SSE-YOLO: Efficient UAV Target Detection With Less Parameters and High Accuracy

Yong Lu * and Minghao Sun

Posted Date: 15 January 2024

doi: 10.20944/preprints202401.1108.v1

Keywords: UAV; small target detection; YOLOv8; feature extraction; deep learning

Article

SSE-YOLO: Efficient UAV Target Detection with Less Parameters and High Accuracy

Yong Lu [1] and Minghao Sun [2]

[1] Key Laboratory of Ethnic Language Intelligent Analysis and Security Governance of MOE, Minzu University of China, Beijing, 2006153@muc.edu.cn

[2] School of Information Engineering , Minzu University of China, Beijing 100081, China ; 23302093@muc.edu.cn

* Correspondence: 2006153@muc.edu.cn

Abstract: Despite UAV multi-target detection exhibits considerable developmental potential worldwide, it suffers distinct challenges compared with traditional tasks in this field. These challenges include insufficient feature extraction capabilities for small targets, limited capabilities for multi-dimensional feature fusion, as well as constraints on hardware computation parameters. Especially in mission scenarios such as disaster detection, these challenges will be further amplified. Consequently, this paper introduces SSE-YOLO, an innovative YOLO framework algorithm specially designed to address these challenges. To enhance the model's feature extraction capability, we employ the SPDConv module to replace the original Conv in the backbone section, utilizing depth-separable convolution instead of traditional convolution pooling. Concurrently, we eliminate the SPPF module at the bottom and address a new Separate Kernel Attention Pyramid Pooling (SKAPP) module, substantially enhancing the feature fusion capability at the model's core. Moreover, to address the challenge of multi-dimensional feature fusion, we replace the Concat module of the neck and head with E-BiFPN, transmitting feature information from the backbone to the lower network through four CBS blocks, which effectively resolves the issue of lost contextual information in the model. Meanwhile, SSE-YOLO undergoes ablation experiments on the VisDrone2019 dataset to evaluate its effectiveness against alternative methods, and experimental results illustrate the model's exceptional precision in detecting UAV targets. In comparison to models with comparable experimental accuracy, SSE-YOLO requires remarkably fewer parameters. On the VisDrone2019-DET-test-dev dataset, SSE-YOLO enhances mAP by 17.3%, with a 42.5% reduction in the parameter amounts. Therefore, the proposed method effectively tackles the challenge of reconciling low parameters and high accuracy, presenting a novel pathway for deep learning-based UAV multi-target detection.

Keywords: UAV; small target detection; YOLOv8; feature extraction; deep learning

1. Introduction

The integration of unmanned aerial vehicle (UAV) remote sensing images and deep learning detection technology has emerged as a popular research area [1,2]. Initially, drones were solely utilized by the military; however, with the rapid technological advancements over the recent years, their applications have expanded to all sectors of society. Due to their small size, flexible movement, and minimal restriction by geography, drones have the potential to greatly expand the scope of human monitoring and provide tremendous opportunities to explore unknown or hazardous areas, this has had a significant impact on multiple areas such as disaster survey, traffic control, landform exploration, and industrial risk management, among others. Drones can create huge value at minimal cost, inspiring researchers to optimize the technology. In recent years, researchers have proposed many excellent model algorithms. These methods have greatly improved the recognition accuracy of drones on the original basis, further promoting the development and application scope of the field of drone image recognition.

The effectiveness of drones has rendered the identification of targets a critical domain in the realm of computer vision. Target detection has undergone notable advancements recently, mainly

 doi:10.20944/preprints202401.1108.v1

due to the rise of deep learning methods[3]. The advent of deep learning has significantly enhanced the overall accuracy of target detection algorithms. Deep neural network methods have increasingly become the primary approach to tasks such as image classification, target detection, and image segmentation[4]. Nonetheless, the amalgamation of these methods still poses many challenges. Firstly, the hardware equipped with UAVs is often resource-constrained, raising an urgent demand in lightweight deployed models for fast inference and low latency[5]. Secondly, as shown in **Figure 1**, compared to traditional images, drone aerial images have multitudes of small and complex targets, as well as complex backgrounds[6] and overlapping occlusions[7], which are caused by aerial photography angles. The complexity of images, their larger scene, and other characteristics present major challenges in target detection[8].

Detection methods used in target recognition aim to identify specific objects or features in a scene. Examples include road vehicle detection for smart transportation, forest fire detection for smart disaster prevention, and industrial parts defect detection. These tasks are typically performed in a clear background environment. Since most recognition targets belong to a large category, few factors affect recognition efficiency. Historically, target detection using deep learning was dominated by convolutional neural networks (CNNs) like early R-CNN[20], AlexNet[21], and VGG[22]. These networks employed CNN for object classification and bounding box prediction. Google later proposed the Vision Transformer[23], a revolutionary model network that introduced the Self-Attention mechanism of the Transformer and eliminated the sequential structure of CNN. This allowed for parallel training and improved the model's ability to extract features by obtaining global information. However, several experimental studies indicate that the visual Transformer's performance may not be optimal in various scenarios. Additionally, when the image resolution is high and contains numerous pixels, the Transformer's calculation based on global self-attention results in a significant computational burden. To address these issues, Liu et al. proposed the Swin Transformer[24], which features a hierarchical design and sliding window operation, effectively resolving the aforementioned problems. The sliding window operation restricts attention calculation to a window, which introduces the locality of CNN and reduces the amount of calculation. To enhance the model's accuracy in future research, Mehta et al. proposed MobileViT[25]. This hybrid architecture of CNN and Transformer provides spatial induction bias through CNN, accelerating the network's convergence and inference speed. Spatial information can be introduced to eliminate the influence of additional spatial position offset, thereby improving the simplicity of network migration. This allows MobileViT to achieve good performance using basic data enhancements while greatly reducing the number of parameters. Although these methods have made great progress on the original basis, they cannot directly obtain identification information such as object category and size through the network model, so they still belong to the two-stage detection method based on region proposal. Simultaneously, most of these two-stage detection methods demand substantial memory overhead and computing resources, making them challenging to directly deploy on low-power graphics processors, such as edge devices. In contrast, there is another one-stage method that directly processes the input image through a network model to obtain identification information. This type of method is also the most important in the current field of UAV image recognition. Undoubtedly, the single-stage recognition method most recognized by researchers in this field is the YOLO series[9–16].

However, the YOLO series does not perform satisfactorily in recognizing small targets. Unlike traditional captured images, images captured by drones often contain multiple types of small and low-pixel targets. Indeed, small object detection poses a significant challenge as smaller objects inherently have lower resolution and limited contextual information for the model to learn. Moreover, they often coexist with larger objects in the same image, leading to a feature-learning process dominated by the larger objects and leaving the smaller ones undetected. Simultaneously, the visibility of these targets is often easily affected by factors such as pixels, light, and complex backgrounds due to the problem of aerial photography angle. This can confuse the detection target and the background in the YOLO series. In target detection tasks, it is common for low-pixel and small-size targets to have their feature values ignored due to a lack of details. Numerous outstanding

 doi:10.20944/preprints202401.1108.v1

target detection methods excel in traditional image recognition tasks but often struggle to address the unique challenges present in the UAV domain.

To address these challenges, researchers have focused on elevating the performance of small object detectors by improving feature representation and optimizing data augmentation techniques [17].Despite substantial progress in enhancing object detection through these methods, they still exhibit certain limitations. Specifically, these approaches reveal insufficient generalization capabilities, particularly when dealing with small objects in the context of multi-class object detection [18]. Complex environmental factors can lead to inaccurate detection outcomes, and existing models may fall short in extracting adequate feature information for small objects to transmit effectively to the subsequent network. In scenes characterized by complex, blurry, and polluted backgrounds, a notable amount of information is lost, posing a formidable challenge for detecting these objects. Additionally, models with high accuracy often come burdened with a large number of computational parameters, imposing a significant strain on embedded hardware devices. Consequently, the field of UAV target recognition demands a model capable of extracting features from intricate images with high accuracy and a reduced parameter volume.

In response to the aforementioned issues, this paper introduces an enhanced SSE-YOLO algorithm structure based on YOLO v8. The outlined model architecture and the accompanying experimental results demonstrate its efficacy for detecting complex small targets. This model was compared against several existing models and exhibited superior performance in the realm of UAV target recognition. The experimental results indicate that our model achieves higher accuracy with a smaller number of parameters, underscoring that SSE-YOLO is more suitable for multi-target recognition tasks in UAV remote sensing images.

The main contributions of this paper can be summarized as follows:

1. In the backbone part, we introduce a new Separate Kernel Attention Pyramid Pooling (SKAPP) as a replacement module for SPPF. In contrast to the original SPPF module, SKAPP incorporates the concept of Large Separable Kernel Attention (LSKA), leading to a significant improvement in the efficiency of model feature extraction and fusion while effectively managing the calculation parameter count. Simultaneously, we use a new convolution SPDConv for low resolution and small targets to replace the traditional Conv. This convolution can greatly improve the accuracy of the model for small target recognition.
2. Four CBS models with a stride and a kernel size of 1 are inserted between the backbone portion and the neck portion to store and transmit the image feature information of the trunk component. This effectively remedies the loss of significant feature information observed in the baseline model.
3. For the feature fusion processing in the neck & head segment, we introduced an Easy Bidirectional Feature Pyramid Network(E-BIFPN) to replace the Concat module. This module can adaptively adjust its structure based on the number of input heads, thereby achieving a more efficient feature fusion operation.
4. Compared with several current advanced deep object detection models, our model stands out with exceptional performance while effectively managing the calculation parameter count. Results of the large-weight model experiments demonstrate that SSE-YOLO-L utilizes only 52% of the parameters of YOLOV8-L while achieving similar accuracy. The SSE-YOLO low-weight model(n,s,m) outperforms other models with similar parameter levels in recognition accuracy and other outcome parameters.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1108.v1

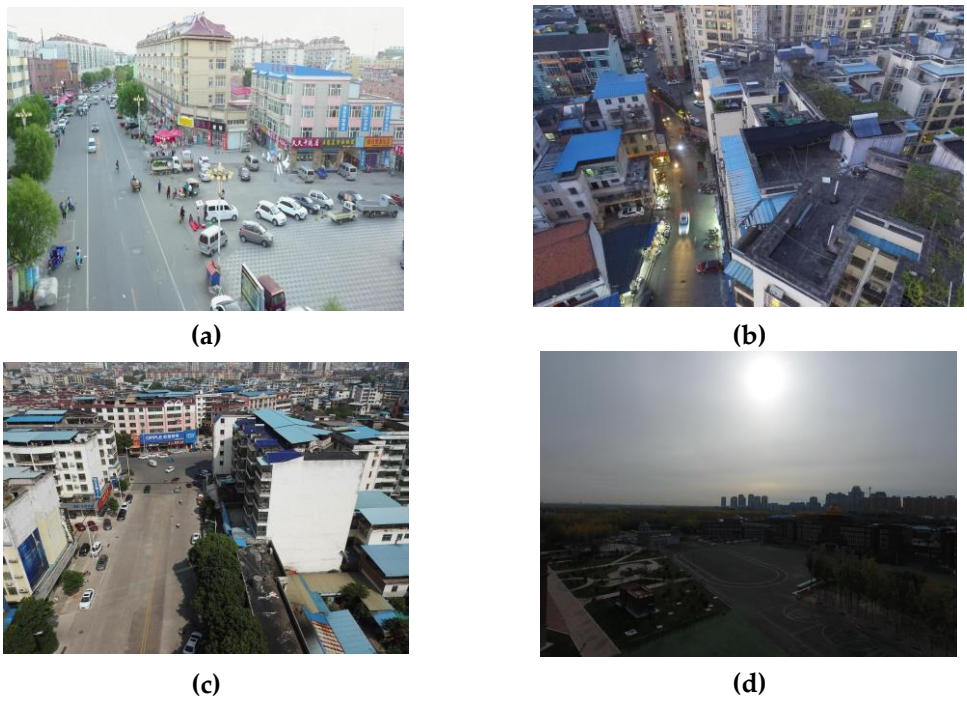


Figure 1. Sample images taken from drones:(a) People and vehicles on the street;(b) Complex real estate layout in the old city;(c)Road covered by trees and buildings;(d)low visibility weather.

2. Related Work

The rapid advancement of drone technology has introduced new avenues for image acquisition, offering enhanced perspectives and dimensional target information from elevated angles. However, these images come with inherent challenges, including diverse scenes, complex backgrounds, low visibility, and target overlap. Traditional target detection methods lose their effectiveness in the face of these challenges, and their application to multi-target recognition tasks in UAV images becomes exceptionally challenging. Fortunately, the development of large-scale datasets tailored for drone-captured images has played a crucial role in achieving substantial breakthroughs in object detection through the application of deep learning-based methods[19].

Researchers in the field have proposed many methods and approaches to address the above challenges. In this section we will focus on the ideas that YOLOv5 and YOLOv8 provide for our work. Jocher et al.[13] proposed YOLOv5 in 2021. Mosaic data enhancement and adaptive anchor frame calculation were introduced at the input end to improve the network's detection effect on small targets. At the same time, a focus module was introduced in the backbone part, which improved the input image processing speed and reduced the amount of model calculations. To improve the feature fusion capability of the model, the CSP2 structure is used in the neck. The team proposed YOLOv8[16] in 2023, using the residual idea to introduce the new C2f module to obtain more gradient information and alleviate the problem of context information loss. To improve the expressiveness of multi-scale features, FPN is modified and PAN is introduced to aggregate shallow and deep feature maps. The above two methods provide important basic ideas for densely distributed object detection tasks. Most of the methods proposed by subsequent researchers are based on these two methods and have been improved to address problems in different directions, such as overlapping target instances, insufficient feature extraction capabilities, etc.

Yan et al.[26] enhanced YOLOv5 by introducing the BottleneckCSP-2 module to replace the BottleneckCSP module in the backbone and incorporating Squeeze and Excitation (SE) attention into the architecture. This modification demonstrated excellent performance in addressing the issue of overlapping target instances. Cao et al.[27] proposed GCL-YOLO, constructing a lightweight backbone network based on GhostConv and employing the Focal Effective Intersection Loss (Focal-EIOU) as the positioning loss. This approach resulted in a 76.7% reduction in parameters compared

 doi:10.20944/preprints202401.1108.v1

to the baseline model. Zhu et al.[28] introduced TPH-YOLOv5, which replaced the original prediction head with Transformer Prediction Heads (TPH) to leverage the prediction potential of the self-attention mechanism. Additionally, a convolutional block attention model (CBAM) was integrated to identify attentional regions in dense object scenes. Compared to the previous state-of-the-art method DPNet, this approach improved the Average Precision (AP) result by 1.81% and demonstrated superior detection accuracy for small targets. Wang et al.[29] proposed UAV-YOLOv8, incorporating the BiFormer attention mechanism to optimize the backbone network, enhancing the model's ability to extract key features. They also designed a feature processing module named Focal FasterNet block (FFNB) and introduced two new detection scales based on this module to fuse shallow and deep feature information efficiently. This method achieved an average accuracy improvement of 7.7% compared to the baseline model. Wang et al.[30] introduced SMFF-YOLO, which integrated the ELAN-SW object detection prediction head to enhance the detection accuracy of small objects. The Adaptive Atrous Space Pyramid Pooling (AASPP) module was also introduced to achieve adaptive feature fusion capability rapidly.

Many studies have demonstrated that modifying the model's backbone and introducing attention mechanism modules can enhance the feature extraction capabilities and contribute to addressing UAV target recognition challenges. These approaches have significantly influenced our work by providing valuable insights into problem-solving strategies. However, these methods still face challenges such as high background confusion rates for small targets, substantial computational parameter requirements, and overall low accuracy. In light of these challenges, we build upon the foundations laid by previous researchers and propose SSE-YOLO, incorporating two key enhancements for addressing small target detection with low pixel counts. The first is to strengthen the feature extraction ability of the model. We introduced SPDConv to replace the backbone Conv to avoid the problem of fine-grained information loss. At the same time, we canceled the SPPF module at the bottom of the backbone, introduced the LSKA attention mechanism, and proposed the Separate Kernel Attention Pyramid Pooling (SKAPP) module. These modifications significantly enhance the feature extraction capabilities of the model's backbone. Secondly, our attention is directed toward preserving contextual feature information while reinforcing the model's multidimensional feature fusion capability. We insert four Contextual Feature Preservation (CSB) modules between the backbone and the neck of the model to retain essential feature information. The output information flow from CSB serves as one of the input information flows for the Enhanced Bi-directional Feature Pyramid Network (E-BiFPN), a structure proposed to replace the Concat module in the neck and head. This replacement reduces computational parameters and improves the fusion ability of multidimensional feature information. Through these enhancements, our method successfully achieves the goal of minimizing parameters while maximizing accuracy. Compared with the baseline model, our approach demonstrates outstanding performance across parameters such as mAP, Parameters, Precision, and various small target detection accuracy metrics.

3. Methods

This section details the improvement ideas of the methods and modules proposed in this article. We will provide a thorough breakdown of the model's structure, revised components, creative modules, and resolved issues. Firstly, traditional convolutional neural networks encounter challenges when processing low-pixel images and recognizing small targets. To address this issue, we replaced the Conv module in the backbone of the baseline model with Space-to-Depth Convolution (SPDConv). This unique convolution technique eliminates downsampling and pooling. It is implemented at the base of the backbone. We present Separable Kernel Attention Pyramid Pooling (SKAPP), which fuses aspects of SPPF (Spatial Pyramid Pooling-Fast) and LSKA (Large Separable Kernel Attention). It uses a kernel decomposition type that works with depth dilation convolution. This approach implements a feature fusion layer that requires fewer parameter calculations, improving efficiency. Secondly, to address the issue of contextual feature information loss, we developed four CBS modules and integrated them into the backbone and neck of the network. This resulted in a new network structure that significantly decreased the expense of

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1108.v1

contextual feature information. Lastly, we present the E-BIFPN structure module. Compared to the Concat module, E-BIFPN employs a weighted bidirectional feature pyramid network structure. This not only improves the fusion of multi-dimensional feature information in the neck but also decreases the parameter calculation required. This resolved the problem of parameter explosion during neck upsampling. The overall architecture of SSE-YOLO is illustrated in **Figure 2**.

SSE-YOLO is further subdivided into SSE-YOLO (nano), SSE-YOLO (small), SSE-YOLO (medium), and SSE-YOLO (large) based on the model's parameter size. The corresponding values for each model on the VisDrone2019 dataset will be presented and analyzed in detail in Section 4.

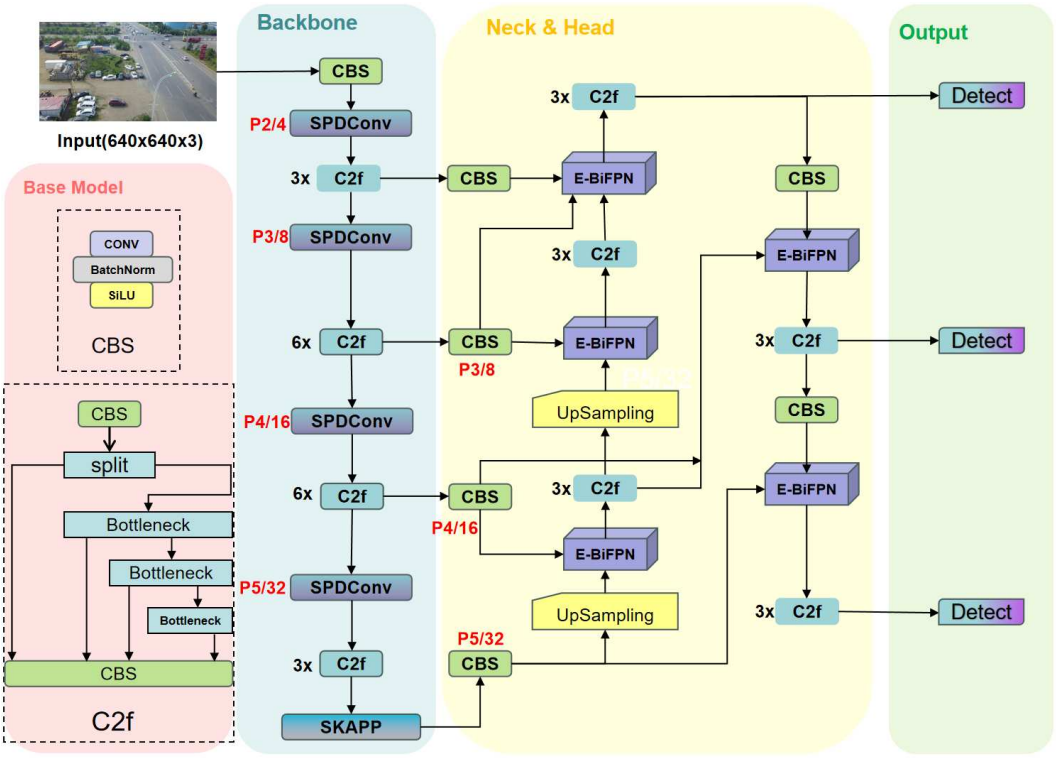


Figure 2. The overall architecture of the proposed SSE-YOLO is only provided for the **CBS** and **C2f** basic modules in the figure, with diagrams of the remaining modules to be added later. The overall architecture is yet to be fully depicted.

3.1. Backbone Improvements

3.1.1. Space-to-Depth Convolution

Convolutional neural networks (CNNs) have demonstrated remarkable success in various computer vision tasks. However, their performance significantly diminishes when confronted with the challenge of detecting low-resolution or small-sized objects. This limitation is attributed to an inherent flaw in the existing CNN architecture, specifically the utilization of stride convolution or pooling layers. This architectural choice diminishes the model's feature extraction capability, leading to challenges in handling fine-grained information loss. To address this issue, we have opted to integrate SPDConv[18], a novel CNN building block. As shown in **Figure 3**, this module comprises a space-to-depth (SPD) layer and a non-strided convolution layer, strategically replacing each strided convolution layer and pooling layer in the architecture. This innovative approach aims to enhance the model's ability to extract features, mitigating the difficulties associated with fine-grained information loss.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1108.v1

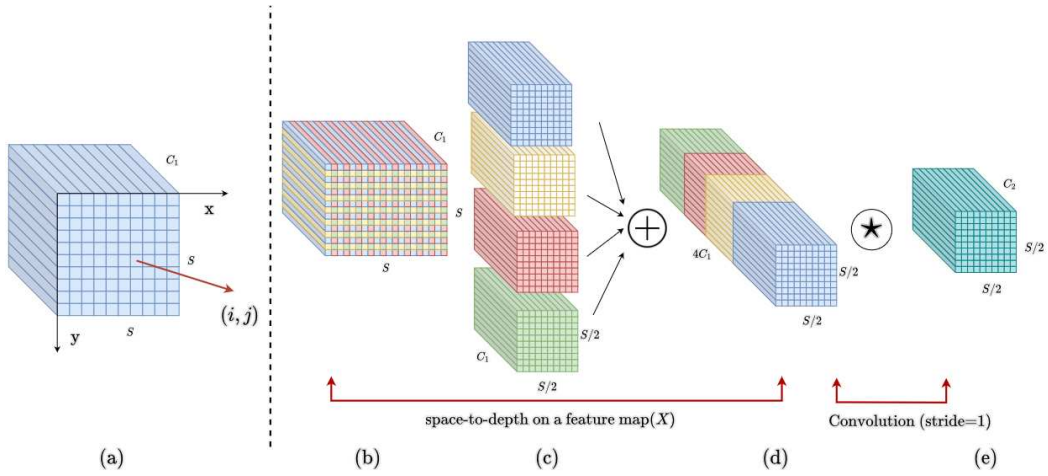


Figure 3. (a)(b)(c) give an example when scale = 2, where we obtain four sub-maps $f_{0,0}$, $f_{1,0}$, $f_{0,1}$, $f_{1,1}$ each of which is of shape $\left(\frac{S}{2}, \frac{S}{2}, C_1\right)$ and downsamples X by a factor of 2.

The SPD layer downscales the feature map X while preserving all information in the channel dimension, ensuring no loss of information. Drawing inspiration from image transformation techniques that re-scale the original image before inputting it into a neural network, this concept is extended to downsampling feature maps, both internally and across the entire network. Additionally, a non-spanning convolution operation is introduced after each SPD to adjust the number of channels through learnable parameters in the newly added convolutional layer, allowing for channel reduction or expansion. Consider any intermediate feature map X of size S × S × C1, slice out a sequence of sub feature maps as Equation (1). In summary, SPD transforms feature maps $X(S, S, C_1)$ into intermediate feature maps $X'\left(\frac{S}{scale}, \frac{S}{scale}, sacle^2 C_1\right)$.

$$
\begin{aligned}
f_{0,0} = X[0{:}S{:}scale, 0{:}S{:}scale], f_{1,0} = X[1{:}S{:}scale, 0{:}S{:}scale], \ldots, \\
f_{scale-1,0} = X[scale-1{:}S{:}scale, 0{:}S{:}sacle]; \\
f_{0,1} = X[0{:}S{:}scale, 1{:}S{:}scale], f_{1,1}, \ldots, \\
f_{scale-1,1} = X[scale-1{:}S{:}scale, 1{:}S{:}sacle]; \qquad (1) \\
\ldots \\
f_{0,scale-1} = X[0{:}S{:}scale, scale-1{:}S{:}scale], f_{1,scale-1}, \ldots, \\
f_{scale-1,scale-1} = X[scale-1{:}S{:}scale, sacle-1{:}S{:}scale].
\end{aligned}
$$

3.1.2. Separate Kernel Attention Pyramid Pooling

Jocher et al.[13] enhanced the Spatial Pyramid Pooling-Fast (SPPF) module in the benchmark model, building upon the structure of the Spatial Pyramid Pooling (SPP) module. This modified module incorporates three consecutive pooling layers, combining the output from each layer to ensure multi-scale fusion. Simultaneously, it reduces computational complexity and significantly improves speed compared to SPP, to achieve an adaptive size output. However, our investigation revealed that this structure struggles to integrate small and fine-grained feature information effectively. Despite reducing parameters through continuous pooling operations, it tends to overlook small and fine-grained information, deviating from our research goals. Consequently, we abandoned the original SPPF structure and proposed the SKAPP structure as a replacement, whose structure is shown in **Figure 4**. Compared with the original structure, our structure can greatly enhance the processing capability of small and fine-grained feature information, and can well integrate multi-dimensional feature information after downsampling by SPDConv.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1108.v1

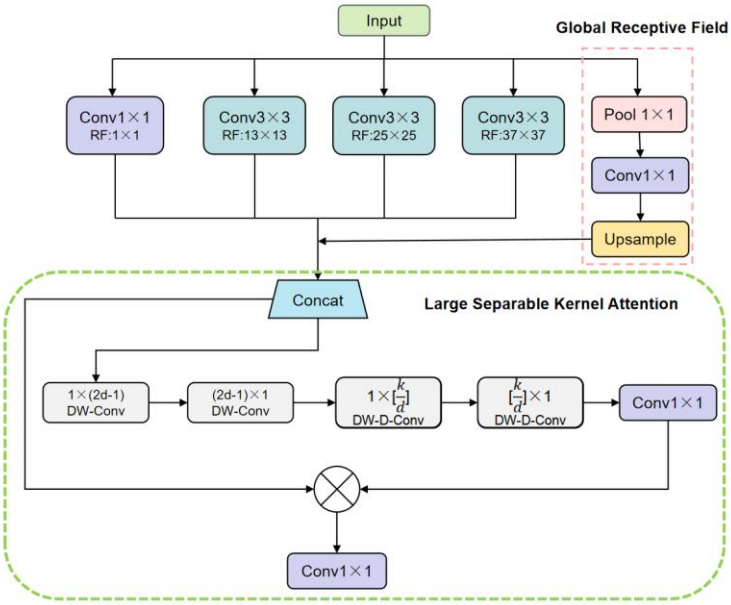


Figure 4. Overall structure of the SKAPP module. Notice that $\otimes$represents Hadamard product, k represents the maximum receptive field, and d represents the dilation rate.

We utilize multiple parallel atrous convolutional layers with varying sampling rates. The features extracted for each sampling rate undergo further processing in separate branches before being fused to generate the final result. This module constructs convolution kernels with different receptive fields by employing various hole rates to acquire multi-scale object information. Additionally, we integrate the pooling layer, convolution layer, and upsampling layer into a global receptive field. Enlarging the global receptive field helps the module avoid information loss and capture broader contextual information, including content in the image far away from the target area. This capability is crucial for understanding the structure of the entire image. Following these operations, we pass the fused feature information to the LSKA block as an input stream, referring to the Large Separable Kernel Attention (LSKA) module proposed by Lau et al.[32]. This block module utilizes a residual structure composed of four Depthwise Separable Convolution blocks and a Pointwise Convolution block with a convolution kernel size of 1. In the depth convolution stage, a separate convolution operation is performed on each channel of the input. If the input has C channels, there will be C convolution kernels, each responsible for processing information from one input channel. This step employs atrous convolution (also known as dilated convolution), enabling the convolution kernel to share parameters between input channels and thereby reducing the number of parameters. In the point-wise convolution stage, a 1x1 convolution kernel is used to linearly combine the output channels of the depth convolution. This step facilitates the interaction and combination between channels, integrating independently processed channel information to form the final output feature map. By processing spatial and channel information separately, depthwise separable convolution reduces computation, particularly in resource-limited scenarios like mobile devices. The decomposition of depth-wise convolution and point-wise convolution results in fewer parameters compared to standard convolution, reducing the model's complexity. This convolutional structure is well-suited for lightweight model design, especially in environments with limited computing and memory resources, such as drone-embedded devices.

The LSKA output is shown in Equation 2-5, and we will explain in detail all the parameters in the formula as well as the calculation symbols. * and $\otimes$ stand for convolution and Hadamard product respectively:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1108.v1

$$X^C = \sum_{H,W} W^C_{(2d-1)\times 1} * \left(\sum_{H,W} W^C_{1\times(2d-1)} * F^C \right) \quad (2)$$

$$Z^C = \sum_{H,W} W^C_{\left[\frac{k}{d}\right]\times 1} * \left(\sum_{H,W} W^C_{1\times\left[\frac{k}{d}\right]} * X^C \right) \quad (3)$$

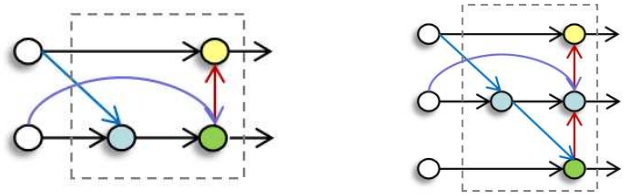
$$A^C = W_{1\times 1} * Z^C \quad (4)$$

$$T^C = A^C \otimes F^C \quad (5)$$

where Z^C is the output of the depth convolution obtained by convolving a kernel W of size k × k with the input feature map F. Note that each channel C in F is convolved with the corresponding channel in kernel W. The output $\bar{T}^C$ of LSKA is the Hadamard product of the attention map A^C and the input feature map F^C.

3.2. Neck and Head Improvements

To enhance multi-dimensional feature fusion during the upsampling process, we opted to eliminate the Concat module in the neck and head, introducing E-BiFPN as its replacement. This improvement is derived from the BiFPN concept initially proposed by Tan et al.[20]. BiFPN incorporates learnable weights to discern the importance of different input features. Structurally, it first eliminates nodes with only one input. Secondly, if the original input and output nodes are at the same level, an additional edge is added between them to encompass more features without incurring additional costs. Finally, each bidirectional path (top-down + down-top) is treated as a feature network layer, and this process is iterated multiple times to facilitate higher-level feature fusion.



(a) **(b)**

Figure 5. There are two structures of E-BIPFN in the model. (a)Two input heads; (b)Three input heads. The dotted line indicates a repeat block.

In the feature fusion process, re-sizing is necessary due to different resolutions. However, since different feature input resolutions vary, the output's contribution to the final feature network should also differ. Consequently, the network needs to learn these weights. We proposed three weighting schemes, as illustrated in Equations 6-8, but we chose to adopt the weighting method from fast normalized feature fusion:

$$O_n = \sum_i w_i I_i \quad (6)$$

$$O_S = \sum_i \frac{e^{w_i}}{\sum_j^e w_j} \quad (7)$$

$$O_q = \sum_i \frac{w_i}{\epsilon + \sum_j^{w_j} 1} I_i \quad (8)$$

where equation (6) represents a generally weighted feature fusion but without constraints on wi, it can lead to convergence difficulties, which do not align with our requirements. Equation (7)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1108.v1

represents feature fusion using Softmax. Although it achieves convergence, the use of numerous exponential operations makes it inefficient and does not meet our requirements. Equation (8) represents fast normalized feature fusion, a weighting method that employs ReLU to constrain w_i and sets $\epsilon = 10^{-4}$ to ensure data stability and avoid Softmax operations. In summary, our E-BiFPN forms the final weighted bidirectional feature pyramid network based on bidirectional cross-scale connections and fast normalized feature fusion.

3.3. One-way Feature Transfer Pyramid

Different PAN[34] network structure than the baseline model, we implement a novel structure known as the One-way Feature Transfer Pyramid (OFTP), as shown in **Figure 6**. Between the backbone and the neck, we introduce four CBS modules with a step size and convolution kernel size of 1 as containers to store backbone feature information. Utilizing the concept of path aggregation, we aggregate shallow feature maps (with low resolution but weak semantic information) and deep feature maps (with high resolution but rich semantic information). Feature information is transferred along specific paths, allowing the strong localization features of the lower layer to be passed upward. These operations further enhance the expressive capabilities of multi-scale features. Building upon this, we use these four containers as the new backbone feature information input stream for the subsequent network, thereby improving the performance of the OFTP structure in detecting low-pixel small targets. However, this structure undoubtedly imposes an additional computational burden and complexity on subsequent networks in terms of multi-dimensional feature processing. Therefore, we pass these data streams to the E-BiFPN module, which replaces the Concat module, cleverly addressing the requirements for multi-dimensional feature fusion and reducing computational load.

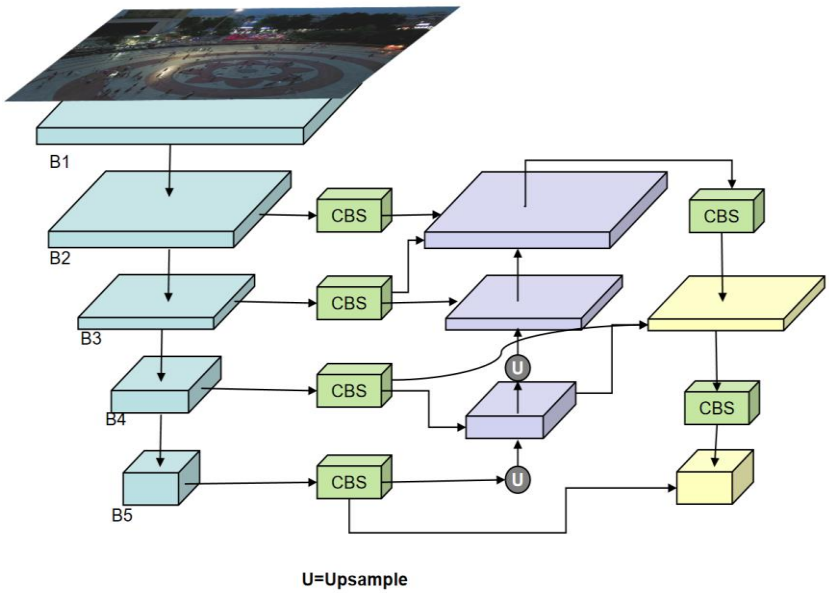


Figure 6. Four CBS modules are inserted between the backbone and neck to store feature information from top to bottom.

4. Experiments and Results

This section will provide an overview of the experimental dataset, the experimental details ,and the experimental evaluation metrics adopted in turn. We will provide a comprehensive analysis and summary of the experimental results, and present the complete and real experimental data in graphs to prove the superiority of the performance of the proposed model. First, the experimental dataset used is presented in detail.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1108.v1

4.1. Datasets and Experimental Environment

Given the markedly distinct characteristics of UAV aerial images compared to those captured by ground personnel, UAV image recognition tasks present a notably greater challenge than traditional image recognition. Standard image sets, such as MS COCO[35] and PASCALVOC[36], may not as suitable foundations for this article. Consequently, the VisDrone2019 dataset collected and annotated by the AISKYEYE team at Tianjin University's Machine Learning Data Mining Laboratory[37] has been adopted for this study. The dataset comprises 6471 images distributed across three categories: training set, verification set, and test set. The images capture different scenes consisting of pedestrians, motorcycles, and various models of cars, among ten other common object categories. Acknowledged as a benchmark in the international drone vision field, this dataset holds authoritative status. The images exhibit extensive diversity, encompassing various settings like streets, roads, residential areas, docks, and similar backgrounds. The VisDrone2019 dataset encompasses a variety of light conditions such as excessive light, sufficient light, insufficient light, cloudy and night. Furthermore, the dataset features drone images with complex backgrounds, both large and small scenes, and various intricate elements. The rich diversity of these features underscores the necessity for a model with robust generalization and feature extraction capabilities.

Our baseline model is YOLOv8, version Ultralytics 8.0.225. In terms of hardware and software, we used an Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50GHz, 16 cores, and 24 threads, a main frequency of 3.19 GHz, 32 GB running memory, graphics processor GeForce RTX 3090, and 24 GB video memory; the deep learning model framework used Python 3.8、Pytorch2.0.0 and CUDA 11.8.

To ensure the fairness of the comparison and ablation experiments, identical hyper-parameters have been employed in both the training and testing phases across all experiments. The crucial hyper-parameter configurations during the training process are detailed in **Table 1**. Notably, Mosaic, Translation, and Scale are employed as data enhancement methods in image processing. Maintaining uniformity in hyperparameter settings is critical to preserving the integrity of the comparison and ensuring accurate evaluation in ablation experiments.

Table 1. Training hyper-parameter setting table.

Hyper-parameters	Setup
Epochs	100
Batch Size	16
Input Size	640
Optimizer	SGD
NMS IoU	0.7
Initial Learning Rate	1×10^{-2}
Final Learning Rate	1×10^{-4}
Weight-Decay	5×10^{-4}
Workers	8
Mosaic	1.0
Translation	0.1
Scale	0.5
Momentum	0.937
Close Mosaic	10
Warmup Epochs	2

4.2. Experiment Metrics

In this paper, we will employ precision (P) evaluation metric with an IoU threshold of 0.7 along with average accuracy mAP50 and mAP90 with an IoU threshold of 0.7 to assess the accuracy performance of our proposed method for identification purposes. Additionally, to assess the model's speed and computing performance, we will utilize Frames Per Second (FPS) to measure the number of frames processed per second, Giga Floating Point Operations Per Second (GFLOPs) to measure the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1108.v1

billions of executed floating-point operations per second and the parameter size (M) of the model, which refers to the size of the parameters in millions of parameters.

Precision (P) assesses the overall performance strength of the model by measuring the proportion of correctly predicted targets among all predicted targets. It is primarily computed using TP and FP, whereby TP denotes the accurately predicted target and FP denotes the inaccurately predicted target. The Equation (9) describes the specific calculation.

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

Recall(R) is the ratio of correctly identified targets to all targets. Its computation is similar to that of precision and can be determined using the specific formula depicted in Equation (10) where FN stands for targets that exist within the dataset but have not been detected.

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

Average precision (AP) is the region encapsulated by the curve generated by precision and recall. A higher value of AP indicates a larger area. Equation (11) depicts its methodology. We utilize two distinct kinds of average accuracy metrics: mAP0.5 and mAP0.95. Here, 0.5 and 0.95 represent the union size between predicted and annotated bounding boxes. For bounding box predictions to be considered accurate, IoU (Intersection over Union) rates must be high. At 0.50 or 0.95, this technique can more comprehensively evaluate the accuracy performance of the model.

$$\mathrm{AP} = \int_0^1 p(r)dr \tag{11}$$

The mean precision (mAP) indicates the average accuracy across all image sample types, calculated using Equation (12). We have adopted two varying average accuracy metrics: mAP0.5 and mAP0.95.

$$\mathrm{mAP} = \frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}} AP_i \tag{12}$$

4.3. Ablation Experiment

To assess the impact of the added and modified modules on the baseline model's performance, we will disassemble the structure of each module and conduct comprehensive ablation experiments for evaluation and comparison. It is worth noting that YOLOv8 and the improved model in this table default to nano. For convenience, we will no longer use YOLOv8-n and SSE-YOLO-n in the table and subsequent statements, opting to use YOLOv8 and SSE-YOLO directly. **E**-YOLO uses only the E-BIFPN structure to modify the model, while $\mathbf{S_P}$-YOLO replaces Conv with SPDConv. By contrast, **SS**-YOLO employs both SPDConv and SKAPP for modification. The $\mathbf{S_PE}$-YOLO model is adjusted using SPDConv and E-BIFPN, while $\mathbf{S_KE}$-YOLO uses the modified SKAPP and E-BIFPN frameworks. In $\mathbf{S_K}$-YOLO, only the SKAPP framework is used. No training files were used to ensure the experiment's fairness. All hyper-parameters implemented in the experiment are consistent, and specific parameters are outlined in **Table 1.** The final results of the experimental data are shown in **Table 2**.

Table 2 presents the experimental evaluation outcomes of SSE-YOLO-n and YOLOv8-n on the VisDrone2019 dataset. This represents an enormous enhancement. The top-performing results in each indicator have been highlighted. Clearly, SSE-YOLO-n demonstrated superior performance in accuracy-related evaluations, including mAP and Precision. Our model improved $mAP_{0.5}$ by 10.18% and $mAP_{0.95}$ by 7.25% from the baseline model, and its Precision also increased by a significant 9.43%. SSE-YOLO-n is not the most optimal choice in terms of FPS and Parameters evaluation metrics. Specifically, its weak real-time processing capability when evaluating FPS is a drawback. In addition, SSE-YOLO-n has a 12% increase in parameters compared to baseline model. However, it is worth

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1108.v1

noting that both models still maintain a similar number of parameters, so this does not necessarily classify as a flaw.

Table 2. Ablation experimental data shows that each result obtained is optimal.

Model	m$AP_{0.5}$(%)	m$AP_{0.95}$(%)	Precision(%)	FPS	GFLOPs	Parameters(M)
YOLOv8-n[16]	30.62	17.25	41.07	113.7	8.2	3.15
E-YOLO-n	32.0	18.3	42.4	153.3	**7.2**	**1.99**
$\mathbf{S_P}$-YOLO-n	33.1	19.3	42.7	**200.3**	10.9	4.18
$\mathbf{S_K}$-YOLO-n	34.0	19.9	46.1	192.2	8.4	3.28
SS-YOLO-n	34.8	20.3	47.7	140.4	11.2	2.65
$\mathbf{S_P}$**E**-YOLO-n	31.6	18.1	43.1	162.0	10.1	3.16
$\mathbf{S_K}$**E**-YOLO-n	31.9	18.4	43.5	159.2	7.3	2.26
SSE-YOLO-n	**40.8**(↑10.18%)	**24.5**(↑7.25%)	**50.5**(↑9.43%)	135.0	10.9	3.55

4.4. Performance Comparison with State-of-the-Art Methods and Baseline Methods

In order to comprehensively evaluate the performance of SSE-YOLO, its experimental results were comprehensively compared with those of 10 other state-of-the-art methods on the VisDrone2019 dataset. The experimental results indicate that the SSE-YOLO series is an excellent model for multi-target detection in UAV images due to its small parameters and high accuracy.

As shown in **Table 3**, our SSE-YOLO achieved 53.2% m $AP_{0.5}$ and 34.3% m $AP_{0.95}$ on VisDrone2019-DET-test-dev, these two evaluation index values are the highest among all compared models. Based on the indicators, TPH-YOLOv5++ has achieved excellent experimental results. Its m $AP_{0.5}$ and m $AP_{0.95}$ are only 0.7% and 0.8% lower than our model. However, our method outperforms TPH-YOLOv5++ in terms of GFLOPs and Parameters. SSE-YOLO (large) has fewer parameters than TPH-YOLOv5++ by 44.5% and also reduces GFLOPs by 25.1%.

However, it is worth noting that SSE-YOLO's performance in Frames Per Second (FPS) is not optimal, with SSE-YOLO (large) having the lowest FPS. However, it is worth noting that SSE-YOLO's performance in Frames Per Second (FPS) is not optimal, with SSE-YOLO (large) having the lowest FPS. It is important to compare FPS with other models. SSE-YOLO (large) is more accurate than YOLOv8-s and YOLOv8-l in the 40~60 range. Based on the FPS comparison, YOLOv5-n performs the best. SSE-YOLO (nano) is 25.2 lower than YOLOv5-n in this indicator data. However, our model outperforms YOLOv5-n in terms of m and m, with 13.8% and 12% higher scores respectively. This represents a significant improvement.

In conclusion, our model considers both small parameter size and high accuracy, albeit at the expense of a small reduction in FPS. This fulfills the two most crucial requirements of the UAV-embedded platform. Consequently, our model is better suited for multi-target recognition tasks in high-precision UAV images, and it presents a novel solution in this field.

Table 3. Comparison of experimental results with ten other state-of-the-art methods on VisDrone2019-DET-test-dev. The best results are shown in bold.

Model	m$AP_{0.5}$(%)	m$AP_{0.95}$(%)	FLOPs(G)	Parameters(M)	FPS
YOLOv5-n	32.4	18.9	**7.2**	**2.65**	**188.1**
YOLOv5-s	34.3	20.1	24.1	9.15	154.2
YOLOv7-tiny[15]	26.6	17.8	-	6.02	-
YOLOv8-s[16]	30.9	17.25	28.7	11.1	58.9
YOLOv8-l[16]	35.9	21.2	165.7	76.7	50.9
Faster R-CNN[38]	31.0	17.2	118.8	41.2	-
RetinaNet [39]	44.3	22.7	35.7	36.4	-
Drone-YOLO(large)[40]	40.7	23.8	-	76.2	-
Modified-YOLOv8[41]	33.7	-	-	9.66	143

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1108.v1

TPH-YOLOv5++[42]	52.5	33.5	207.0	99.1	-
SSE-YOLO(nano)	46.2	30.9	10.2	3.44	162.9
SSE-YOLO(small)	48.7	32.8	37.2	13.1	128.9
SSE-YOLO(medium)	50.2	34.0	96.0	27.5	69.1
SSE-YOLO(large)	**53.2**	**34.3**	193.2	44.1	44.0

The **Table 4** presents the evaluation index data of the proposed method and 10 comparative models, which were experimentally obtained on the VisDrone2019-val dataset. As GFLOPs are missing for most of the compared models in this table, we have excluded this evaluation metric from the table. It is evident that SSE-YOLO (large) achieved the best results in both $mAP_{0.5}$ and $mAP_{0.95}$. In comparison to the benchmark model YOLOv8-l, $mAP_{0.5}$ increased by 13.9%, $mAP_{0.95}$ increased by 14.2%, and the number of parameters is only 57.5% of YOLOv8-l. Meanwhile, although YOLOv5-n has the fewest parameters, SSE-YOLO (nano) has achieved a 4.9% increase in $mAP_{0.5}$ and a 3.4% increase in $mAP_{0.95}$ compared to YOLOv5, with only 0.79M more parameters than YOLOv5-n.

Table 4. Comparison of experimental results with other advanced methods on the VisDrone2019-DET-val dataset. The best results are highlighted in bold.

Model	$mAP_{0.5}$(%)	$mAP_{0.95}$(%)	Parameters(M)	FPS
YOLOv5-n	46.2	32.3	**2.65**	**188.1**
YOLOv5-s	49.4	35.1	9.12	154.2
YOLOv7-l[15]	47.1	26.4	71.4	-
YOLOv8-l[16]	43.7	26.9	76.7	50.9
Drone-YOLO (large)[40]	51.3	33.2	76.2	-
Modified-YOLOv8[41]	42.2	-	9.66	143
ACAM-YOLO[43]	49.5	29.6	15.9	-
MS-YOLOv7[44]	53.1	31.3	79.7	-
EdgeYOLO[45]	44.8	26.4	40.5	34
SSE-YOLO(nano)	50.1	35.7	3.44	162.9
SSE-YOLO(small)	53.5	38.1	13.1	128.9
SSE-YOLO(medium)	56.7	40.8	27.5	69.1
SSE-YOLO(large)	**57.6**	**41.1**	44.1	44.0

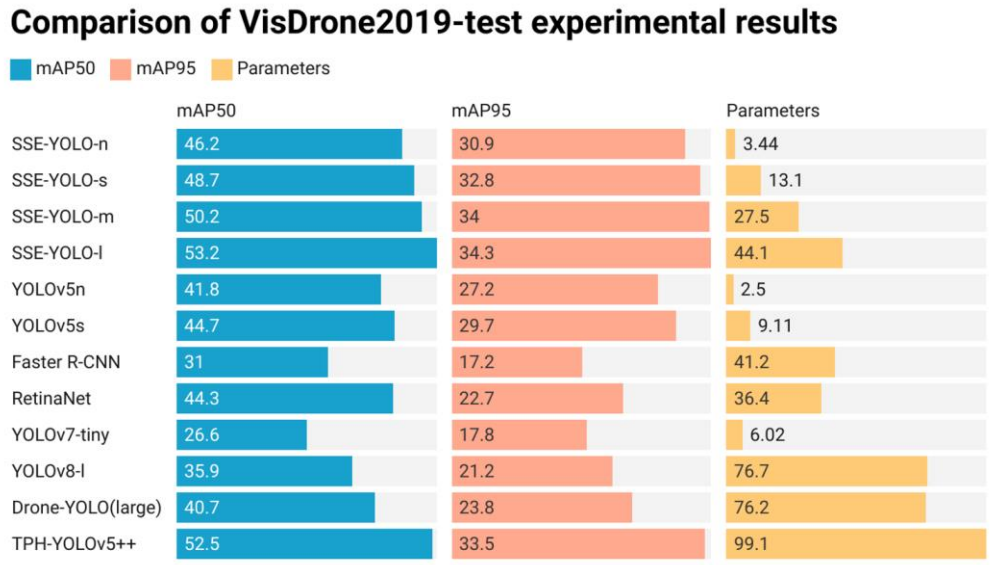
5. Visualization and discussion of experimental data

Deep learning is often referred to as a 'black box'. Despite the widespread use of deep learning models in various engineering fields, their lack of interpret-ability has hindered their progress in some high-tech fields[40]. Therefore, interpret-ability of deep learning has become a mainstream research direction in contemporary artificial intelligence. Drones play a crucial role in various industries, including smart agriculture and the military, making their interpret-ability a key factor in building their models. This section will use visualization methods to provide a clear explanation and summary of the experimental data. Firstly, we will select the experimental results obtained by several typical models on the test and verification sets as the data for the visual chart, as shown in **Figure 7**.

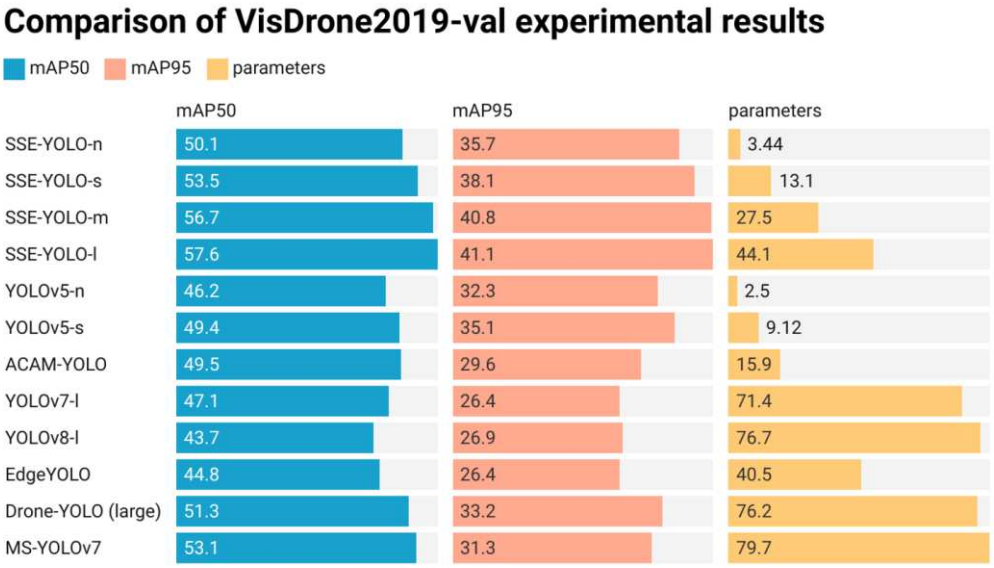
The two visualization tables in **Figure 7** clearly demonstrate that our method achieves remarkably high accuracy with a minimal number of parameters. Notably, SSE-YOLO-n boasts the lowest parameter count while outperforming most comparison models in both mAPs. Simultaneously, SSE-YOLO-l achieves accuracy comparable to TPH-YOLOv5++ and MS-YOLOv7, yet with significantly fewer parameters. This underscores our model's excellence in balancing accuracy and parameter control, yielding outstanding results. Nevertheless, these two visualization diagrams fall short of fully showcasing the superiority of our method, particularly in terms of recognition accuracy for small targets. To provide a more comprehensive demonstration, we will compare the confusion matrix diagrams of several experimental models with our method. This aims

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1108.v1

to offer a more intuitive representation of the enhancement in small target recognition accuracy achieved by our approach.



(a)



(b)

Figure 7. Data visualization chart of experimental results:(a)Experimental results of the example model on VisDrone2019-test;(b)Experimental results of the example model in VisDrone2019-val.

The confusion matrix derived from the experimental results reveals a significant number of omissions for all three models, suggesting confusion between the recognition target and the background category. To better illustrate the enhancement in our model's accuracy for small targets, we will specifically select several representative small targets as parameters for comparison. The comparative results will be visually presented to showcase the performance of our model. Please refer to **Figure 8** for the results. In particular, these lightweight models have demonstrated exceptional results in recognizing large targets, with car types serving as a notable reference. It is not difficult to see that all four models have high recognition accuracy for this type of recognition. Therefore, the 6% improvement of our model over YOLOv8-s may not be deemed a particularly significant enhancement in this context. However, our model has achieved a great improvement in recognition accuracy for small targets. Given the inherently lower accuracy, relying solely on

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1108.v1

percentage improvement is inappropriate, so we will use the SSE-YOLO-s divided by the YOLOv8-s method to better show the extent of the improvement. Our model achieved 154.5%, 200%, 110.7%, 121.4%, and 131.6% of the accuracy of the benchmark model in the categories of person, bicycle, car, tricycle, awning-tricycle, and motor vehicle, respectively. The experimental results underscore the substantial progress our model has made in the task of detecting small target types.

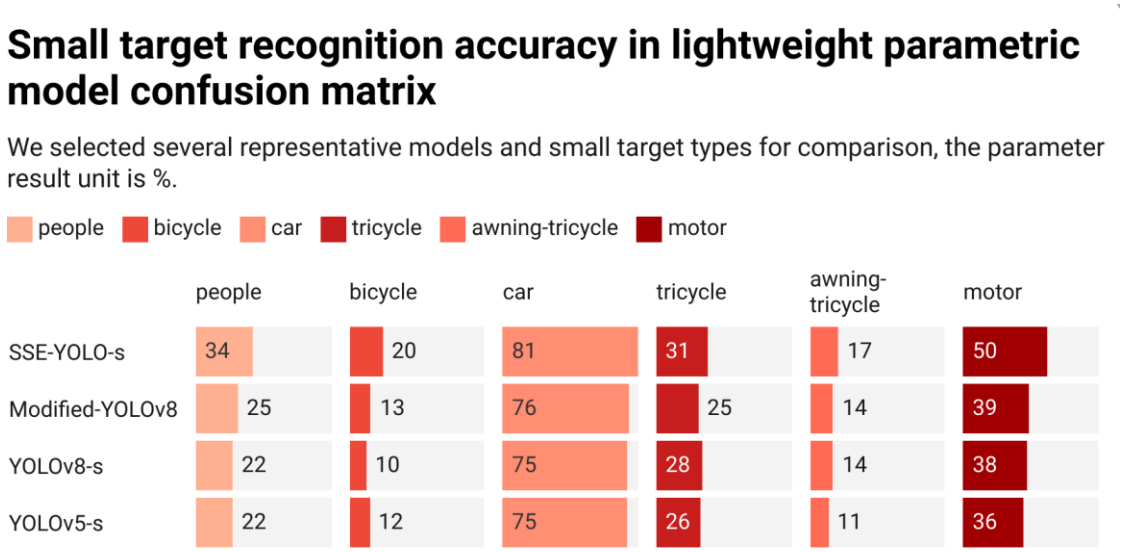


Figure 8. Confusion matrix plot of selected three models. (**a**)YOLOv5-s confusion matrix diagram. (**b**)Li et al. confusion matrix diagram[41]. (**c**)Confusion matrix diagram of the SSE-YOLO-s in this paper.

6. Conclusions

This paper proposes SSE-YOLO, which aims to address the shortcomings of UAV target recognition by combining low parameter volume and high accuracy to identify small targets and complex backgrounds. Utilizing deep learning techniques, this model adeptly mitigates the adverse effects of diverse factors, encompassing background complexities, visibility issues, and scale variations, in UAV detection tasks. Firstly, we introduced SPDConv, a novel convolutional method designed for low resolution, replacing the standard Conv in the baseline model's backbone. This modification addresses the loss of fine-grained information in the original module and enhances the accuracy of extracting tiny targets, effectively mitigating these inherent flaws. In a second improvement, we incorporated the concept of LSKA at the bottom of the backbone section and introduced a novel structure named SKAPP to substitute the SPPF in the baseline model. This innovation achieves a more sophisticated feature fusion approach by thoroughly considering and reusing multi-scale features. It significantly enhances the feature fusion capabilities of the backbone and contributes to the advancement of multi-dimensional feature extraction in the neck and head segments. As a third enhancement, we addressed the issue of the baseline model's tendency to lose contextual feature information. To overcome this, we incorporated four CBS modules with a convolution kernel size step number of 1 in both the backbone and neck sections to retain crucial feature information. This preserved information is subsequently forwarded to the next network for processing. As a final improvement, we introduced the E-BIFPN module, inspired by the BIFPN concept, to replace the Concat module in the Neck & Head segment. This enhancement employs a deep network capable of adapting to various input heads, enabling more effective multi-dimensional feature fusion. It is worth noting that the above four improvements have reduced the number of calculation parameters while solving different problems. Therefore, our model can have high accuracy with a small number of parameters.

In experiments on the VisDrone2019 dataset, our method demonstrated superior performance. In both the test and validation datasets, SSE-YOLO (large) outperformed other comparison models in $mAP_{0.5}$ and $mAP_{0.95}$ metrics, with its parameter size being only 57.5% of the benchmark model YOLOv8-l. The smaller model, SSE-YOLO (nano), with a parameter size of only 3.44M, outperformed RetinaNet (36.4M parameters) and ACAM-YOLO (15.9M parameters) in $mAP_{0.5}$. This signifies the remarkable success and applicability of our model, making a significant contribution to the advancement of UAV multi-target recognition.

While our method excels in high-precision identification and parameter control, it currently falls short in terms of real-time image processing capability. Future research will focus on enhancing the model's real-time processing speed without compromising accuracy. To address the challenge of weak real-time processing capabilities, our future research will delve into PP-YOLOE[46]. We aim to explore this model's fundamental ideas and methods to enhance computational speed and integrate them into our model. Additionally, we plan to experiment with incorporating BiFormer[47] dynamic sparse attention mechanism in our future work to reduce computational and memory loads further. We aim to achieve a faster and more precise real-time target recognition algorithm. Additionally, we plan to explore image enhancement and lightweight strategies to bolster multi-dimensional feature extraction capabilities. To broaden the application scope, we are working on labeling ground cracks, fires, and other types of natural geographical disasters in custom datasets to further train our method. Our aspiration is for this approach to find application in a wider range of areas, providing increased value to the field.

Author Contributions: Conceptualization: M.S. and Y.L., methodology: M.S. and Y.L., formal analysis: M.S. and Y.L., investigation: M.S. and Y.L., data curation: M.S., writing– original draft preparation: M.S., writing–review and editing: M.S. and Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 62176273).

Data Availability Statement: We will place the experimental data results in M.S. ***GitHub*** library.

Acknowledgments: We would like to thank the authors of the baseline model, the authors of the comparison methods, and the authors of the articles who provided us with ideas for improvement. We express our deepest gratitude to the reviewers and the editor for their careful work and thoughtful suggestions, which greatly improved this paper and raised the standard and rigor of the entire article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wu, X.; Li, W.; Hong, D.; Tao, R.; & Du, Q. (2021). Deep learning for unmanned aerial vehicle-based object detection and tracking: A survey. IEEE Geoscience and Remote Sensing Magazine, 10(1), 91-124.
2. Li, W.; Chen, Y.; Hu, K.; & Zhu, J. (2022). Oriented reppoints for aerial object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 1829-1838).
3. Li, F.; Zeng, A.; Liu, S.; Zhang, H.; Li, H.; Zhang, L.; Ni, L.M. Lite DETR: An interleaved multi-scale encoder for efficient detr. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 18558–18567.
4. Gu, J.; Su; T., Wang, Q.; Du, X.; & Guizani, M. (2018). Multiple moving targets surveillance based on a cooperative network for multi-UAV. IEEE Communications Magazine, 56(4), 82-89.
5. Du, B.; Huang, Y.; Chen, J.; & Huang, D. (2023). Adaptive Sparse Convolutional Networks with Global Context Enhancement for Faster Object Detection on Drone Images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 13435-13444).
6. Cai, W.; & Wei, Z. (2020). Remote sensing image classification based on a cross-attention mechanism and graph convolution. IEEE Geoscience and Remote Sensing Letters, 19, 1-5.
7. Domozi, Z.; Stojcsics, D.; Benhamida, A.; Kozlovszky, M.; Molnar, A. Real Time Object Detection for Aerial Search and Rescue Missions for Missing Persons. In Proceedings of the 2020 IEEE 15th International Conference of System of Systems Engineering (SoSE), Budapest, Hungary, 2–4 June 2020; pp. 519–524.
8. Kirchheim, K.; Konstantin, K.; Tim Gonschorek, T.; and Frank Ortmeier F., O.; "Addressing randomness in evaluation protocols for out-of-distribution detection." arXiv preprint arXiv:2203.00382 (2022).
9. Redmon, J.; Divvala, S.; Girshick, R.; & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

10. Redmon, J.; Farhadi, A.; (2017). YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).
11. Redmon, R.; Joseph, J.; Ali Farhadi, A. F. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018).
12. Bochkovskiy, A.; Wang, C. Y.; Liao, H. Y. M. Yolov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv:2004.10934, 2020.
13. Jocher, G.; Stoken, A.; Chaurasia, A.; Borovec, J.; Kwon, Y.; Michael, K.; Changyu, L.; Fang, J.; Skalski, P.; Hogan, A.; et al. Ultralytics/Yolov5: V6.0—YOLOv5n 'Nano' Models, Roboflow Integration, TensorFlow Export, OpenCV DNN Support, 2021. Available online: https://zenodo.org/record/5563715 (accessed on 30 June 2023).
14. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; ... & Wei, X. (2022). YOLOv6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976.
15. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, Canada, 18–22 June 2023; pp. 7464–7475.
16. Jocher, G.; Chaurasia, A.; Qiu, J. YOLO by Ultralytics. 2023. Available online: https://github.com/ultralytics/ultralytics/blob/ main/CITATION.cff (accessed on 30 June 2023).
17. Zhu, L.; Xiong, J.; Xiong, F.; Hu, H.; Jiang, Z. YOLO-Drone: Airborne real-time detection of dense small objects from high-altitude perspective. arXiv 2023, arXiv:2304.06925.
18. Yang, Z.; Liu, S.; Hu, H.; Wang, L.; Lin, S. Reppoints: Point set representation for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October – 2 November 2019; pp. 9657 – 9666.
19. Du, D.; Qi, Y.; Yu, H.; Yang, Y.; Duan, K.; Li, G.; Zhang, W.; Huang, Q.; Tian, Q. The unmanned aerial vehicle benchmark: Object detection and tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8 – 14 September 2018; pp. 370 – 386.
20. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
21. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; Volume 25.
22. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv 2014, arXiv:1409.1556.
23. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.
24. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; ... & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 10012-10022).
25. Mehta, S.; Rastegari, M.; Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer[J]. arXiv preprint arXiv:2110.02178, 2021.
26. Yan, B.; Fan, P.; Lei, X.; Liu, Z.; & Yang, F. (2021). A real-time apple targets detection method for picking robot based on improved YOLOv5. Remote Sensing, 13(9), 1619.
27. Cao, J.; Bao, W.; Shang, H.; Yuan, M.; & Cheng, Q. (2023). GCL-YOLO: A GhostConv-Based Lightweight YOLO Network for UAV Small Object Detection. Remote Sensing, 15(20), 4932.
28. Zhu, X.; Lyu, S.; Wang, X.; & Zhao, Q. (2021). TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 2778-2788).
29. Wang, G.; Chen, Y.; An, P., Hong; H., Hu, J.; & Huang, T. (2023). UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios. Sensors, 23(16), 7190.
30. Wang, Y.; Zou, H.; Yin, M.; & Zhang, X. (2023). SMFF-YOLO: A Scale-Adaptive YOLO Algorithm with Multi-Level Feature Fusion for Object Detection in UAV Scenes. Remote Sensing, 15(18), 4580.
31. Sunkara, R.; & Luo, T. (2022, September). No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 443-459). Cham: Springer Nature Switzerland.
32. Lau, K. W.; Po, L. M.; & Rehman, Y. A. U. (2024). Large Separable Kernel Attention: Rethinking the Large Kernel Attention Design in CNN. Expert Systems with Applications, 236, 121352.
33. Tan, M.; Pang, R.; & Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10781-10790).
34. Liu, S.; Qi, L.; Qin, H.; Shi, J.; & Jia, J. (2018). Path aggregation network for instance segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8759-8768)..

35. Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll´ ar, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision, pages 740 – 755. Springer, 2014
36. Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis., 88(2):303–338, 2010.
37. Du, D.; Zhu, P.; Wen, L.; Bian, X.; Lin, H.; Hu, Q.; Peng, T.; Zheng, J.; Wang, X.; Zhang, Y.; et al. VisDrone-DET2019: The vision meets drone object detection in image challenge results. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea 27–28 October 2019.
38. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 28.
39. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
40. Zhang, Z. (2023). Drone-YOLO: an efficient neural network method for target detection in drone images. Drones, 7(8), 526.
41. Li, Y.; Fan, Q.; Huang, H.; et al. A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition[J]. Drones, 2023, 7(5): 304.
42. Zhao, Q.; Liu, B.; Lyu, S.; Wang, C.; Zhang, H. TPH-YOLOv5++: Boosting Object Detection on Drone-Captured Scenarios with Cross-Layer Asymmetric Transformer. Remote Sens. 2023, 15, 1687.
43. Li, Z.; Wang, Z.; He, Y. Aerial Photography Dense Small Target Detection Algorithm Based on Adaptive Collaborative Attention Mechanism. J. Aeronaut 2023, 44, 327944.
44. Zhao, L.; Zhu, M. MS-YOLOv7: YOLOv7 Based on Multi-Scale for Object Detection on UAV Aerial Photography. Drones 2023, 7, 188.
45. Liu, S.; Zha, J.; Sun, J.; Li, Z.; Wang, G. EdgeYOLO: An Edge-Real-Time Object Detector. arXiv 2023, arXiv:2302.07483.
46. Xu, S.; Wang, X.; Lv, W.; Chang, Q.; Cui, C.; Deng, K.; ... & Lai, B. (2022). PP-YOLOE: An evolved version of YOLO. arXiv preprint arXiv:2203.16250.
47. Zhu, L.; Wang, X.; Ke, Z.; Zhang, W.; & Lau, R. W. (2023). BiFormer: Vision Transformer with Bi-Level Routing Attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10323-10333).