPDarkNet	**53.40**	**73.00**	**62.00**	41.50	130.34

TABLE II
COMPARISON OF DIFFERENT SWINBLOCK AND SW-BLOCK

SwinBlock Number	SW-Block Number	mAP	IoU50	IoU75	Params(M)	FLOPs
4	4	44.81	64.00	46.78	**35.61**	**101.01**
4	6	46.27	65.92	48.77	36.82	107.25
6	4	49.63	68.75	50.19	38.91	119.34
6	6	**53.40**	71.35	**55.80**	43.52	130.23
8	6	53.31	**72.18**	54.13	42.62	160.22

As depicted in Table II, the influence of different combinations of SwinBlocks and SW-Blocks on model performance is apparent. Specifically, three combinations were considered: SwinBlock of 6 and SW-Block of 4, SwinBlock of 8 and SW-Block of 6, and both SwinBlock and SW-Block set to 6.

Firstly, the combination where SwinBlock and SW-Block are set to 6 is observed. According to the data performance, this combination achieved the highest values for performance metrics, with mAP, IoU50, and IoU75 reaching 53.40, 71.35, and 55.80, respectively. This indicates the capability of this combination to accurately detect more targets in object detection tasks with higher overlap. Secondly, when SwinBlock is set to 8 and SW-Block to 6, there is a slight decrease in model performance, with mAP at 53.31, IoU50 at 72.18, and IoU75 at 54.13. This suggests that for this particular task or dataset, this combination might not adequately capture the features and contextual information of the targets, resulting in a performance decline.

Achieving optimal mAP in object detection tasks related to drone imagery enables more accurate identification and localization of small objects, even when they are embedded within complex and cluttered backgrounds. This enhanced capability ensures that the model can effectively detect these objects with a high degree of confidence, significantly boosting its robustness in real-world applications. Furthermore, when the model achieves superior performance across a wide range of IoU thresholds, it indicates that it maintains a consistent and reliable detection performance in diverse and challenging scenarios, ranging from simple to highly complex scenes and backgrounds. This consistency demonstrates the model's strong generalization capabilities, making it adaptable to various environments and conditions.

Excelling at an IoU threshold of 75

C. Ablation Experiment

This paper contrasts the results with and without the utilization of SCFM, as depicted in Table III. SCFM, a method derived from FPN architecture enhancements, aims to integrate the effects of features across different scales. Experimental findings demonstrate that the introduction of SCFM not only enhances performance but also improves computational efficiency. SCFM facilitates information exchange between different windows by conveying strong localization features from the bottom up and strong semantic features from the top down. Such feature fusion contributes to enhancing accuracy and robustness in object detection.

This paper similarly contrasts the results with and without the utilization of SAHI, as presented in Table IV. According to the comparative results, it is evident that the performance of object detection slightly improves when SAHI is employed. Specifically, mAP increases from 52.91 to 53.18, IoU50 rises from 70.52 to 71.37, and IoU75 enhances from 52.05 to 54.53, representing improvements of 0.27%, 0.85%, and 2.48%, respectively. To some extent, this outcome demonstrates the positive impact of SAHI on object detection tasks, aiding the model in better learning and adaptation to complex scenarios.

VI. CONCLUSIONS

This paper examines the common challenges in image recognition and proposes a model specifically designed for UAV image recognition based on DETR, termed OptiDETR. OptiDETR introduces an efficient hybrid encoder to replace the original Transformer encoder, enabling efficient processing of features at different scales. The decoder's initialization scheme for object queries is crucial for detection performance. OptiDETR employs IoU-aware query selection to enhance performance further, incorporating IoU constraints during training. Additionally, this study introduces the SW-Block in the DETR decoder, leveraging the Swin Transformer model's capability for global context modeling and optimized feature representation.

Experimental results indicate that the OptiDETR model performs superiorly on the VisDrone dataset compared to other mainstream image recognition models, establishing it as a new baseline for UAV image recognition. Finally, the various submodules of the OptiDETR model can be easily integrated into other models, offering new avenues for research in related fields. This study improves the robustness and performance of object detection systems and lays the groundwork for future research and applications.

TABLE III
COMPARISON OF SCFM

	Input_Size	Epoch	mAP	IoU50	IoU75	Params(M)	FLOPs
Without SCFM	640	80	51.20	68.91	49.62	39.32	126.05
With SCFM	640	80	53.41	71.37	54.24	42.28	136.19

TABLE IV
COMPARISON OF SAHI

	Input_Size	Epoch	mAP	IoU50	IoU75
Without SAHI	640	80	52.91	70.52	52.05
With SAHI	640	80	53.18	71.37	54.53

REFERENCES

[1] V. C. Hollman, "Drone Photography and the Re-Aestheticisation of Nature," *Decolonising and Internationalising Geography: Essays in the History of Contested Science*, pp. 57–66, 2020.

[2] S. Zhao, "The Role of Drone Photography in City Mapping," in *Application of Intelligent Systems in Multi-modal Information Analytics: Proceedings of the 2020 International Conference on Multi-model Information Analytics (MMIA2020), Volume 2*. Springer, 2021, pp. 343–348.

[3] I. Suroso and E. Irmawan, "Analysis of UAV Multicopter of Air Photography in New Yogyakarta International Airports," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 17, no. 1, pp. 521–528, 2019.

[4] X. Li, F. Wang, A. Xu, and G. Zhang, "UAV Aerial Photography Target Detection and Tracking Based on Deep Learning," in *Proceedings of the 5th China Aeronautical Science and Technology Conference*. Springer, 2022, pp. 426–438.

[5] J. Sun, B. Li, Y. Jiang, and C.-y. Wen, "A Camera-based Target Detection and Positioning UAV System for Search and Rescue (SAR) Purposes," *Sensors*, vol. 16, no. 11, p. 1778, 2016.

[6] J. Li, D. H. Ye, T. Chung, M. Kolsch, J. Wachs, and C. Bouman, "Multi-Target Detection and Tracking from a Single Camera in Unmanned Aerial Vehicles (UAVs)," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4992–4997.

[7] J. Zhao, X. Zhang, C. Gao, X. Qiu, Y. Tian, Y. Zhu, and W. Cao, "Rapid Mosaicking of Unmanned Aerial Vehicle (UAV) Images for Crop Growth Monitoring using the SIFT Algorithm," *Remote Sensing*, vol. 11, no. 10, p. 1226, 2019.

[8] J. Liu, Q. Wei, and Y. Bai, "Fast Stitching of UAV Images based on Improved SURF Algorithm," in *2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*. IEEE, 2021, pp. 1310–1313.

[9] L. Xiaoping, L. Songze, Z. Boxing, W. Yanhong, and X. Feng, "Fast Aerial UAV Detection using Improved Inter-Frame Difference and SVM," in *Journal of Physics: Conference Series*, vol. 1187, no. 3. IOP Publishing, 2019, p. 032082.

[10] D. Avola, L. Cinque, A. Diko, A. Fagioli, G. L. Foresti, A. Mecca, D. Pannone, and C. Piciarelli, "MS-Faster R-CNN: Multi-stream backbone for improved Faster R-CNN object detection and aerial tracking from UAV images," *Remote Sensing*, vol. 13, no. 9, p. 1670, 2021.

[11] Y. Hu, X. Wu, G. Zheng, and X. Liu, "Object Detection of UAV for Anti-UAV based on Improved YOLO V3," in *2019 Chinese Control Conference (CCC)*. IEEE, 2019, pp. 8386–8390.

[12] L. Yundong, D. Han, L. Hongguang, X. Zhang, B. Zhang, and X. Zhifeng, "Multi-Block SSD based on Small Object Detection for UAV Railway Scene Surveillance," *Chinese Journal of Aeronautics*, vol. 33, no. 6, pp. 1747–1755, 2020.

[13] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end Object Detection with Transformers," in *European Conference on Computer Vision*. Springer, 2020, pp. 213–229.

[14] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[15] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong *et al.*, "Swin Transformer V2: Scaling up Capacity and Resolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 12 009–12 019.

[16] F. C. Akyon, S. O. Altinuc, and A. Temizel, "Slicing Aided Hyper Inference and Fine-Tuning for Small Object Detection," in *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 966–970.

[17] N. Haider and A. K. Mehta, "A Novel Approach for Small-Object Detection based on Fine-tuning and Sliced Inference," in *2022 2nd Asian Conference on Innovation in Technology (ASIANCON)*. IEEE, 2022, pp. 1–6.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.