IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 May 2024, accepted 11 June 2024, date of publication 17 June 2024, date of current version 24 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3415385

RESEARCH ARTICLE

SMB-YOLOv5: A Lightweight Airport Flying Bird Detection Algorithm Based on Deep Neural Networks

HAIJUN LIANG, XIANGWEI ZHANG, JIANGUO KONG, ZHIWEI ZHAO, AND KEXIN MA
College of Air Traffic Management, Civil Aviation Flight University of China, Guanghan 618307, China

Corresponding author: Haijun Liang (navyliang@cafuc.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFF0603904 and in part by the Fundamental Research Funds for the Central Universities under Grant PHD2023-035.

ABSTRACT Birds pose a serious threat to the safe operation of aircrafts. Existing object detection methods have achieved good results for big and medium instances; however, for small flying bird instances, drawbacks such as slow speed, low accuracy, and large model size are still present. Therefore, to overcome these shortcomings, we propose the SMB-YOLOv5 model to detect birds near airports. First, we introduce a self-supervised predictive convolution attention block to enable YOLOv5s6 to focus on critical information, thereby enhancing detection performance. Second, we introduce a multi-branch block (MBB) that enhances the expressive capability of the network by incorporating branches with different receptive fields. Third, to enhance the feature fusion capability of the model and detection mAP@50 for small-bird instances, drawing inspiration from the bidirectional feature pyramid network, we reutilize the shallow-level features of the feature extraction network. We also remove some modules to ensure an increased accuracy without excessively inflating the model size. Finally, to increase the convergence speed of the network, we modify its loss function by replacing complete IoU (CIoU) with efficient IoU (EIoU), which improves the detection mAP@50 of the network. Compared to the YOLOv5s6 model, the proposed SMB-YOLOv5 model achieves a 2.6% increase in mAP@50 on the test dataset. The detection speed has reached 24 fps. We find that the SMB-YOLOv5 has a higher mAP@50 than the other algorithms in the test dataset and the lowest number of parameters, and it can be applied in airport bird detection systems to provide more precise bird orientation information for airport bird detection tasks.

INDEX TERMS Bird detection, airport bird strike, YOLOv5s6, attention mechanism, feature fusion, efficient IoU loss.

I. INTRODUCTION

Bird strikes, also known as bird collisions, refer to instances in which aircrafts collide with birds during take-off, landing, or flight, leading to aviation safety concerns or accidents [1]. According to the Federal Aviation Administration (FAA) Wildlife Strike Database, which records wildlife strikes with aircrafts, birds are the primary wildlife involved in these strikes. Between 1990 and 2022, the database received reports of 276,846 bird strike incidents, with 272,016 of these occurring in the United States. The database also reveals a yearly increase in bird strike events. Bird strikes can have various negative impacts on flights, including potential engine damage, structural damage to the aircraft surface, wing deformation owing to substantial contact forces, and damage to cockpit windshields and sensors on the aircraft surface. Therefore, the accurate monitoring of birds near airport runways is crucial. Such monitoring can support decision-making in bird control at airports and ensure the safety of civil aviation operations.

Airport bird control methods usually include auditory deterrence, visual deterrence, killing, chemical bird control,

The associate editor coordinating the review of this manuscript and approving it for publication was Rosario Pecora.

radar bird control, drone-based bird control, and environmental and scientific bird control [2]. Recognizing bird targets and their locations before deploying bird control techniques is vital. Currently, two major approaches for detecting birds exist: radar-based and manual. However, computer vision technologies have recently provided fresh insights into bird recognition using visible light images or videos.

Object detection in computer vision involves the identification and classification of various objects present in an image [3]. Object detection methods can be categorized into traditional and deep learning-based methods. With technological advancements, traditional object detection techniques are insufficient to support current data processing and application requirements. Therefore, deep learning object detection methodologies have emerged as a trending subject of interest within the domain of object detection. Deep learning object detection algorithms can be divided into two- and single-stage methods. Two-stage methods involve generating candidate boxes that potentially contain objects using a region proposal network (RPN), and thereafter classifying and regressing box coordinates using a convolutional neural network (CNN). Representative algorithms include the Region-CNN series. Unlike two-stage methods, single-stage methods do not rely on an RPN; they use a CNN directly to extract image features and offer an end-to-end detection approach. One-stage deep learning object detection methods include the single-shot multibox detector (SSD) [4] and the you only look once (YOLO) series [5], [6], [7], [8], [9], [10], particularly the ultralytics version of YOLO [11], [12].

In recent years, there has been a relative scarcity of research papers utilizing deep learning methods for detecting birds near airports. The published papers mostly employ attention mechanisms, feature fusion techniques, and improved loss functions to enhance the detection accuracy of small birds by deep neural networks, achieving significant results. However, these studies still have certain limitations. In reference [13], for instance, the authors directly utilized bird images from the COCO dataset to train models for detecting foreign objects on airport runways. Yet, since these images differ greatly from real airport scenarios, applying the trained models to actual airports may lead to a decrease in detection accuracy. Similarly, in reference [14], the authors utilized bird data from publicly available datasets of wild birds in a wind farm; in reference [15], they used the BIRDS450 dataset. In reference [16], the authors employed the Drone vs. Bird Detection Challenge dataset to investigate intrusions of drones and birds into airports. Moreover, existing research exhibits relatively low detection accuracy in detecting airport birds. For example, in reference [16], by replacing the detection algorithm's backbone with SqeezeNet, a detection accuracy of 77% was achieved; in reference [17], the authors improved the detection transformer model using attention mechanisms, resulting in a detection accuracy of 75.2%; in reference [18], YOLOv4 algorithm was utilized for detecting airport bird targets, with a final detection accuracy of 71.89%. We can conclude from the above information that the use of deep learning methods to detect airport birds is still a relatively new field, and it is necessary to continue to improve the accuracy of airport bird detection algorithms. In this study, real airport bird images were utilized to study the airport bird detection task, which ultimately improved the detection accuracy to 77.1%. The motivation behind this study lies in the ability of deep learning bird detection methods to overcome the drawbacks of high radar detection costs, low accuracy and slow speed of manual observation. Such methods are well-suited for application in small- and medium-sized airports with limited budgets. By employing this approach, these airports can effectively detect birds near runways, thereby providing decision support for air traffic controllers and pilots, ultimately ensuring the safe operation of aircraft.

As shown in Figure 1, in bird detection, bird targets typically occupy fewer pixels, and the size of the instances is much smaller. This implies that bird features in the images are not prominent and are susceptible to background interference, making it challenging for the algorithm to extract feature information and resulting in poor detection performance.

Attention mechanisms emulate the perceptual and cognitive faculties inherent in the human sensory apparatus, enabling neural networks to concentrate their focus on pertinent aspects while processing the input images [19]. By leveraging the attention mechanism, neural networks can actively and selectively acquire crucial information from the input data, leading to enhanced model performance and improved generalization capabilities. This attention method enables the network to focus its attention on important features, thereby facilitating more effective learning and better adaptation to diverse datasets. Reference [20] embedded a channel and spatial attention block in YOLOv5's Backbone after the SPP block, which enables the network to capture the location information of the target more accurately.

Owing to their smaller receptive fields, shallow feature maps within a network tend to gather more detailed and specific information. Consequently, shallower layers of the network contain finer details and intricate information [21]. Utilizing the shallow-layer information of a neural network is a popular method for recognizing small targets. The combination of shallow- and deep-layer features can improve the detection mAP@50 of the YOLOv5 network.

To address the challenges of detecting bird targets near airport runways, such as bird targets with a small pixel size and the difficulty in detecting or omitting bird targets, this study proposes SMB-YOLOv5, based on YOLOv5s6, to improve bird detection accuracy while significantly reducing the model size. By redesigning the feature fusion network and reusing shallow-level feature maps, the capability of the network to accurately detect small targets was significantly improved, along with the removal of certain modules and the detection head, leading to a reduction in network size.

In conclusion, this study makes the following notable contributions:

FIGURE 1. In-flight bird images.

- Introducing the self-supervised predictive convolution attention block (SSPCAB): The incorporation of the SSPCAB attention mechanism enhances the feature extraction capabilities of the feature extraction network (i.e., Backbone). The SSPCAB minimizes the model size while augmenting the capacity of the model to discern various features within the image. It enhances the detection of small- and low-contrast targets within scenes.
- A more powerful multi-branch block (MBB) was designed by replacing the C3 block with an MBB block. The MBB comprises parallel convolutional branches with different receptive fields. This improves the network's expressive capabilities, enriching the feature space, and reducing instances where the model predicts the background as bird targets.
- Fusing shallow-level features from the Backbone: Drawing inspiration from [22], shallow-level features from the Backbone were fused by incorporating the output feature map of the first C3 block into the Neck. The replacement of the path aggregation network (PANet) with a bidirectional feature pyramid network (BiFPN) [23] enables the network to leverage more detailed information from shallow-level features, thus improving the mAP@50 of small target detection.
- Reduction in network parameters and model size: The removal of one Conv module and one C3 module from the Backbone network, along with the removal of one detection head, effectively reduced the network parameters and model size.
- In the loss function of YOLOv5s6, replacing the complete IoU loss (CIoU) with an efficient IoU loss (EIoU) improves the object detection accuracy. The model's convergence speed has improved.

II. RELATED WORK

A. RESEARCH STATUS

Image object detection is applied in various fields, including bird recognition, facial recognition, and traffic sign recognition [24]. At present, deep learning-based object detection algorithms have demonstrated promising performance on datasets containing medium- to large-sized targets. The small size of the bird target within a picture results in fewer pixels, making it challenging for the YOLOv5s6 to extract features of the small bird target, thus leading to reduced detection performance. Two-stage algorithms offer high accuracy in object detection tasks but suffer from slow frames per second (FPS), complex training, and large model sizes. To address these issues, a single-stage deep learning based object detection algorithm, the YOLO algorithm, was introduced. The YOLO method greatly improves detection speed at the expense of little accuracy. The YOLO algorithm also has a low number of parameters, low memory consumption for training, and short training time. Although the YOLO series algorithms excel in detecting large objects, their efficacy diminishes when detecting tiny birds at airports. This is primarily owing to difficulties in extracting features from small objects, the uneven distribution of small object samples in training data, challenges in setting prior boxes, difficulties in defining loss functions, and problems matching negative and positive samples [22].

Several methods have been proposed to improve YOLOv5s6's ability to extract features for small birds. For instance, reference [25] introduced a refined feature pyramid method known as the AF-FPN. This model integrates the adaptive attention module (AAM) and feature enhancement module (FEM) to minimize information loss during feature map generation, thereby augmenting the representation capabilities of the feature pyramid. Reference [26] incorporated a

convolutional channel attention block following each feature fusion to improve the detection mAP@50 for minor defects. Reference [27] introduced a convolutional block attention module (CBAM) into the YOLOv7 model. This method models the relationships among channels and dynamically learns weights to adjust feature responses across channels. This directs the model to prioritize features with rich information, consequently enhancing the accuracy of detecting small objects. Reference [28] integrated a dense block into the YOLOv2 network. This inclusion bolstered the proficiency of the network in capturing features from diminutive objects, thereby augmenting the detection accuracy.

Reference [29] emphasized that during the network inference process, the features and spatial details of small objects tend to gradually diminish or decrease. This can be mitigated using multiscale feature methods (i.e., feature fusion) to combine the fine-grained information from shallower-level feature maps with the semantic information from higher-level feature maps, thereby reducing information loss and improving the network's accuracy. Reference [30] introduced a feature pyramid network (FPN) comprising a bottom-up network, top-down network, and lateral connections between them to achieve feature fusion. Building upon the FPN, reference [31] made further improvements by adding an additional bottom-up network and lateral connections to create PANet, further enhancing the feature fusion efficiency and boosting network accuracy. Reference [32] introduced an enhanced feature fusion architecture (PB-FPN) derived from PANet and BiFPN, which substantially boosted the capability of the model to detect small targets. In addition to incorporating attention mechanisms and employing feature fusion techniques, modifying the loss function of the model can enhance the detection accuracy. Reference [33] introduced a Focal DIoU loss into YOLOv3's loss function for calculating bounding box regression loss, ultimately improving the accuracy and convergence speed of the algorithm.

In the field of computer vision, CNNs have long dominated due to their powerful ability to extract local features. However, in recent years, vision transformers (ViTs) have emerged as a promising alternative. They have demonstrated performance comparable to, or even superior to, CNNs in certain visual tasks. In reference [34], the authors utilized a ViT-based deep neural network to classify brain tumors, achieving a classification accuracy of 98.7%, effectively alleviating the burden on radiologists. Reference [24] compared the performance of seven CNNs and five ViTs on three traffic sign datasets, but the results revealed that vision transformers do not possess a competitive advantage. Reference [35] introduced adaptive vision transformers (AdaViT), which learn to determine usage policies for self-attention heads, patches, and transformer blocks individually for each input, aiming to enhance the inference efficiency of vision transformers while minimizing the decrease in accuracy. Reference [36] introduced the application of feature fusion strategies to cross-attention multi-scale vision transformers (CrossViT), utilizing two branches to process large-scale and small-scale image patches, and proposed a lightweight token fusion block based on cross attention. The accuracy of the model is the same or even better than that of CNNs. Many ViT methods have the drawbacks of a larger number of model parameters and computation, higher hardware requirements; higher GPU memory consumption for model training; and long training time. For example, during the training of the real time detection transformer (RT-DETR) [37], it was necessary to reduce the number of channels in each layer to prevent the GPU memory required for training from surpassing the GPU's memory capacity.

While the aforementioned improvement methods indeed enhance the detection accuracy of the network, they also come with certain drawbacks. For instance, some introduced enhancements increase the parameters and computation of the object detection network. Moreover, they may not fully leverage the richer detail information contained in the shallow features of the feature extraction network. In this study, we aim to mitigate these shortcomings as much as possible to further enhance the overall performance of the object detection network.

B. STRUCTURE OF YOLOV5S6

The YOLOv5 algorithm is the fifth iteration of the YOLO series and is a classic single-stage object detection model. Figure 2 shows its network architecture. YOLOv5, developed by Ultralytics, is an effective and lightweight deep learning architecture. YOLO is a set of object detection models recognized for its simplicity and speed, with the most recent version being the YOLOv8 series. Because of its excellent computational efficiency and ability to execute quickly on various hardware platforms, YOLOv5 has garnered significant attention for industrial use. Consequently, the use of YOLOv5 for bird detection at airports is expected to produce better detection results.

The network structure of YOLOv5s6 includes several modules, making it convenient to modify the network architecture and add or remove modules to achieve a better performance. The compositions of these modules are shown in Figure 2. The C3 module comprises three Conv modules and n BottleNecks, allowing for better learning of the residual features while increasing the YOLOv5s6 network depth and receptive fields. The BottleNeck module in the Backbone network employs residual connections, as shown in Figure 2, whereas the remaining part of YOLOv5s6 does not use residual connections, implying that the BottleNeck module comprises only two consecutive convolutional modules. The spatial pyramid pooling (SPP) module, proposed by He in 2015, addresses issues related to image cropping, scaling operations, image distortion, and the extraction of redundant features in CNNs. This significantly increases the speed of generating candidate boxes and reduces the computational cost [38]. The SPP fast (SPPF), introduced by Glenn Jocher, the author of YOLOv5, is a faster version of SPP.

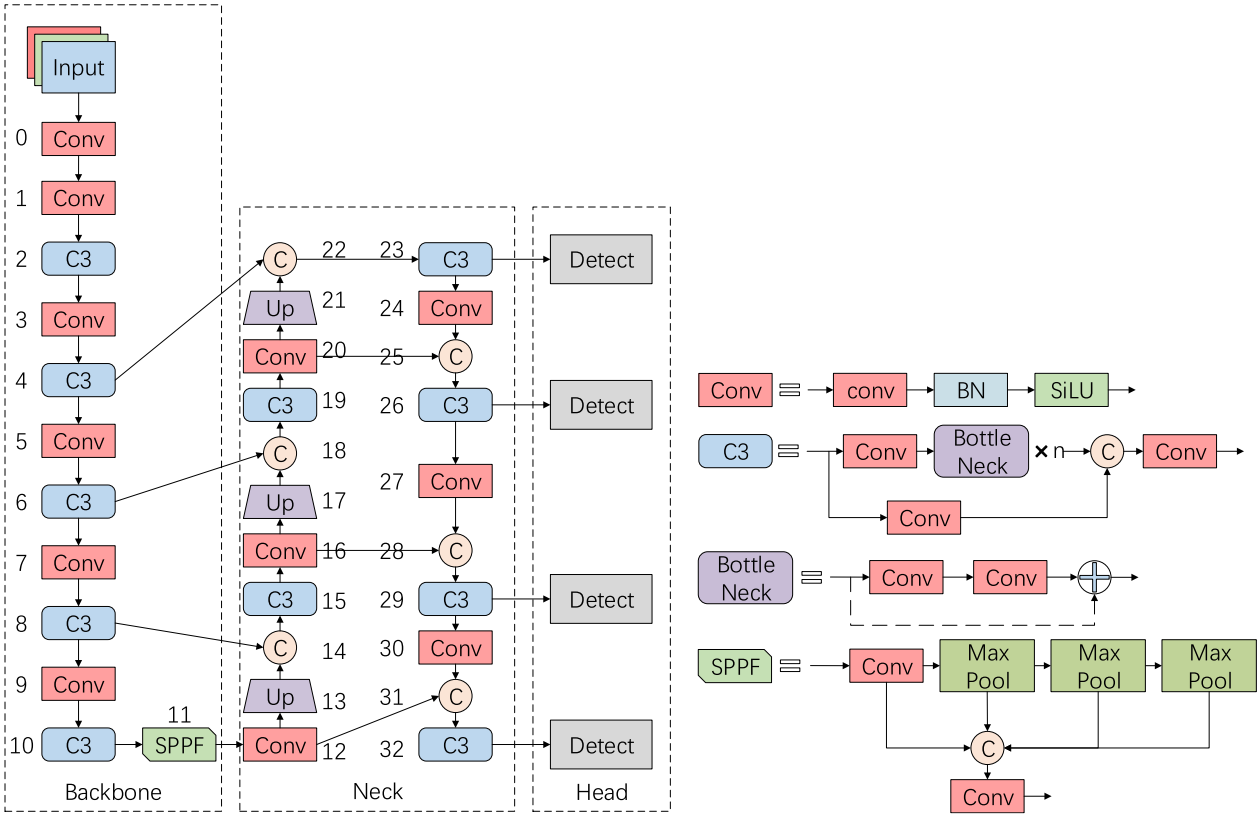

FIGURE 2. Structure of YOLOv5s6 and composition of Conv, C3, BottleNeck, and SPPF modules in YOLOv5s6. C denotes Concat.

III. PROPOSED SMB-YOLOV5 ALGORITHM

This section presents a comprehensive overview of the architectural design of the SMB-YOLOv5 model. The SMB-YOLOv5 network is based on a CNN, whereas the overall structure of the network is built upon YOLOv5s6. YOLOv5s6, a variant of YOLOv5, is more accurate than YOLOv5s and can efficiently handle instances of different scales. To balance the computational efficiency of the network with the details required to detect smaller objects, an image size of 1024 × 1024 pixels was employed for model training and testing. The YOLO series is a prominent example of a one-stage detector that has consistently maintained efficiency and accuracy. Section A describes the fundamental principles of the proposed SSPCAB module and its position in the network. Section B describes the fundamental principles of an MBB and its position in a network. In Section C, we replace the CIoU in the loss function with the EIoU and explain the formula for the EIoU. In Section D, we describe the utilization of the shallow-level features of the network more effectively and making of the network more lightweight.

A. SPPCAB

The SSPCAB module [39] can learn to predict mask information using contextual information. The SSPCAB module's input and output tensors are the same size, making them easy to integrate into any CNN. The principles of the SSPCAB module are elaborated as follows. Figure 3 (1) illustrates the architecture of the SSPCAB.

First, an operation of padding is carried out on the input feature map, where padding = kernel_size + dilation. Second, assume that the width of the feature map after padding is W and the height is H. Slice the padded feature map four times: The first slice, with the top-left corner of the feature map as the origin, preserves a feature map of size $(W - A) \times (H - A)$; the second slice, with the top-right corner of the feature map as the origin, preserves a feature map of size $(W - A) \times (H - A)$; the third slice, with the bottom-left corner of the feature map as the origin, preserves a feature map of size $(W - A) \times (H - A)$; the fourth slice, with the bottom-right corner of the feature map as the origin, preserves a feature map of size $(W - A) \times (H - A)$. Each of the four sliced feature maps is passed through separate 1×1 convolutional layers. The formula for the cropping length A is $A =$ kernel_size + 2dilation + 1. Third, the output features from the aforementioned four convolutional layers are summed, and the ReLU activation function is applied. Fourth, squeeze and excitation networks (SENet) are applied to the activated features. Figure 4 illustrates the architecture of the SENet module. The final output feature map bears the same shape as the original feature map.

The convolutional operation in SSPCAB is referred to as the masked convolution operation, and the masked convolutional kernel is illustrated in Figure 3 (2). The learnable

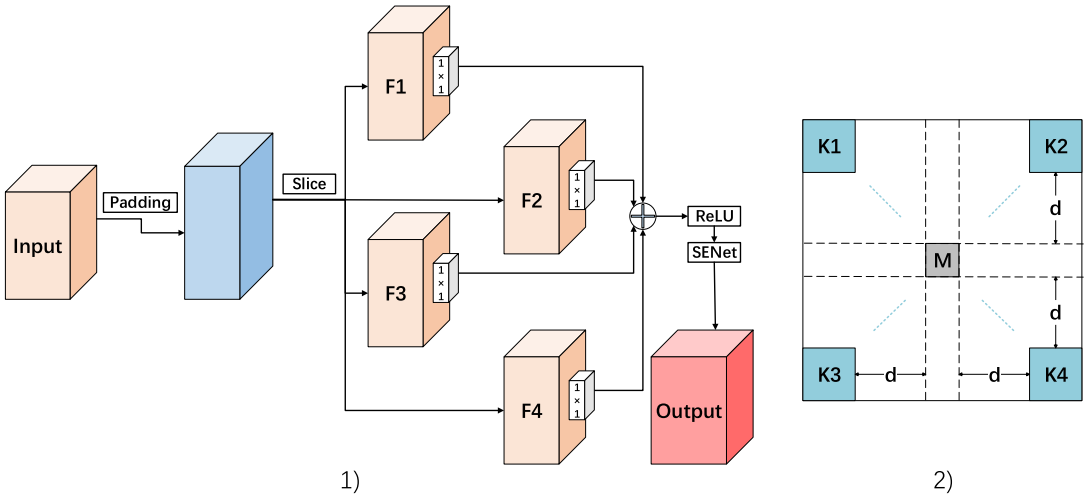

FIGURE 3. Structure of the SSPCAB module and receptive field of the SSPCAB module.

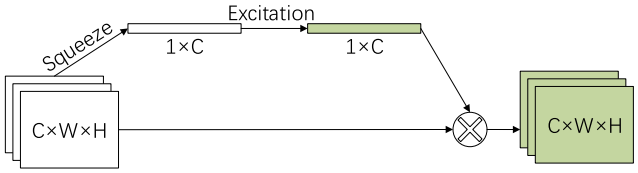

FIGURE 4. Structure of SENet.

parameters of the convolution are located at the corners of the Figure 3 (2) and are denoted by the sub-kernels K_i, $\forall_i \in \{1, 2, 3, 4\}$. As shown by M, each kernel K_i is positioned at a distance (determined by the dilation rate) $d \in N^+$ away from the masked region at the receptive field's center.

C3 before SPPF is replaced with SSPCAB. Incorporating the SSPCAB module into the feature extraction network enhances feature representation, improves object detection performance for low-contrast scenarios, and enables adaptive channel attention mechanisms to enhance the model performance in detecting small birds. The self-supervised learning and global structure learning capabilities of the SSPCAB module introduced more prior information into the model, aiding in better generalization during training. This helps reduce the risk of overfitting on small datasets and enhances the performance across different datasets and scenarios.

B. PROPOSED MUTI-BRANCH BLOCK

In the YOLO series, increased network depth notably enhanced the detection performance. However, fewer parallel branch structures are present within the YOLOv5 network module. Different convolutional branches possess varying receptive fields and feature extraction capabilities, allowing them to extract diverse and complex abstract information from targets. In this study, from the perspective of augmenting multiple parallel convolutional branches, we combined feature maps extracted from multiple branches to obtain output features with richer information. We introduced a module known as the MBB module. MBB includes parallel convolutional branches with distinct receptive fields, such as 1 × 1 convolution, 3 × 3 convolution, and 1 × 1 convolution and average pooling layers, etc. The incorporation of multiple convolutional branches enhances the expressive capability of the network and enriches the feature space. By employing the MBB module, the model reduces instances in which the background is misclassified as a bird target in object detection, thereby enhancing the precision of predicting small bird targets. The MBB architecture is shown in Figure 5.

The formula for the MBB is as follows:

$$x1, x2 = split(f_{Conv}^{k=1}(F)) \tag{1}$$

$$\begin{cases} y1 = f_{conv}^{k=1}(f_{Conv}^{k=1}(x1)) \\ y2 = f_{conv}^{k=3}(f_{conv}^{k=1}(f_{Conv}^{k=1}(x1))) \\ y3 = f_{conv}^{k=3}(f_{Conv}^{k=1}(x1)) \\ y4 = f_{AvgPool}(f_{conv}^{k=1}(f_{Conv}^{k=1}(x1))) \end{cases} \tag{2}$$

$$x1' = SiLU(y1 + y2 + y3 + y4) \tag{3}$$

$$F' = f_{Conv}^{k=1}(Cat(x2, x1')) \tag{4}$$

where F denotes the input feature map, $f_{Conv}^{k=1}$ denotes a normal convolution layer, kernel size $= 1 \times 1$, $Split()$ implies that the output features are divided into two parts, $f_{conv}^{k=1}$ denotes a 1×1 convolution operation, when $k = 3$, it implies that the kernel size $= 3 \times 3$, $SiLU()$ denotes the SiLU activation function, $Cat()$ denotes concatenating feature maps according to the channel dimensions, and F' denotes the output feature map.

Introducing a 3 × 3 convolution block into the MBB module increases the receptive field of YOLOv5s6, enabling it to capture more complex feature patterns and finer details. Simultaneously, retaining the 1 × 1 convolution allows for the adjustment of channel numbers in the feature maps through weight adjustments, enhancing the nonlinear

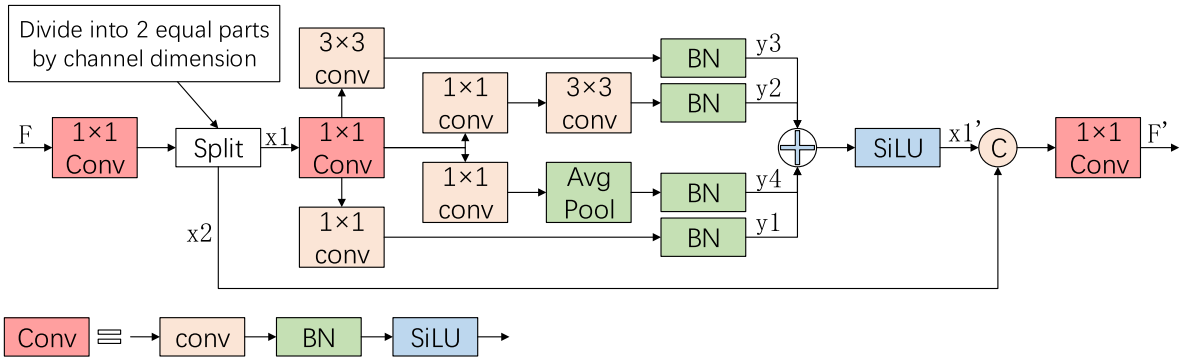

FIGURE 5. Module structure of multi-branch block (MBB).

expressive capabilities while reducing the computational load. The combination of 1 × 1 and 3 × 3 convolutions strikes a balance between the improved feature extraction and network performance. Average pooling reduces the spatial dimensions of feature maps, diminishing the computational load and parameter count while preserving the overall trends of significant features, particularly the relative positional relationships among features, to enhance the robustness of YOLOv5s6. After the convolutional and pooling operations in the four branches, batch normalization was required. The integration of diverse convolutional branches amplifies the feature expression capacity of the module, enabling it to learn more abstract information, thereby enhancing the accuracy of the network. Moreover, the residual connections within the MBB prevent the gradient from vanishing in the network, thereby accelerating the network training and improving its performance.

C. EIOU LOSS

To improve detection accuracy, this study considered improving the loss function. Determining the real position of the target bounding box is an important task in object detection. The YOLOv5s6 model used the CIoU loss function for training. A good regression loss should include the coverage area, center-point distance, and aspect ratio. The CIoU is the sum of the aspect ratio and the distance IoU loss (DIoU). The calculation formula is:

$$CIoU = IoU - \frac{\rho^2(b, b^{gt})}{c^2} - \alpha v \tag{5}$$

$$\alpha = \frac{v}{1 - IoU + v} \tag{6}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \tag{7}$$

where ρ denotes the Euclidean distance between the center points, b^{gt} denotes the center point of the ground truth box, b denotes the center point of the predicted bounding box, c denotes the diagonal length of the minimum bounding rectangle that can simultaneously contain both the predicted box and the ground truth box, α denotes the weight coefficient, v denotes the consistency coefficient used to measure the aspect ratio between the predicted box and the ground truth box, h and h^{gt} denotes the heights of the predicted box and ground truth box, respectively, and w and w^{gt} denotes the width of the predicted box and the ground truth box, respectively.

The formula for the CIoU loss is:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{8}$$

On the basis of DIoU, CIoU also considers the aspect ratio, but v simply reflects the difference in aspect ratio, instead of the real relations between w and w^{gt} or h and h^{gt}, which does not increase the similarity of the aspect ratio, while it prevents the model from efficiently reducing the real difference between (w, h) and (w^{gt}, h^{gt}) [40]. To address this issue, EIoU has been proposed. Based on the CIoU, the aspect ratios were separated, and the differences in width and height were calculated separately. This can accelerate the regression speed of the prediction box, focus the box regression process on better anchors, and improve regression accuracy of the prediction box. The calculation formula is:

$$L_{EIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \frac{\rho^2(w, w^{gt})}{c_w^2} + \frac{\rho^2(h, h^{gt})}{c_h^2} \tag{9}$$

where c_w and c_h denote the width and height of the minimum bounding rectangle that encompasses both the predicted box and ground truth box.

D. UTILIZING SHALLOW-LEVEL FEATURES OF THE NETWORK AND THE OVERALL STRUCTURE OF SMB-YOLOV5

When CNNs are used for feature extraction, different network depths correspond to features at different levels [41]. To detect small objects, lower-level features are desirable because they often have higher resolution and contain richer details about the objects that need to be detected. The resolution of the feature maps gradually decreases as the YOLOv5s6 network depth increases, and the network becomes less sensitive to fine details. However, at this stage, the features contain more semantic information than shallower-level features.

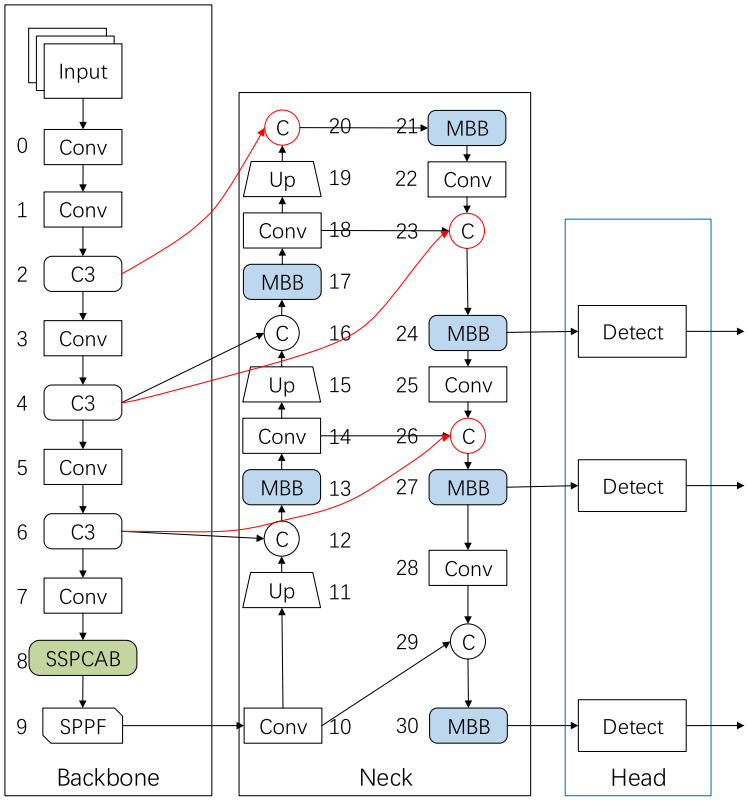

FIGURE 6. Structure of SMB-YOLOv5.

Therefore, for scenarios such as the detection of small birds in an image, utilizing the shallow-level features of the Backbone network is crucial. To achieve this, the following changes are made to the network: First, in Figure 2, the 9th and 10th modules are deleted. Second, the feature maps from the shallower C3 module, which is the 2nd module, are connected to the subsequent PANet in the Neck section. Third, inspired by BiFPN [23], the features provided by the shallower layer, indicated by the red lines in Figure 6, are utilized. Specifically, the 23rd Concat module is connected to the 4th C3 block in the Backbone of YOLOv5s6, and the 26th Concat module is connected to the 6th C3 module in the Backbone. Finally, because the features generated by the 21st MBB block in Figure 6 would significantly increase the computational cost when passed to the detection head, the detection head is removed. The resulting improved YOLOv5s6 model with these four modifications is shown in Figure 6.

These modifications significantly reduce the network parameter count, and because the network utilizes shallow-level features, it significantly improved its capability to detect small birds. This achieved the goals of a lightweight model and improved detection accuracy.

IV. EXPERIMENTAL RESULTS

A. DATASETS

The dataset used in this study was obtained from part of the AirBirds dataset, as described in [42], for training the SMB-YOLOv5 model. The AirBirds dataset is the first large-scale image dataset specifically designed to study bird strikes at airports. It comprises 118,312 images and 409,967 YOLO-formatted bird annotation boxes, with the majority of the images having a resolution of 1920 $\times$ 1080 pixels. The dataset contains images captured by a camera network deployed at a real-world airport over the course of one year, spanning four seasons, covering various bird species, lighting conditions, and 13 types of weather conditions. The average annotation size of bird instances in the 1920 $\times$ 1080 pixels-sized images in this dataset is $<$10 pixels. We first selected 15,000 images with larger instances and sorted them in descending order, based on the pixel size occupied by the largest instance. Subsequently, we divided these images into three datasets: big, medium, and small. The first 5,000 images constituted the big dataset, the subsequent 5,000 images comprised the small dataset, and the 5,000 images in the middle constituted the medium dataset. A split ratio of 70% for training, 10% for validation, and 20% for testing was used to divide each dataset into subsets for training, testing, and validation. Finally, the best-performing dataset among these three was selected for the remaining experiments.

To further understand the sizes of the bird instances in the aforementioned three datasets. The size distributions of the lengths and widths of the instances for the three datasets are shown in Figure 7. The height and width were normalized. The vertical axis represents the height while the horizontal axis represents the width. Big datasets contain

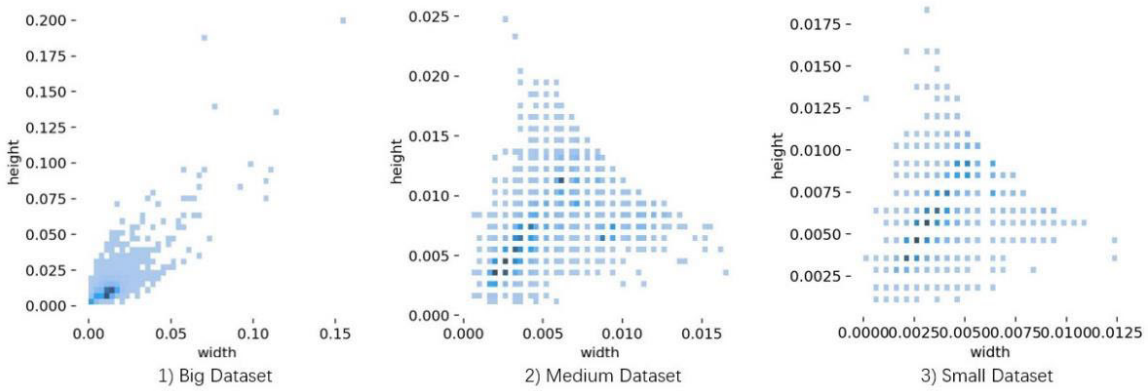

FIGURE 7. Size distribution of the length and width of instances for three datasets. From left to right, it represents the instance size distribution for the big, medium, and small datasets, respectively.

several instances with larger sizes, whereas medium and small datasets have instances with smaller sizes.

B. EXPERIMENTAL ENVIRONMENT AND HYPERPARAMETER CONFIGURATION

The training and testing environment configurations used in this study are listed in Table 2. The hyperparameters for this experiment were configured as follows: during the training, the batch size was eight, training was 300 rounds, using eight workers, the initial learning rate was 0.01, the minimum learning rate was 0.0001, using stochastic gradient descent (SGD) as the optimizer, the momentum was 0.937, and a cosine annealing strategy was used to adjust the learning rate. The image size used during training and testing was 1024 × 1024 (the image size was set to 1280 × 1280 for the three-dataset performance comparison experiments described in Section D).

TABLE 1. Hardware and software environment configuration.

Name	Version
CPU	Intel(R) Xeon(R) Silver 4110 CPU
GPU	NVIDIA GeForce RTX 2080Ti 11GB
RAM	128GB
System	Windows 10 Professional 22H2
Python	3.8.17
Pytorch	1.8.1
CUDA	10.2

C. EVALUATION METRICS

The evaluation metrics utilized in this work include recall (R), precision (P), mean average precision (mAP@50), model detection speed in FPS, model parameter size (Params), and weight size. Precision represents the proportion of correctly predicted positive images among the images recognized by the model. Recall denotes the fraction of positive predictions among all actual positive samples. The average precision (AP) is indicative of the area under the precision–recall curve,

	Positive	Negative
Predicted as positive	TP (True Positive)	FP (False Positive)
Predicted as negative	FN (False Negative)	TN (True Negative)

FIGURE 8. Prediction of positive and negative samples.

and the mAP refers to the average value of all categories of AP in the object detection task. The following situations were encountered during training, as shown in Figure 8.

More advanced evaluation metrics such as P, R, AP, mAP can be obtained from the confusion matrix [43] in Figure 8, and their formulas are shown in (10)–(13) as follows:

$$P = \frac{TP}{TP + FP} \tag{10}$$

$$R = \frac{TP}{TP + FN} \tag{11}$$

$$AP = \int_0^1 P(t)dt \tag{12}$$

$$mAP = \frac{\sum_{n=1}^{n} AP_n}{N} \tag{13}$$

where AP denotes the average precision and $P(t)$ denotes the precision rate when the threshold value of IoU is taken as t. N denotes the number of classes, and in this experiment, N = 1; and AP_n denotes the AP of the nth category.

FPS denotes the maximum number of images the algorithm can process within a one-second interval.

D. EXPERIMENTAL RESULTS OF THREE DATASETS

In comparative experiments with three differently sized datasets, the configuration utilized involved a batch size of 12 and images sized at 1280 × 1280 pixels (Owing to hardware limitations, the batch and image sizes were 8 and 1024 × 1024 pixels, respectively, in later ablation and comparative experiments), with the remaining hyperparameters set as previously described. Following training and validation

TABLE 2. Testing results of three datasets.

Dataset	P	R	mAP@50	FPS	Params	Size/MB
Big	81.4%	70.6%	75.0%	22	12.3 M	25.4
Medium	64.2%	51.7%	52.7%	24	12.3 M	25.4
Small	62.3%	53.1%	50.3%	19	12.3 M	25.4

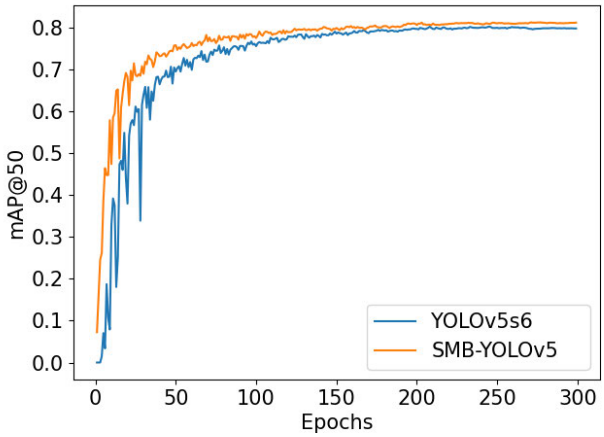

FIGURE 9. mAP@50 curves of SMB-YOLOv5 and YOLOv5s6.

on the three datasets, the test set images were tested, and the test results are listed in Table 2.

Table 2 shows that the big dataset had higher P, R, and mAP@50 values than the other datasets. Therefore, the best-performing big dataset was selected for ablation and comparative experiments in this study.

E. TRANING RESULTS OF SMB-YOLOV5 AND YOLOV5S6

To ascertain the efficacy of the SMB-YOLOv5 model, we performed a comparative analysis of the training outcomes of the YOLOv5s6 and SMB-YOLOv5 models. The results are shown in Figures 9, 10, and 11. The SMB-YOLOv5 model demonstrates faster convergence than the YOLOv5s6 model, consistently maintaining higher mAP@50 values throughout the training process. The training and validation losses of SMB-YOLOv5 were less than those of YOLOv5s6 during training. Both models exhibited instability during the initial 100 training epochs, causing fluctuations in the mAP@50 values. After 100 epochs, the mAP@50 values slowly increased, gradually flattened, and then no longer increased. During the training process, the Precision of the SMB-YOLOv5 has always been higher than that of the YOLOv5, and the convergence speed is faster than that of the YOLOv5. For the Recall values during the training process, during the initial 150 training epochs, SMB-YOLOv5 had a higher Recall value than YOLOv5; About 150 to 270 epochs, the Recall values of the two models are similar; After 270 epochs, the Recall value of SMB-YOLOv5 is higher than YOLOv5.

F. ABLATION EXPERIMENTS

In neural networks, ablation experiments are frequently used to test the impact of a certain module or modification of the network. Table 3 lists the results of the ablation experiments.

To validate the effectiveness of each improvement, we conducted comparative experiments between each method and the YOLOv5s6 model. Subsequently, we integrated all four improvements into a single model and conducted an experiment. First, introducing the SSPCAB attention mechanism into the Backbone network of YOLOv5s6 resulted in increases of 0.002 in R and 0.005 in mAP@50, along with decreases of 0.008 in P, 2 frames per second (FPS), 0.1 million Params, and 0.7 megabytes (MB) in weight size. Second, introducing an MBB into the Neck of YOLOv5s6 led to increases of 0.007 in P, 0.007 in R, and 0.006 in mAP@50, with no change in FPS, along with reductions of 1.3 million Params and 3.5 MB in weight size. Next, leveraging the shallow features of the Backbone network and removing some modules resulted in increases of 0.001 in P, 0.006 in R, and 0.011 in mAP@50, along with decreases of 3 FPS, 5.5 million Params, and 11.1 MB in weight. Finally, replacing CIoU with EIoU led to decreases of 0.006 in P, 0.001 in R, and 4 FPS, along with an increase of 0.013 in mAP@50, with no change in parameters or weight size. Ultimately, integrating all four improvements into a single model resulted in increases of 0.008 in P, 0.008 in R, and 0.026 in mAP@50, along with decreases of 1 FPS, 6.7 million Params, and 13.6 MB in weight size.

The amalgamation of all four improvement methods led to a higher mAP@50 value compared with employing each improvement method individually. Additionally, the SMB-YOLOv5 model demonstrated a lower parameter count and weight size than the other four individual improvement models.

G. COMPARATIVE EXPERIMENTS

We compared the SMB-YOLOv5 model with YOLOv5s6, YOLOv8s, YOLOv4-tiny, YOLOv3-tiny, SSD-Mobilenetv2, YOLOX, CenterNet, YOLOv3, YOLOv6s, and RT-DETR algorithms in our comparative experiments. All the experiments were conducted in the same hardware and software environment. The test results are summarized in Table 4. The SMB-YOLOv5 algorithm exhibited the best mAP@50 performance of 77.1%, which was 64.2% higher than the poorest-performing YOLOv4-tiny algorithm. The YOLOv4-tiny model exhibited the highest FPS performance at 89 fps. SSD-Mobilenetv2 had the lowest model parameter count of

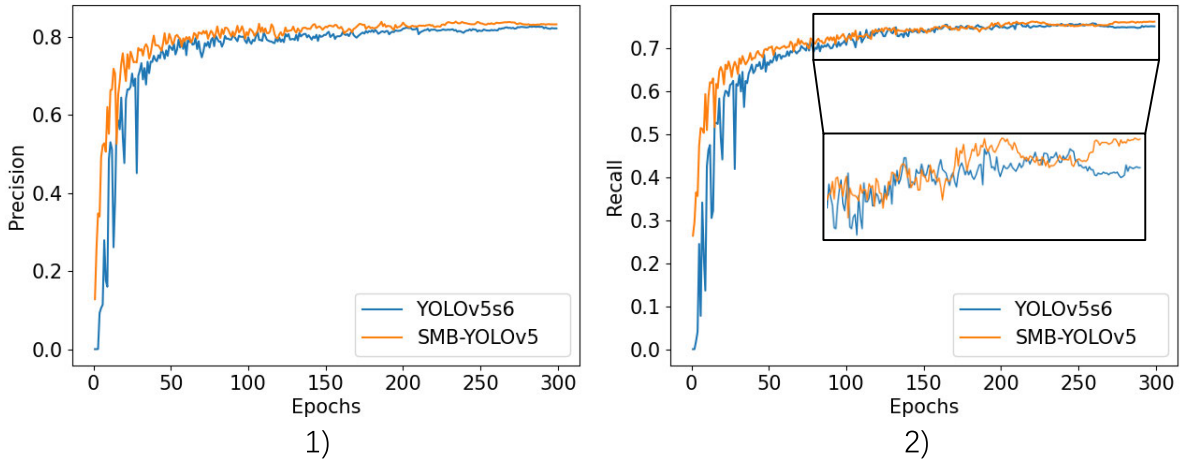

FIGURE 10. Precision and recall curves of SMB-YOLOv5 and YOLOv5s6.

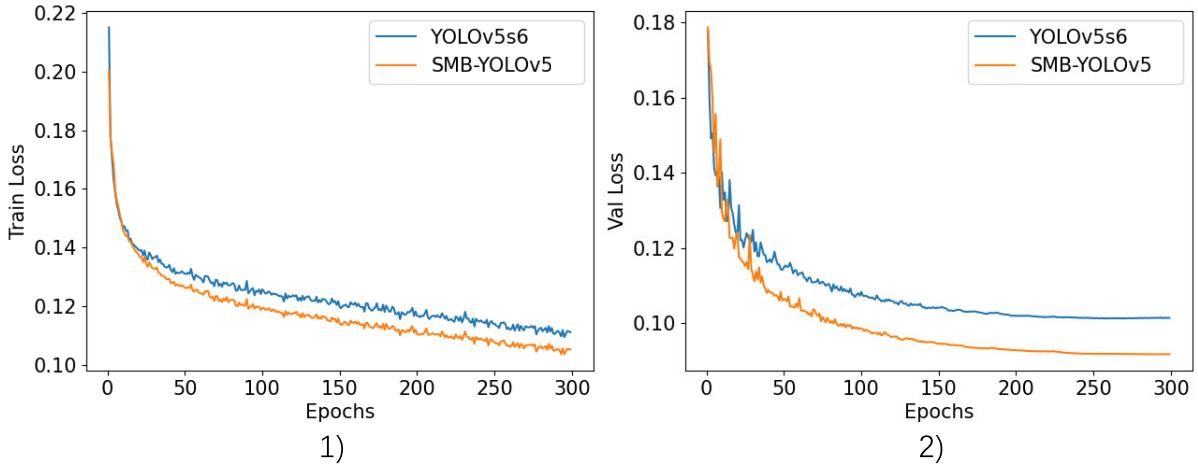

FIGURE 11. Train and validation losses of SMB-YOLOv5 and YOLOv5s6.

TABLE 3. Results of the ablation experiment.

Method	Image Size	P	R	mAP@50	FPS	Params	Size/MB
YOLOv5s6	1024	81.8%	70.3%	74.5%	25	12.3 M	25.2
YOLOv5s6 + SSPCAB	1024	81.0%	70.5%	74.9%	23	12.2 M	24.5
improvement	-	-0.008	**+0.002**	**+0.004**	-2	**-0.1 M**	**-0.7**
YOLOv5s6	1024	81.8%	70.3%	74.5%	25	12.3 M	25.2
YOLOv5s6 + MBB	1024	82.5%	71.0%	75.1%	25	11.0 M	21.7
improvement	-	**+0.007**	**+0.007**	**+0.006**	0	**-1.3 M**	**-3.5**
YOLOv5s6	1024	81.8%	70.3%	74.5%	25	12.3 M	25.2
YOLOv5s6 + USF	1024	81.9%	70.9%	75.6%	22	6.8 M	14.1
improvement	-	**+0.001**	**+0.006**	**+0.011**	-3	**-5.5 M**	**-11.1**
YOLOv5s6	1024	81.8%	70.3%	74.5%	25	12.3 M	25.2
YOLOv5s6 + EIoU	1024	81.2%	70.25	75.8%	21	12.3 M	25.2
improvement	-	-0.006	-0.001	**+0.013**	-4	0	0
YOLOv5s6	1024	81.8%	70.3%	74.5%	25	12.3 M	25.2
Ours	1024	82.6%	71.1%	77.1%	24	5.6M	11.6
improvement	-	**+0.008**	**+0.008**	**+0.026**	-1	**-6.7 M**	**-13.6**

FIGURE 12. SMB-YOLOv5's prediction results on larger instances.

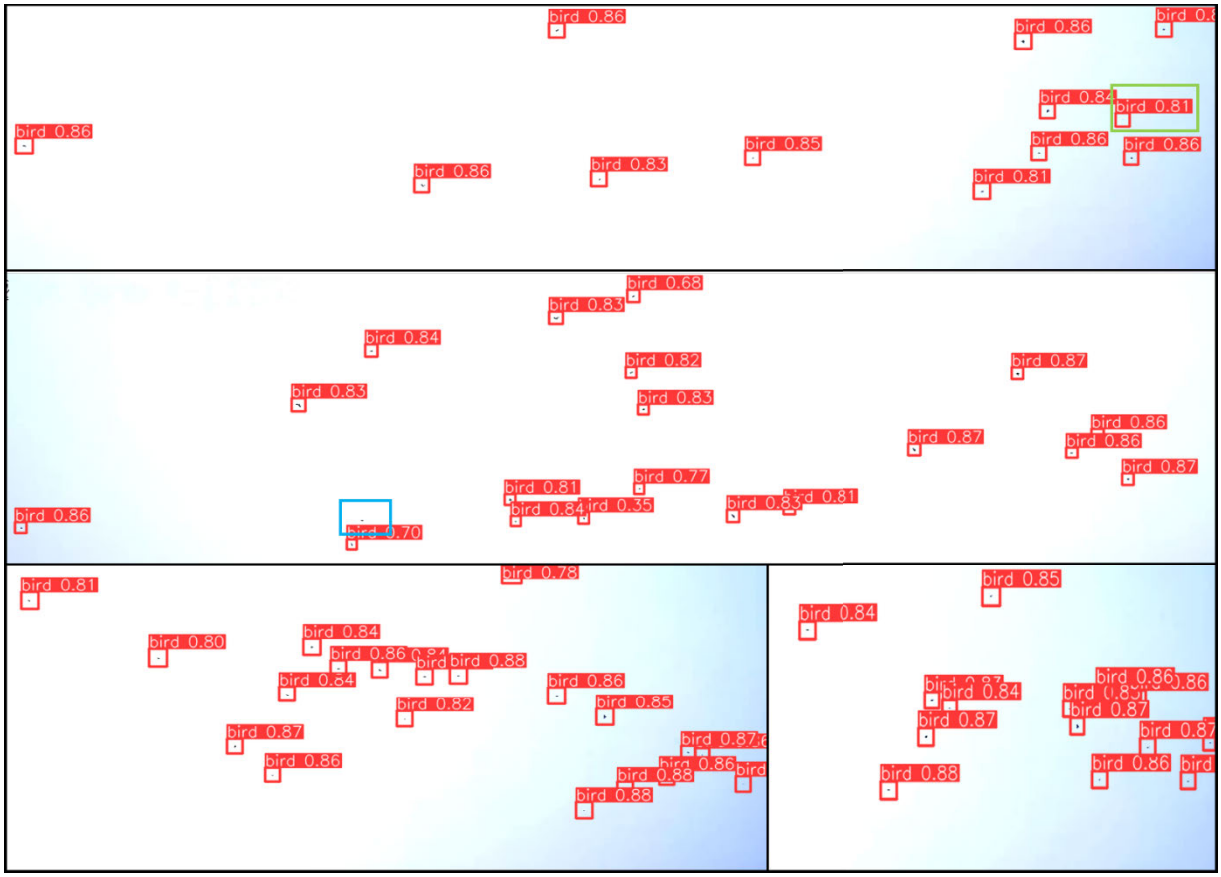

FIGURE 13. SMB-YOLOv5's prediction results on smaller instances.

only 3.5 million, followed by the SMB-YOLOv5 algorithm. The SMB-YOLOv5 algorithm also had the smallest weight size at 11.6 MB, which is 112.4 MB lower than that of the CenterNet model. Among all algorithms, the SMB-YOLOv5 algorithm achieved the highest precision (P) and recall (R) scores of 82.6% and 71.1%, respectively.

TABLE 4. Results of the comparative experiment.

Method	Image Size	P	R	mAP@50	FPS	Params	Size/MB
YOLOv5s6	1024	81.8%	70.3%	74.5%	25	12.3 M	25.2
YOLOv3-tiny	1024	77.8%	61.0%	65.8%	46	8.7 M	17.5
YOLOv4-tiny	1024	32.2%	19.4%	12.9%	89	5.9 M	22.4
YOLOX	1024	-	-	71.1%	33	8.9 M	68.5
YOLOv8s	1024	74.0%	60.0%	66.7%	25	11.1 M	22.0
SSD-Mobilenetv2	1024	2.44%	65.8%	16.6%	75	3.5 M	14.3
CenterNet	1024	37.0%	74.6%	47.6%	29	32.7 M	124.0
YOLOv3	1024	81.4%	69.8%	74.2%	23	61.5 M	123.7
YOLOv6s	1024	70.1%	55.3%	60.4%	26	16.3 M	31.3
RT-DETR	1024	36.2%	45.7%	31.3%	28	9.7 M	19.8
Ours	1024	**82.6%**	**71.1%**	**77.1%**	24	**5.6 M**	**11.6**

From the ablation and comparative experiments, we can conclude that the SMB-YOLOv5 model achieves commendable performance in mAP@50, precision (P), and recall (R) metrics while reducing the model parameter count and weight size.

H. SMB-YOLOV5 DETECTION RESULTS

We can see the detection results through Figure 12 and 13 below. And Figure 12 displays the model's detection results on larger bird instance images, where the model successfully detected all bird instances. Figure 13 illustrates the model's detection results on densely packed and relatively small bird images. The model successfully detected the majority of bird instances. However, there are still instances of predicting the background as birds (As shown in the green rectangle in Figure 13) and missing detections (As shown in the blue rectangle in Figure 13).

V. CONCLUSION

In this study, we proposed an SMB-YOLOv5 model for detecting birds near airport runways. SMB-YOLOv5 incorporates some of the latest computer vision techniques, such as the SSPCAB attention mechanism and BiFPN, and utilizes data augmentation and training techniques. Incorporating SSPCAB enabled the network to concentrate on pertinent regions of interest, thereby improving its ability to detect small birds. The MBB enhances the expressive capabilities of YOLOv5s6, enriches the feature space, reduces instances where the object detection model misclassifies the background as a bird target, and increases the precision of the network in predicting small bird targets. Owing to the rich and detailed features contained in shallow features, the network leverages shallow features by incorporating them into the Neck structure, combined with the BiFPN network, achieving the reutilization of shallow features. Finally, replacing the CIoU loss with EIoU loss accelerated model convergence and improved the mAP@50 value of the model. These four improvements enhanced network performance. By testing the test dataset, SMB-YOLOv5 achieved an accuracy of 77.1%, 2.6% more accurate than YOLOv5s6. A notable feature of the SMB-YOLOv5 network is its network architecture, which significantly reduces the Params and weight size of the SMB-YOLOv5 network. This translates into a lighter model with a smaller memory footprint, making it an ideal choice for deployment in devices with limited resources.

By deploying the SMB-YOLOv5 model on the central processing equipment of an airport, processing image information from distributed monitoring devices can accurately and quickly detect flying birds in the airspace near the airport. Therefore, the SMB-YOLOv5 object detection algorithm can overcome the shortcomings of manual observation and radar detection, improve airport operation efficiency, and ensure airport operational safety. In addition to the improvements in the aforementioned performance metrics, the detection speed of the model decreased. Therefore, future studies should focus on enhancing the FPS of this model. In the near future, we aim to integrate the SMB-YOLOv5 algorithm with airport bird deterrent systems. The goal of this integration is to improve the speed and accuracy of avian target detection and tracking, thereby providing a deeper understanding of bird behavior. By analyzing these data, more effective bird deterrent measures can be implemented to ensure the safety and efficiency of civil aviation.

REFERENCES

[1] I. C. Metz, J. Ellerbroek, T. Mühlhausen, D. Kügler, and J. M. Hoekstra, "The bird strike challenge," *Aerospace*, vol. 7, no. 3, p. 26, Mar. 2020, doi: 10.3390/aerospace7030026.

[2] A. A. S. Desoky, "A review of bird control methods at airports," *Global J. Sci. Frontier Res. E*, vol. 14, pp. 41–50, Jan. 2014.

[3] Y. Amit, P. Felzenszwalb, and R. Girshick, "Object detection," in *Computer Vision—ECCV*. Cham, Switzerland: Springer, 2020, pp. 1–9, doi: 10.1007/978-3-030-03243-2_660-1.

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Computer Vision—ECCV* (Lecture Notes in Computer Science), vol. 9905, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[6] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Honolulu, HI, USA, 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.

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

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

[9] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, "YOLOv6: A single-stage object detection framework for industrial applications," 2022, *arXiv:2209.02976*.

[10] C.-Y. Wang, A. Bochkovskiy, and H.-Y. Mark Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7464–7475, doi: 10.1109/CVPR52729.2023.00721.

[11] *GitHub*. Accessed: Dec. 1, 2023. [Online]. Available: https://github.com/ultralytics/yolov5

[12] *GitHub*. Accessed: Dec. 1, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[13] J. Taupik, T. Alamsyah, A. Wulandari, E. U. Armin, and A. Hikmaturokhman, "Airport runway foreign object debris (FOD) detection based on YOLOX architecture," in *Proc. Int. Conf. Comput. Sci., Inf. Technol. Eng. (ICCoSITE)*, Jakarta, Indonesia, Feb. 2023, pp. 40–43, doi: 10.1109/ICCoSITE57641.2023.10127676.

[14] A. Schumann, L. Sommer, J. Klatte, T. Schuchert, and J. Beyerer, "Deep cross-domain flying object classification for robust UAV detection," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveillance (AVSS)*, Lecce, Italy, Aug. 2017, pp. 1–6, doi: 10.1109/AVSS.2017.8078558.

[15] Y. Chen, Y. Liu, and Z. Wang, "Design of simulation experimental platform for airport bird control linkage system based on improved YOLOv5," in *Proc. IEEE 11th Joint Int. Inf. Technol. Artif. Intell. Conf. (ITAIC)*, vol. 11, Dec. 2023, pp. 1279–1284, doi: 10.1109/ITAIC58329.2023.10408937.

[16] H. J. Al Dawasari, M. Bilal, M. Moinuddin, K. Arshad, and K. Assaleh, "DeepVision: Enhanced drone detection and recognition in visible imagery through deep learning networks," *Sensors*, vol. 23, no. 21, p. 8711, Oct. 2023, doi: 10.3390/s23218711.

[17] L. Shanliang, L. Yunlong, Q. Jingyi, and W. Renbiao, "Airport UAV and birds detection based on deformable DETR," *J. Phys., Conf.*, vol. 2253, no. 1, Apr. 2022, Art. no. 012024, doi: 10.1088/1742-6596/2253/1/012024.

[18] K. Ummah, M. F. Hidayat, D. Kurniawan, and J. Sembiring, "Bird detection system design at the airport using artificial intelligence," *Avia*, vol. 4, no. 2, pp. 59–67, 2022.

[19] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, Sep. 2021, doi: 10.1016/j.neucom.2021.03.091.

[20] W. Zhao, M. Syafrudin, and N. L. Fitriyani, "CRAS-YOLO: A novel multi-category vessel detection and classification model based on YOLOv5s algorithm," *IEEE Access*, vol. 11, pp. 11463–11478, 2023, doi: 10.1109/ACCESS.2023.3241630.

[21] S. Yang, D. Jiao, T. Wang, and Y. He, "Tire speckle interference bubble defect detection based on improved faster RCNN-FPN," *Sensors*, vol. 22, no. 10, p. 3907, May 2022, doi: 10.3390/s22103907.

[22] K. Tong, Y. Wu, and F. Zhou, "Recent advances in small object detection based on deep learning: A review," *Image Vis. Comput.*, vol. 97, May 2020, Art. no. 103910, doi: 10.1016/j.imavis.2020.103910.

[23] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," 2019, *arXiv:1911.09070*.

[24] Y. Zheng and W. Jiang, "Evaluation of vision transformers for traffic sign classification," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–14, Jun. 2022, doi: 10.1155/2022/3041117.

[25] J. Wang, Y. Chen, Z. Dong, and M. Gao, "Improved YOLOv5 network for real-time multi-scale traffic sign detection," *Neural Comput. Appl.*, vol. 35, no. 10, pp. 7853–7865, Apr. 2023.

[26] Y. Xu, F. Sun, and L. Wang, "YOLOv5-PD: A model for common asphalt pavement defects detection," *J. Sensors*, vol. 2022, pp. 1–12, Nov. 2022, doi: 10.1155/2022/7530361.

[27] K. Jiang, T. Xie, R. Yan, X. Wen, D. Li, H. Jiang, N. Jiang, L. Feng, X. Duan, and J. Wang, "An attention mechanism-improved YOLOv7 object detection algorithm for hemp duck count estimation," *Agriculture*, vol. 12, no. 10, p. 1659, Oct. 2022, doi: 10.3390/agriculture12101659.

[28] Z. Huang, J. Wang, X. Fu, T. Yu, Y. Guo, and R. Wang, "DC-SPP-YOLO: Dense connection and spatial pyramid pooling based Yolo for object detection," *Inf. Sci.*, vol. 522, pp. 241–258, Jun. 2020, doi: 10.1016/j.ins.2020.02.067.

[29] K. Tong and Y. Wu, "Deep learning-based detection from the perspective of small or tiny objects: A survey," *Image Vis. Comput.*, vol. 123, Jul. 2022, Art. no. 104471, doi: 10.1016/j.imavis.2022.104471.

[30] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.

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

[32] H. Liu, F. Sun, J. Gu, and L. Deng, "SF-YOLOv5: A lightweight small object detection algorithm based on improved feature fusion mode," *Sensors*, vol. 22, no. 15, p. 5817, Aug. 2022, doi: 10.3390/s22155817.

[33] L. Kong, J. Wang, and P. Zhao, "YOLO-G: A lightweight network model for improving the performance of military targets detection," *IEEE Access*, vol. 10, pp. 55546–55564, 2022, doi: 10.1109/ACCESS.2022.3177628.

[34] S. Tummala, S. Kadry, S. A. C. Bukhari, and H. T. Rauf, "Classification of brain tumor from magnetic resonance imaging using vision transformers ensembling," *Current Oncol.*, vol. 29, no. 10, pp. 7498–7511, 2022.

[35] L. Meng, H. Li, B.-C. Chen, S. Lan, Z. Wu, Y.-G. Jiang, and S.-N. Lim, "AdaViT: Adaptive vision transformers for efficient image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 12299–12308, doi: 10.1109/CVPR52688.2022.01199.

[36] C. R. Chen, Q. Fan, and R. Panda, "CrossViT: Cross-attention multi-scale vision transformer for image classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 347–356, doi: 10.1109/ICCV48922.2021.00041.

[37] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "DETRs beat YOLOs on real-time object detection," 2023, *arXiv:2304.08069*.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015, doi: 10.1109/TPAMI.2015.2389824.

[39] N.-C. Ristea, N. Madan, R. T. Ionescu, K. Nasrollahi, F. S. Khan, T. B. Moeslund, and M. Shah, "Self-supervised predictive convolutional attentive block for anomaly detection," 2021, *arXiv:2111.09099*.

[40] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, "Focal and efficient IOU loss for accurate bounding box regression," 2021, *arXiv:2101.08158*.

[41] J. Chen, H. Mai, L. Luo, X. Chen, and K. Wu, "Effective feature fusion network in BIFPN for small object detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Anchorage, AK, USA, Sep. 2021, pp. 699–703, doi: 10.1109/ICIP42928.2021.9506347.

[42] H. Sun, Y. Wang, X. Cai, P. Wang, Z. Huang, D. Li, Y. Shao, and S. Wang, "AirBirds: A large-scale challenging dataset for bird strike prevention in real-world airports," in *Computer Vision—ACCV* (Lecture Notes in Computer Science), vol. 13845, L. Wang, J. Gall, T.-J. Chin, I. Sato, and R. Chellappa, Eds. Cham, Switzerland: Springer, 2023, pp. 409–424, doi: 10.1007/978-3-031-26348-4_24.

[43] M. Heydarian, T. E. Doyle, and R. Samavi, "MLCM: Multi-label confusion matrix," *IEEE Access*, vol. 10, pp. 19083–19095, 2022, doi: 10.1109/ACCESS.2022.3151048.

HAIJUN LIANG was born in Shandong, China, in 1983. He received the Ph.D. degree in computer science and technology from Sichuan University, Chengdu, China, in 2014. He is currently an Associate Professor with the Civil Aviation Flight University of China, Guanghan, China. His current research interests include air traffic control and computer vision.

XIANGWEI ZHANG was born in Hebei, China, in 2001. He received the B.S. degree in electrical engineering and intelligent control from the North China Institute of Aerospace Engineering, Langfang, China, in 2022. He is currently pursuing the M.S. degree with the Civil Aviation Flight University of China, Guanghan, China. His current research interest includes object detection.

JIANGUO KONG was born in Shaanxi, China, in 1974. He received the M.S. degree in communications and transportation from Southwest Jiaotong University, Chengdu, China, in 2004. He is currently a Professor with the Civil Aviation Flight University of China, Guanghan, China. His current research interests include air traffic control and deep learning.

ZHIWEI ZHAO was born in Henan, China, in 1997. He received the B.S. degree in engineering from the Civil Aviation Flight University of China, Guanghan, China, in 2019, where he is currently pursuing the M.S. degree. His current research interest includes image recognition.

KEXIN MA was born in Shaanxi, China, in 1999. She received the B.S. degree in engineering from Xi'an Aeronautical Institute, Xi'an, China, in 2021. She is currently pursuing the M.S. degree with the Civil Aviation Flight University of China, Guanghan, China. Her current research interest includes air traffic management.