IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 May 2024, accepted 28 May 2024, date of publication 3 June 2024, date of current version 10 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3408269

RESEARCH ARTICLE

BSM-YOLO: A Dynamic Sparse Attention-Based Approach for Mousehole Detection

TIANSHUO XIE, XIAOLING LUO, AND XIN PAN
School of Computer and Information Engineering, Inner Mongolia Agricultural University, Hohhot 010000, China
Corresponding author: Xiaoling Luo (luoxl@imau.edu.cn)

This work was supported in part by the Inner Mongolia Autonomous Region Natural Science Foundation under Grant 2023LHMS06020.

ABSTRACT In recent years, the proliferation of mousehole in grasslands has exacerbated desertification and compromised grassland productivity, posing potential threats to human safety. Consequently, the identification and forecasting of mouse-hole dynamics for effective infestation control have emerged as pressing concerns. Manual mousehole detection is labor-intensive and time-consuming, hindering comprehensive spatial understanding. Moreover, prevailing detection models lack robust feature extraction for small targets like mousehole, resulting in suboptimal recognition capabilities and diminished accuracy. Addressing these challenges, we propose an enhanced one-stage detection model BSM-YOLO based on YOLOv5 architecture. Firstly, the model integrates a BiFormer module leveraging Bi-Level Routing Attention to capture both global and local features within mousehole images. Subsequently, the incorporation of Shuffle Attention mechanisms enhances the learning of feature dependencies and intricate relationships. Lastly, the adoption of the MPDIoU loss function accurately delineates bounding box characteristics, mitigating redundant box generation and expediting model convergence. In our experimental framework, we curated a dataset comprising 2397 mousehole images to train the BSM-YOLO model. Results indicate that the BSM-YOLO model achieves an average detection accuracy of 94.5%, representing a 5.4% enhancement over the baseline YOLOv5s model. Additionally, the model demonstrates an 8.7 f/s improvement in detection speed. Furthermore, ablation experiments confirm the efficacy of each refinement incorporated into the BSM-YOLO model.

INDEX TERMS YOLOv5, object detection, deep learning, mousehole.

I. INTRODUCTION

Rodents are widely distributed in the grassland area of Inner Mongolia, China, and the population crisis of rodents has frequently erupted due to climate change and other factors. Their random burrowing activities can damage grasslands, as well as pose a threat to local livestock production [1], sanitary and epidemiological defense, and ecological environment construction. Brandt's vole(BV) is a major pest in Inner Mongolian grassland systems. It is a small, seasonally breeding rodent, and due to the complex burrow system excavated by the BV, the grassland in the study area is degraded into a habitat that is increasingly suitable for its life [2]. By accurately detecting rodent holes, mouse activities can be detected and controlled in a timely manner to inhibit the growth of rodent populations in order to maintain ecological balance. Meanwhile, by analyzing the spatial distribution characteristics of rat holes, it helps us to select appropriate rat extermination programs and strategies, so as to improve the efficiency of rat extermination.

The earliest methods for mouse hole detection were primarily manual, involving techniques such as fixed-point observation, marking recapture, and day and night surveys [3], [4]. These methods are labor-intensive, time-consuming, and cost-prohibitive, limiting their applicability to small study areas and failing to provide comprehensive coverage of larger regions. With the advancement of unmanned aerial vehicle (UAV) technology and the evolution of machine learning algorithms, researchers have turned to UAV imagery

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

combined with machine learning-based target detection algorithms for mouse hole detection. Traditional target detection algorithms, such as the sliding window-based Viola-Jones (VJ) detector [5], [6], Histograms of Oriented Gradients (HOG) detector [7], and Deformable Parts Model (DPM) detector [8], rely on hand-crafted features and machine learning techniques. While these methods adequately address target detection requirements, they suffer from certain limitations. The sliding window approach entails excessive computations, leading to slow detection speeds. Moreover, hand-designed feature extractors exhibit varying adaptability and robustness across different targets and environments, proving less effective in scenarios with complex backgrounds and occlusions.

Since the introduction of Convolutional Neural Networks (CNNs) in 2012, target detection has transitioned into the era of deep learning. CNN-based target detection methods can be broadly classified into two categories: two-stage target detection algorithms and one-stage target detection algorithms. Two-stage algorithms typically involve region proposal as the initial step, where bounding boxes likely to contain the target object are generated. Subsequently, these regions are classified by a convolutional neural network. However, two-stage algorithms are often criticized for their slow detection speed and tendency to produce false positives, leading to reduced detection accuracy. In response to these limitations, researchers have developed one-stage target detection algorithms, which dispense with the separate region proposal step and directly extract features within the network to predict object classification and localization. By integrating candidate box generation, classification, or regression into a single step, one-stage algorithms significantly reduce computational overhead, resulting in faster detection. Moreover, these algorithms typically employ an end-to-end training approach, enabling direct learning of the mapping from input images to target location and category, thereby enhancing generalization capability. Despite their advantages in terms of speed and model simplicity, one-stage detection algorithms still have shortcomings. They may struggle with detecting small targets due to the limited amount of feature information extracted, leading to challenges in accurately detecting and recognizing such targets. Additionally, one-stage algorithms often generate a large number of candidate bounding boxes, necessitating post-processing to remove redundant boxes. This process may inadvertently eliminate actual targets, thereby lowering mean Average Precision (mAP) and Recall (R).

To solve the above problems, we propose a mousehole detection model BSM-YOLO, and the main contributions of this paper are as follows:

a) To address the model's limited feature extraction capacity for small targets, we integrate the BiFormer module. This module captures both global and local features from mouse hole images, thereby enhancing the detection accuracy of small targets and improving the model's capability to adapt to multi-scale features.

b) In order to facilitate the model in learning intricate feature relationships and mitigating background interference in mousehole detection, we introduce the Shuffle Attention (SA) mechanism. SA effectively enhances the model's feature representation capacity.

c) To provide a more precise characterization of bounding box location and shape while simplifying computational complexity, we adopt the Minimum Points Distance Intersection over Union (MPDIoU) loss function. This replacement enhances the model's detection performance for targets with varying sizes and aspect ratios.

The remainder of the paper is structured as follows: Section II provides an overview of related work in the field of target detection. In Section III, the architecture of our proposed BSM-YOLO model is detailed. Section IV outlines the experimental setup and presents an analysis of the experimental results. Lastly, Section V summarizes the key conclusions drawn from this study.

II. RELATED WORK

In the realm of computer vision, target detection stands as a prominent research focus due to its extensive application domains and profound research significance. Over time, notable advancements have been achieved in this area. Target detection algorithms are predominantly classified into two categories: two-stage target detection algorithms and one-stage target detection algorithms. This section offers an analysis and synthesis of the research advancements within these two categories of target detection algorithms.

A. TWO-STAGE TARGET DETECTION ALGORITHM

The two-stage target detection algorithm adopts a coarse-to-fine network structure. The initial stage involves coarse detection, where candidate regions are generated using a region generation algorithm. Subsequently, in the fine detection stage, candidate frames retained from the previous stage undergo categorization and positional adjustment to determine category probabilities and target positions. This algorithm, characterized by its precision in generating candidate frames, yields high accuracy and reduces the false detection rate. In 2014, Girshick et al. [9] introduced a target detection algorithm based on Region-based Convolutional Neural Networks (R-CNN). This approach employs a selective search strategy to extract target candidate frames from the input image, followed by feature extraction using convolutional neural networks, and subsequent training of a classifier for target classification. However, R-CNN's requirement for independent feature extraction for each candidate area results in extensive repeated computations, thereby impeding detection speed. To address this limitation, He et al. proposed Spatial Pyramid Pooling in Deep Convolutional Networks (SPP-NET) in 2014 [10]. SPP-NET utilizes a designed spatial pyramid network to circumvent compatibility issues with input image size encountered in traditional convolutional neural networks. This model can convert an arbitrarily sized feature map into a fixed-

sized feature vector without necessitating image cropping or scaling during processing. SPP-NET requires only a single convolutional computation for the entire image, mitigating the issue of redundant convolutions in the same region and thereby alleviating the problem of repeated computations, thus significantly enhancing detection speed. In 2015, Fast-RCNN [11] was introduced as an optimized version of R-CNN. Fast-RCNN removes the SVM [12] in the R-CNN detection head classifier and employs a convolutional neural network along with a Softmax layer for feature extraction, classification, and bounding box regression. To address the more pronounced issue of the time-consuming R-CNN model, Ren et al. proposed Faster-RCNN [13] in 2017. Faster-RCNN utilizes a CNN-based region proposal network (RPN) to replace the conventional selective search algorithm. This approach integrates tasks such as candidate region generation and subsequent classification regression within a single convolutional neural network, enabling end-to-end training and testing with exceptionally high detection accuracy. The two-stage based detection algorithm introduces convolutional neural network into the field of target detection, which changes the main research idea of target detection task and greatly improves the target detection effect. However, the two-stage detection model still cannot satisfy the demand of real-time detection.

B. ONE-STAGE TARGET DETECTION ALGORITHM

One-stage target detection algorithms employ a regression strategy for target detection, where, given an input image, a prediction frame is directly generated, and the category and location of the predicted object are calculated within this frame. In recent years, researchers have introduced various one-stage target detection algorithms, successfully applying them to real-time detection tasks. Prominent examples include the YOLO series algorithms and the Single Shot MultiBox Detector (SSD). The SSD model, proposed by Liu et al. [14], implements a multi-scale target detection strategy. This approach enables the detection of targets with varying sizes by extracting feature maps at multiple scales, thereby enhancing the model's ability to detect objects across different scales. Moreover, with continuous updates and iterations in the YOLO series algorithms, detection accuracy has steadily improved. For instance, the YOLOv5 algorithm, introduced in 2020 [15], not only exhibits rapid detection speed but also achieves significantly enhanced detection accuracy. This algorithm strikes an optimal balance between speed and accuracy, making it particularly advantageous in scenarios with high real-time demands. In pursuit of even faster detection speed and higher accuracy, Wang et al. [16] proposed the YOLOv7 algorithm in 2022. Leveraging a more effective aggregation network and novel training methods, YOLOv7 further enhances speed and accuracy. In the same year, Li et al. [17] introduced the YOLOv6 framework, which presents a range of deployable networks of varying sizes to accommodate diverse application scenarios. It adopts the structure-heavy parameterization method of RepVGG to implement a multi-branch structure during the training phase and a planar architecture during the inference phase, striking a balance between speed and accuracy. Furthermore, it introduces a hybrid channel strategy to construct more effective decoupling heads and reduce the number of intermediate convolutions, thus enhancing detection efficiency. However, as the model capacity expands, the computational overhead and parameter count of YOLOv6 gradually increase, potentially limiting its applicability in certain scenarios. To address this challenge, Reis et al. [18] proposed the YOLOv8 target detection algorithm in 2023. This algorithm achieves a harmonious balance between speed and accuracy through an innovative composite scaling method, enabling the adjustment of model size according to specific application requirements. This feature empowers YOLOv8 with remarkable adaptability across different scenes and targets, while further enhancing its generalization capability. Nevertheless, due to its deeper network structure and increased parameter count, YOLOv8 requires relatively longer training times, which may impede rapid model iteration and experimental efficiency.

In recent years, owing to the flourishing advancement of one-stage target detection models, an increasing number of researchers have been employing them to address challenges in agricultural production, automated driving, and rodent hole detection, yielding promising outcomes. Zhang et al. [19] proposed the YOLOv5 network to tackle the issue of detecting small targets in images, specifically targeting cherry fruit recognition. By integrating BiFPN and shallow sub-sampling, this model enhances the efficiency and accuracy of feature fusion during the recognition process. Similarly, Yang and Fan introduced the YOLOv8-Lite model [20] to address the stringent real-time demands of autonomous driving technology. Utilizing the FastDet structure, TFPN pyramid structure, and CBAM attention mechanism, this model effectively enhances both performance and efficiency to meet the real-time requisites of autonomous driving technology. Moreover, Du et al. [21] proposed a framework that combines a UAS image acquisition platform with deep learning techniques. They employed six deep learning target detection models (comprising three two-stage and three one-stage models) on a two-season UAS Bush vole mouse burrow image dataset, demonstrating the efficacy of their approach. The findings indicate that the one-stage models Faster R-CNN and YOLOv4 demonstrate high-precision detection capability for Bristol's vole mouse holes in UAS images. Cui et al. [22] employed the YOLOv3 network and its lightweight YOLOv3-tiny variant to re-cluster the number and aspect ratio dimensions of target candidate frames in a large gerbil hole dataset, achieving precise identification and localization of the holes. However, due to the limited quantity of shaped holes in the dataset, it failed to adequately capture the features of holes of various morphologies, resulting in leakage issues that impacted detection accuracy. On the other hand, Li et al. [23] introduced the CGT-YOLOv5n model

to enhance the detection of small-target mouse holes and reduce interference from obstacle shadows. By incorporating CAM, ODConv, and TSCODE modules, this model improves the detection accuracy of mouse holes in complex grassland environments. Nevertheless, it still grapples with challenges related to insufficient feature extraction of small-target mouse holes, susceptibility to interference information, and limited model robustness.

In conclusion, the one-stage detection algorithm has emerged as a prominent research focus in the domain of target detection owing to its rapid detection speed. However, as the model structure deepens and the number of parameters increases, enhancing experimental efficiency while ensuring detection accuracy remains a pressing challenge in this field.

III. METHOD

This section first briefly describes the overall structure of the BSM-YOLO model and then details a series of improvement modules.

A. BSM-YOLO FRAMEWORK

YOLOv5 is a target detection model developed by Ultralytics, which employs a PyTorch-based architecture with high flexibility and performance. The YOLOv5 model achieves both fast detection speed and high recognition accuracy. The YOLOv5 model is n, s, m, l, and x in ascending order of size, depth, and width. In this paper, we propose a BSM-YOLO baseline model is YOLOv5s. We make three significant improvements on the baseline model: first, we introduce the BiFormer module to achieve high-accuracy small-target detection; second, we introduce the SA mechanism to achieve the enhanced ability of the model to learn the complex relationship between features and to improve the feature representations of the model; and lastly, we optimize the model by using the MPDIoU loss function.

The general framework diagram of BSM-YOLO is illustrated in Fig. 1. Initially, the input side receives data with an image size of 640 × 640 pixels and undergoes preprocessing, including image normalization and channel order adjustment. To enhance the model's generalization ability, mosaic data augmentation is applied to the input side, incorporating random scaling, cropping, flipping, color dithering, etc., thereby augmenting dataset diversity and richness. Subsequently, the image traverses multiple stacked C3 modules and multiple CBS convolutional layers, followed by entry into the BiFormer module based on Bi-Level Routing Attention (BRA). This module employs a two-layer routing attention mechanism to capture both global and local feature information, facilitating high-precision detection of small targets. Proceeding further, after traversing the feature pyramid network, the input image proceeds to the SA mechanism, which effectively learns feature dependencies and complex relationships, thereby enhancing model performance expression and thoroughly extracting target features. Ultimately, the MPDIoU loss function is employed to expedite convergence speed and bolster model robustness.

B. BiFormer

BiFormer serves as a visual transformation backbone founded on dynamic sparse attention, facilitating more adaptable content-aware computational allocation through a two-layer routing mechanism. At the heart of the BiFormer module lies the BRA [24], which orchestrates the model's effective processing of global and local information via a dual-tier routing attention mechanism. Illustrated in Fig. 2, BAR partitions the input image into S × S non-overlapping regions, with each region housing $\frac{HW}{S^2}$ feature vectors. The input image, denoted as $X \in R^{H\times W\times C}$, comprises a height H, width W, and C channels. Upon reshaping, X transforms into $X^r \in R^{S^2\times\frac{HW}{S^2}\times C}$. Subsequently, the linear projection yields $Q, K, V \in R^{S^2\times\frac{HW}{S^2}\times C}$ through mapping, where Query, Key, and Value undergo the following linear projection:

$$Q = X^r W^q, K = X^r W^k, V = X^r W^v \tag{1}$$

ncluded among these $W^q, W^k, W^v \in X^{C\times C}$.

Subsequently, we formulate directed graphs to identify region-to-region attention relations, signifying regions of attention for each designated region. Initially, the region-level $Q^r, K^r \in R^{S^2\times C}$ are obtained by computing the average value of each region on Q and K, respectively. Then, by utilizing the adjacency matrix of correlations between regions Q^r and K^r:

$$A^r = Q^r (K^r)^T, A^r \in S^{S^2\times S^2} \tag{2}$$

The elements within the adjacency matrix A^r quantify the semantic correlation between the two regions. This generates a global representation containing crucial features and comprehensive information about the entire image.

The subsequent crucial step involves retaining, for each region, only the first K most closely associated regions. To achieve this, a routing index matrix $I^r \in R^{S^2\times K}$ is obtained, which retains the indices of the first K connections row-wise.

$$I^r = topkIndex(A^r) \tag{3}$$

The $i-th$ row of I^r comprises the indices of the first K most relevant regions to the $i-th$ region. Subsequently, a localized representation is produced, containing detailed features and information regarding the image block and its adjacent K regions.

Finally, the fine-grained token-to-token attention is computed solely using the region-to-region routing index matrix I^r, effectively filtering out the least relevant tokens at the coarse-grained level. This approach reflects the sparsity design of BRA, which reduces computational overhead by focusing on only a small portion of the most relevant image chunks or features during computation. For each Query token in region i, attention is directed to the K most attention routing regions indexed by $I^r_{(i,1)}, I^r_{(i,2)} \ldots I^r_{(i,k)}$, gathering all K and V

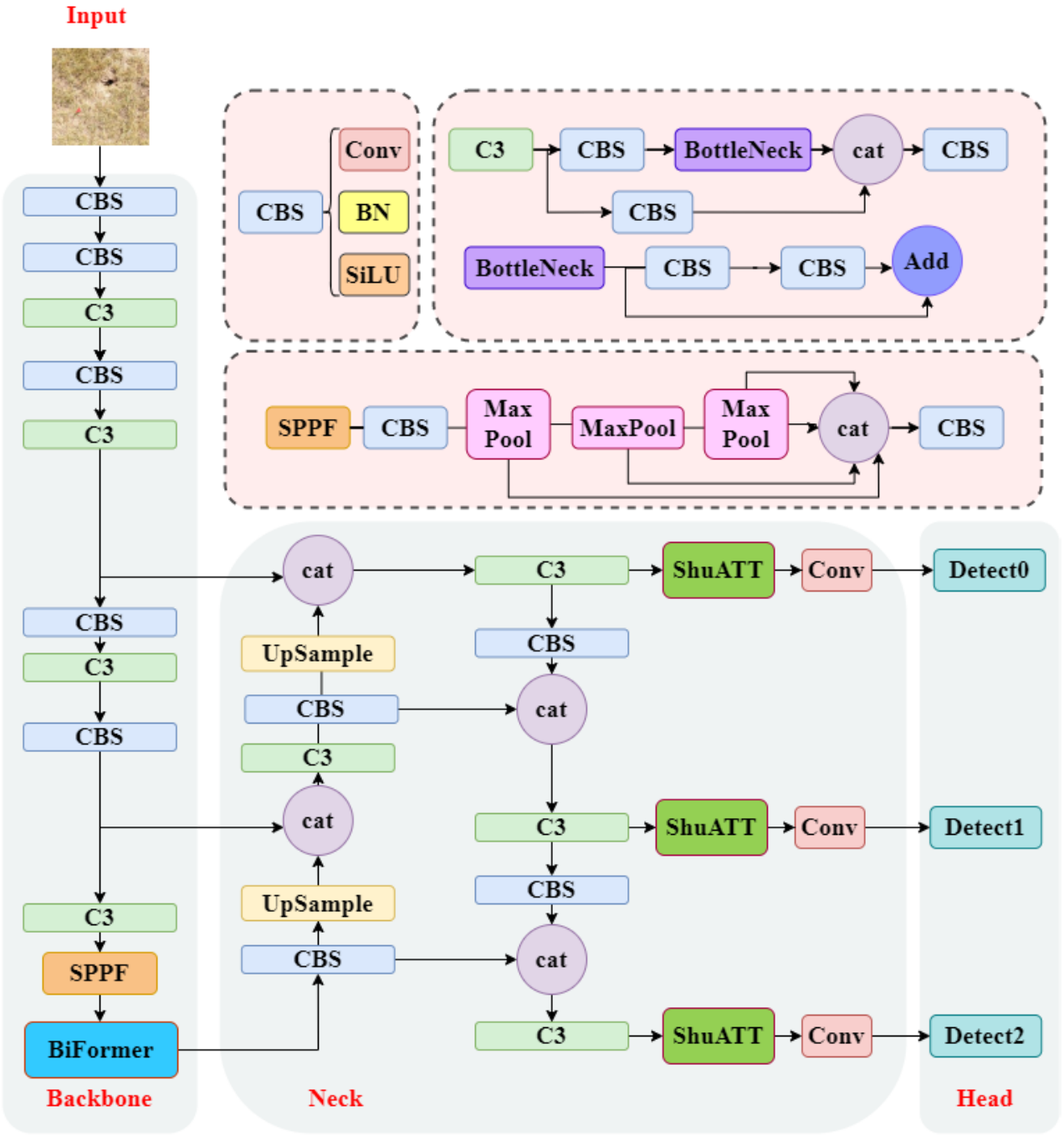

FIGURE 1. Illustrates the network structure of the BSM-YOLO model. The CBS module comprises Convolutional layers, Batch Normalization, and SiLU activation functions. The BottleNeck module features a residual connection structure. The SPPF module follows the CBS layers and integrates pooling kernels of various sizes.

in these regions. The formulas for collecting the K and V tensors are as follows:

$$K^g = gather(K, I^r), V^g gather(V, I^r) \tag{4}$$

where $K^g, V^g \in R^{S^2 \times \frac{HW}{S^2} \times C}$, and then apply attention to collecting K^g,V^g:

$$O = Attention(Q, K^g, V^g) + LCE(V) \tag{5}$$

Here a local context enhancement LEC(V) [25] is introduced and the function LCE is parameterized using deep convolution with kernel size set to 5.

The BiFormer, constructed based on the BRA module, adopts a four-stage pyramid structure. In the first stage, the image is segmented into small patches and converted into vectors using the overlap block and the BiFormer block. Subsequently, in the second to fourth stages, the subsampling

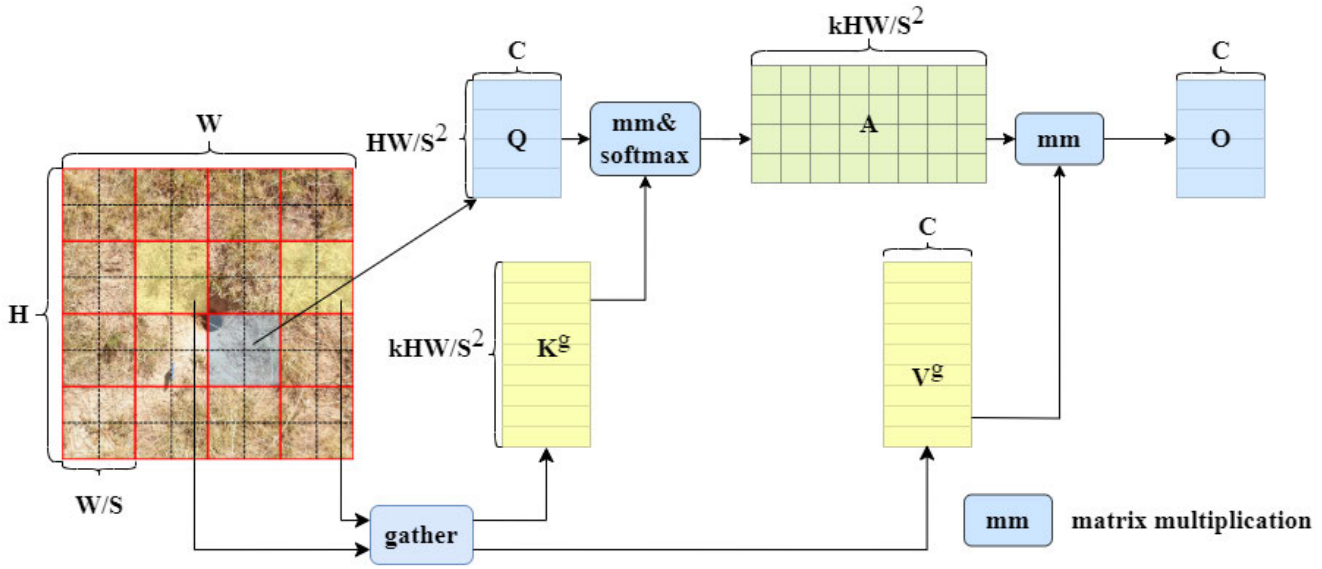

FIGURE 2. BRA structure diagram.

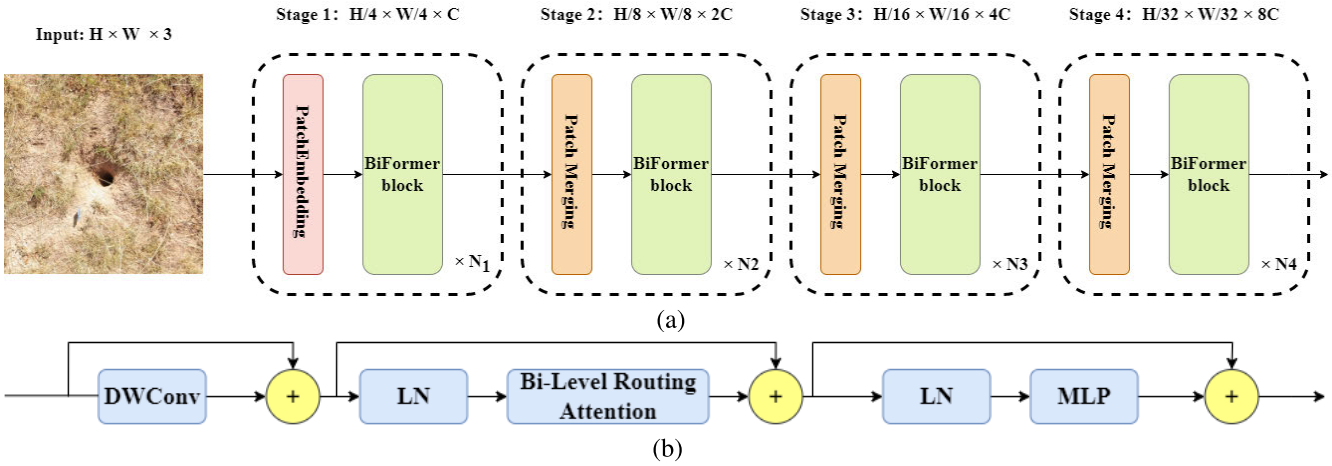

FIGURE 3. (a) BiFormer structure (b) BiFormer block details.

and the BiFormer block are combined to reduce the spatial resolution of the input image and increase the number of channels. The changes in the size of the image and the number of channels are illustrated in Fig. 3(a). The detailed structure of the BiFormer block is depicted in Fig. 3(b). The BiFormer block comprises two key components: the upper router and the lower router, both included in the BRA. The upper router captures global contextual information, while the lower router focuses on the details of the local region. This mechanism enhances the performance of the BiFormer model in the rat-hole detection task.

C. SHUFFLE ATTENTION

Attention mechanisms such as SE [26], CBAM [27] and ECA [28] have the limitation of paying attention to the weights on the channel or spatial dimension, ignoring the relationship between the channel and spatial dimension. All of them on them have the problems of high computational complexity and cannot fully capture the information of spatial dimension, etc. The SA mechanism [29] can effectively solve the above problems by grouping the input features, randomly arranging them, attentional weight computation, and feature recombination and other steps.

As shown in Fig. 4, first, SA groups the input feature maps according to the channel dimension. Assuming that the number of channels of the input feature map is c, it can be divided into g groups, each group contains c/g channels, and then each feature X_k will be split into two branches along the channel dimension, one branch X_{k1} is used to learn channel attention features, and the other branch X_{k2} is used to learn spatial attention features. The channel attention branch goes through the steps of global average pooling, fully connected

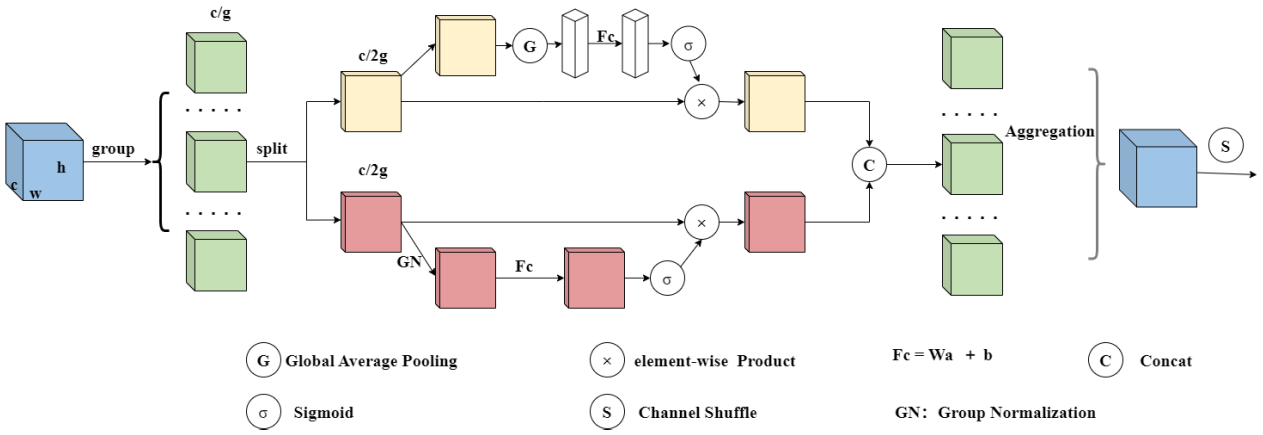

FIGURE 4. Shuffle attention.

layers, and Sigmoid activation function, and the model learns the importance of each channel for the mouse hole detection task. Next, the channel attention weights obtained through element-wise product are used to weight the channel features within each group, enriching the contextual information and inter-channel dependencies of the weighted features.

The spatial attention branch identifies critical spatial locations for the mouse hole detection task using group normalization (GN), fully connected layers, and Sigmoid activation functions. Spatial attention weights, which highlight important spatial locations in feature maps, are calculated using element-wise product operations. GN reduces training instability by normalizing features within groups. By emphasizing significant spatial locations in the input feature map, the spatial attention mechanism enhances the model's feature representation and performance.

Next, all features from both the spatial and channel branches are integrated through concatenation, and the input feature map is recombined into g groups. At this stage, each group's information remains independent, posing a challenge for interaction and integration between groups. The SA mechanism addresses this issue with a channel shuffle operation. Channel shuffle disrupts the independence of groups by rearranging the channels, enabling information flow across groups and enhancing the model's expressive capability. This operation allows each convolutional layer to access channel information from different groups, thereby increasing the diversity and richness of features. Combined with grouped convolution, channel shuffling achieves better performance while maintaining low model complexity. This simple yet effective operation significantly improves the SA mechanism's performance and applicability by breaking channel independence, enhancing feature representation, improving information flow, and increasing computational efficiency.

D. LOSS FUNCTION

The baseline model employs a loss function that characterizes the horizontal-to-vertical ratio of the prediction box in relative terms, potentially affecting the model's accuracy in predicting target size and shape. In contrast, the BSM-YOLO model utilizes the MPDIoU loss function [30]. This function enhances the network's capacity to learn bounding box location and shape by maximizing the Distance-IOU value between two bounding boxes. Unlike other loss functions, MPDIoU considers various factors such as centroid distances, width, and height deviations, providing a more comprehensive assessment of bounding box similarity. Moreover, the MPDIoU loss function streamlines the computation process, facilitating faster calculation of loss values during training and accelerating model convergence. Unlike existing loss functions, MPDIoU can optimize predictions even when the aspect ratio of the predicted bounding box matches that of the real bounding box, but their width and height values are markedly different, as illustrated in Fig. 5. For instance, if there exists a certain real box with dimensions (W_{gt}, H_{gt}), and two predicted boxes with centers coinciding with it, namely ($\frac{W_{gt}}{k}$, $\frac{H_{gt}}{K}$) and (kW_{gt}, kH_{gt}), MPDIoU sets the predicted box as located inside the real box if it lies below, whereas it flags the predicted bounding box as outside the true labeled bounding box.

We understand that a unique rectangle is defined by the coordinates of its upper-left and lower-right points, a property leveraged by MPDIoU to expedite bounding box regression by directly minimizing the distances between the upper-left and lower-right points of the predicted and real labeled bounding boxes. Illustrated in Fig. 6, let's consider a scenario where a real labeled box has upper-left corner coordinates of (x_1^{gt}, y_1^{gt}) and lower-right corner coordinates of (x_2^{gt}, y_2^{gt}), alongside a predicted box with upper-left corner coordinates of (x_1^{pre}, y_1^{pre}) and lower-right corner coordinates of $(x_2^{p}re, y_2^{pre})$. Here, d_1 represents the straight-line distance between the two upper-left corner coordinates, while d_2 denotes the straight-line distance between the two lower-right corner coordinates. MPDIoU is expressed as follows:

$$MPDIoU = IoU - \frac{d_1^2}{w^2 + h^2} - \frac{d_2^2}{w^2 + h^2} \tag{6}$$

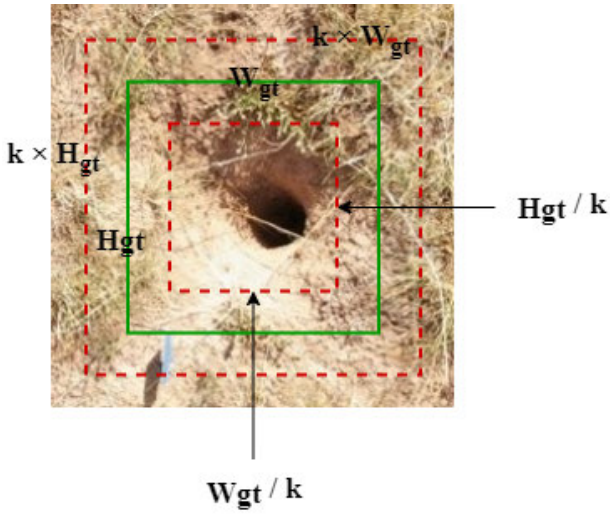

FIGURE 5. Unable to optimize the situation.

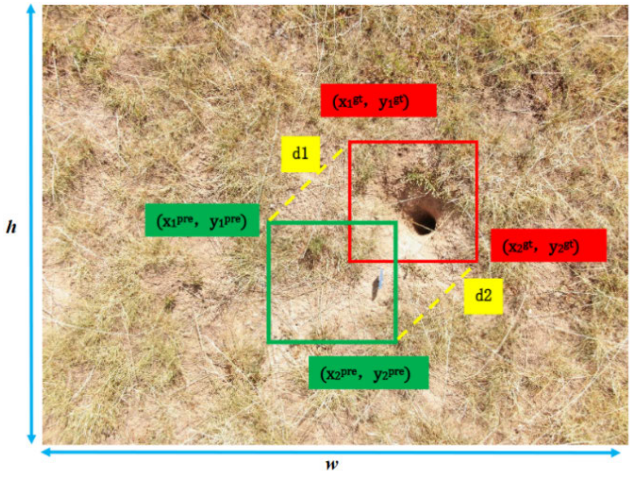

FIGURE 6. Unable to optimize the situation.

MPDIoU Loss can be defined based on MPDIoU:

$$L_{MPDIoU} = 1 - MPDIoU \tag{7}$$

IV. EXPERIMENTATION AND ANALYSIS

A. DATASET

The mousehole image data were collected between 2022 and 2023 during the summer months of July and the winter months of November in the Shilamuren Grassland. Following a preliminary survey, BV were identified as the predominant rodent species in the Shilamuren Grassland. Their burrows typically exhibit diameters ranging from 4 to 6 cm, often accompanied by darker-colored soil or mounds in close proximity. The imaging equipment utilized for data collection was the DJI-Royal Mavic2 Zoom drone, featuring a 12-megapixel effective camera pixel, a focal length of 28 mm, and an approximate lens angle of view of 83°. Prior to photography, the holes were manually visually interpreted and marked as belonging to BV burrows. A total of 2,397 photographs of mouse-holes were captured under natural environmental conditions employing a multi-angle photography technique. Subsequently, the rat hole images underwent labeling using LabelImg, with the labeling results saved in the PASCAL VOC format. Before the training process, the VOC files were converted into TXT files.

TABLE 1. Environment configuration.

Parameters	Configuration
operating system	Windows 11
CPU	Intel(R) Core(TM) i5-12600KF @3.70GHz
GPU	NVIDIA GeForce RTX 3080
experimental environment	CUDA11.3,Python3.8

TABLE 2. Main training parameters.

Parameters	Values
input image size	(640,640)
epoch	100
batch size	16
optimizer	SGD
learning rate	0.01
momentum	0.9
weight decay	0.0005

B. EXPERIMENTAL SETTINGS

The experiments were conducted under specific environment configurations including Window 11 operating system, Intel(R) Core(TM) i5-12600KF @3.70GHz, NVIDIA GeForce RTX 3080 and CUDA11.3, Python3.8, etc. See Table 1 for details.

We configure the hyper-parameters prior to the experiments to achieve optimal model performance. To mitigate data bias and errors in dataset distribution, we employ a mosaic enhancement strategy to enhance the diversity of training samples. The experiments use SGD as an optimizer to train the models with an input image size of 640 × 640 pixels, 100 iterations of each model are trained, the batch size is set to 16. The initial learning rate is set to 0.01, the SGD optimizer accelerates the training by learning the rate momentum, which is set to 0.9, and weight decay is used to prevent the model from overfitting, which is set to an initial value of 0.0005. See Table 2 for details.

C. EVALUATION MATRIX

To evaluate the performance of our proposed BSM-YOLO model, we use four evaluation metrics: Precision (P), Recall (R), mean Average Precision (mAP), and Frames Per Second (FPS). True Positive (TP) indicates the number of positive samples that were correctly detected, False Positive (FP) indicates the number of negative samples incorrectly detected as positive samples, and False Negative (FN) indicates the number of positive samples incorrectly detected as negative

TABLE 3. Model performance comparison.

Models	P/%	R/%	mAP@0.5/%	FPS/f·s^{-1}	Para/M	GFLOPs
SSD	65.8	59.2	63.5	54.6	29.2	34.2
Faster-RCNN	78.1	75.7	76.7	12.5	135.6	**15.4**
YOLOv3	72.9	76.0	74.6	53.2	62.6	117.1
YOLOv5s	86.7	84.4	89.1	81.5	7.3	15.9
YOLOv6	75.1	73.2	74.3	68.8	**6.6**	29.1
YOLOv7	76.8	75.7	78.6	72.9	36.9	104.6
YOLOv8	76.3	74.4	76.5	64.6	11.1	28.4
BSM-YOLO	**93.7**	**91.6**	**94.5**	**90.2**	8.2	57.4

samples, and then the formulas for P, R, mAP, and FPS are as follows:

$$P = \frac{TP}{TP + FP} \tag{8}$$

$$R = \frac{TP}{TP + FN} \tag{9}$$

$$mAP = \frac{1}{n} \sum_{i \in (0,1,2 \ldots i)}^{n} \rho(i) \tag{10}$$

$$FPS = \frac{1}{AverageProcessingTime} \tag{11}$$

Precision is the ratio of the number of positive samples labeled as positive to the number of samples detected as positive. Recall is the ratio of the number of samples computed labeled as positive to the number of actual positive samples. mAP is the average of the APs computed for all categories. In addition to this, we use the number of detection frames per second to evaluate the real-time detection performance of the model.

D. COMPARISON EXPERIMENT

In order to assess the feasibility, realism, and robustness of the BSM-YOLO model proposed in this study, we conducted training and testing using our customized dataset. The experimental parameters were kept consistent across classical mainstream target detection models such as SSD, Faster R-CNN, YOLOv3, YOLOv6, YOLOv7, and the latest model, YOLOv8. Evaluation metrics utilized for comparing the experiments encompass P, R, mAP, FPS, number of parameters, and GFLOPs. As depicted in Table 3, our proposed model demonstrates enhancements in both detection accuracy and inference speed.

Compared to SSD and Faster R-CNN, BSM-YOLO exhibits notable enhancements across various metrics. P sees a boost of 27.9% and 17.9%, R improves by 32.4% and 13.5%, mAP increases by 31% and 18.8%, and FPS rises by 35.6 f·s^{-1} and 77.7 f·s^{-1}, respectively. YOLOv3, YOLOv6, and YOLOv7 all performed worse on mAP compared to the baseline model. When compared to YOLOv3, YOLOv6, and YOLOv7, BSM-YOLO showcases marked improvements in P, R, and FPS, along with a 19.9%, 20.%, and 15.9% mAP increase.

However, YOLOv8 demonstrates a 12.6% lower mAP compared to the baseline model, likely due to its complex network architecture, which demands higher computational resources for optimal performance. Moreover, since mouse hole detection involves small target detection, YOLOv8 underperforms compared to algorithms tailored for such scenarios. In contrast, BSM-YOLO surpasses YOLOv8, boasting a 17.4% increase in P, 17.2% in R, 18% in mAP, and 25.6 f·s^{-1} in FPS. To enhance the prediction accuracy and robustness of the BSM-YOLO model while mitigating dataset distribution errors and data bias during validation, Test Time Augmentation (TTA) was implemented. Enabling TTA resulted in a slight increase in inference time, from 5.2 ms to 7 ms, while improving detection accuracy by 2%. TTA effectively enhances the predictive performance of the BSM-YOLO model by augmenting the diversity of test data without adding complexity to the model.

To visually illustrate the performance enhancement in rat-hole detection, we present Loss and mAP curves for the baseline model versus BSM-YOLO in Figure 7. Figure 7(a) illustrates the faster convergence of BSM-YOLO, while Figure 7(b) highlights its significant mAP improvement, specifically by 5.4%. The improvements in P, R, and FPS are also evident from Table 3.

The incorporation of the BiFormer and SA mechanisms in the BSM-YOLO model leads to an expansion in model parameters and a reduction in computational efficiency. However, despite these changes, the model achieves optimal results in terms of mAP. The introduction of these modules inevitably results in an augmentation of model parameters and an escalation in computational complexity. Specifically, the BSM-YOLO model experiences an increase in model parameters by 0.9 M and a threefold rise in FLOPs compared to the baseline model. Subsequent efforts should focus on mitigating the computational complexity of the model while preserving the current mAP performance.

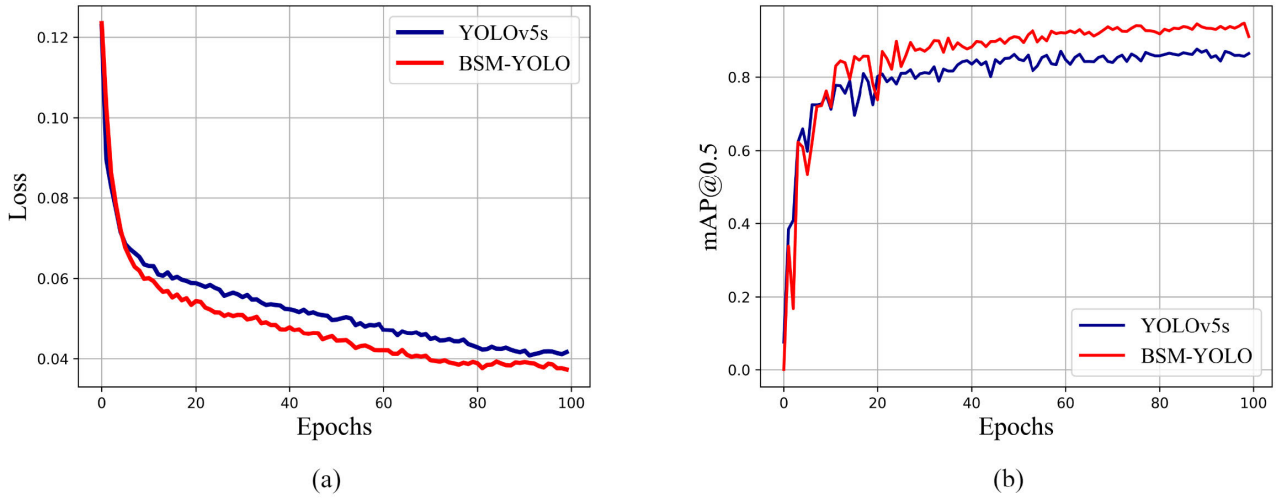

FIGURE 7. Illustrates the comparison of model performance before and after improvement. Subfigure (a) presents the change curves of the loss function, while subfigure (b) depicts the mAP@0.5 curves before and after improvement.

TABLE 4. Ablation experiments.

Num	BiFormer	SA	MPDLoss	P/%	R/%	mAP@0.5/%
YOLOv5s				87.8	84.5	89.1
-	✓			88.1	87.1	90.9(↑1.8)
-		✓		90.2	90.7	91.7(↑2.6)
-			✓	89.4	88.2	89.3(↑0.2)
-	✓	✓		92.1	91.6	93.2(↑4.1)
BSM-YOLO	✓	✓	✓	93.4	92.6	**94.5(↑5.4)**

E. ABLATION EXPERIMENTS

In order to demonstrate the performance contribution of each proposed improvement to BSM-YOLO, we perform a series of ablation experiments, the results of which are presented in Table 4.Through the ablation experiments, we evaluate the impact of each improvement point on the model.

The introduction of the BiFormer module to the baseline model led to an improvement in mAP by 1.8%, P by 0.3%, and R by 2.6%. Furthermore, when the BiFormer module was combined with the SA mechanism, P increased by 4.2%, R by 7.1%, and mAP by 4.1%. These results demonstrate that the addition of the BiFormer module enhances the model's ability to detect small targets, resulting in a significant overall improvement in detection accuracy. Upon introducing the SA mechanism to the baseline model alone, there was an improvement of 2.4% in P, 6.2% in R, and 2.6% in mAP. This experimentation confirmed the SA mechanism's superior recognition ability for rat-hole target detection, contributing to a notable enhancement in the model's detection accuracy by better capturing the intricate relationship between features. Replacing the loss function with MPDIoU led to improvements in P, R, and mAP, albeit not significantly. The primary impact of the loss function change lies in accelerating the model's convergence speed. Overall, the BSM-YOLO model, incorporating the aforementioned three improvements, demonstrated a remarkable enhancement in mAP by 5.4% compared to the baseline model, alongside improvements in P by 5.6% and R by 8.1%. Figure 8 illustrates the performance comparison between the baseline model (Fig. 8(a)(c)) and BSM-YOLO (Fig. 8(b)(d)) across various detection scenarios. Notably, BSM-YOLO exhibits enhanced confidence in detecting individual mouse holes under weed shading, improved detection of multiple mouse holes without weed cover, and reduced false detections of multiple mouse holes in the presence of weed cover.

In summary, our proposed BSM-YOLO model achieves a high average detection accuracy in the Brandt's vole mouse burrow detection task, and the ablation experiments demonstrate the effectiveness of each part of the improvement on the model.

FIGURE 8. Comparison of detection results before and after improvement: (a) (c)represents the performance of the unimproved YOLOv5s algorithm, while (b)(d) illustrates the performance of our proposed BSM-YOLO.

V. CONCLUSION

Rodent hole detection plays a crucial role in rodent pest management. To achieve precise identification and efficient detection of small target rodent holes, this study introduces a rodent hole detection model based on BSM-YOLO. The BSM-YOLO model addresses challenges related to inadequate feature extraction and cannot accurate differentiation between target and background in small target detection. Experimental results demonstrate that the algorithm exhibits superior performance in terms of accuracy, robustness, and

practicality. Despite these advancements, we acknowledge the potential for further enhancement of the BSM-YOLO algorithm in mouse hole detection. The proposed model entails an increased number of parameters and heightened computational complexity following the addition of modules. Consequently, our future efforts will focus on reducing computational overhead and model size by employing a lightweight backbone network while maintaining detection accuracy.

REFERENCES

[1] L. M. Hua and S. Q. Chai., "Current situation, problems and countermeasures of rodent prevention and control in grassland in China," *Acta Phytophylacica Sinica*, vol. 49, pp. 415–423, Jan. 2022.
[2] W. Wang, Q. S. Feng, H. Yu, and T. G. Liang, "Application of '3S' technology in monitoring and prediction of rodent and insect pests in grasslands," *Pratacultural Sci.*, vol. 27, pp. 31–39, Jan. 2010.
[3] W. Liu, W. Q. Zhong, and D. H. Wang, "Seasonal patterns of survival of long-clawed gerbil populations and their dynamic mechanisms in the agro-pastoral zone of Inner Mongolia," *Acta theriologica sinica*, vol. 40, pp. 571–584, 2020.
[4] L. Q. Wang, H. Wang, F. S. Zhang, Y. P. Yang, Y. F. Xie, and W. H. Dong, "Annual trends in rodent community composition around the eastern edge of the Kubuqi Sandlands," *Chin. J. Grassland*, vol. 57, pp. 154–159, 2020.
[5] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2004.
[6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2001.
[7] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005.
[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014.
[10] 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.
[11] R. Girshick, "Fast R-CNN," *Comput. Sci.*, Nov. 2015.
[12] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Tech. Rep., 1998.
[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, and C.-Y. Fu, "SSD: Single shot MultiBox detector," in *Proc. 14th Eur. Conf.*, Amsterdam, The Netherlands, Oct. 2016, pp. 21–37.
[15] Ultralytics. *YOLOv5*. Accessed: Nov. 1, 2020. [Online]. Available: https://github.com/ultralytics/YOLOv5
[16] C.-Y. Wang, A. Bochkovskiy, and H.-Y.-M. 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.
[17] 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*.
[18] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with YOLOv8," 2023, *arXiv:2305.09972*.
[19] Z. Y. Zhang, M. Y. Luo, S. X. Guo, G. Liu, S. P. Li, and Y. Zhang, "Cherry fruit recognition in natural environment based on improved YOLO v5," *Trans. Chin. Soc. Agricult. Machinery*, vol. 53, pp. 232–240, Oct. 2022.
[20] M. Yang and X. Fan, "YOLOv8-lite: A lightweight object detection model for real-time autonomous driving systems," *IECE Trans. Emerg. Topics Artif. Intell.*, vol. 1, pp. 1–16, 2024.
[21] M. Du, D. Wang, S. Liu, C. Lv, and Y. Zhu, "Rodent hole detection in a typical steppe ecosystem using UAS and deep learning," *Frontiers Plant Sci.*, vol. 13, Dec. 2022, Art. no. 992789.
[22] B. C. Cui, J. H. Zheng, Z. J. Liu, T. Ma, J. L. Shen, and X. M. Zhao, "YOLOv3 mousehole recognition for UAV remote sensing images," *Scientia Silvae Sinicae*, vol. 56, pp. 199–208, Feb. 2020.
[23] C. Li, X. Luo, and X. Pan, "CGT-YOLOv5n: A precision model for detecting mouse holes amid complex grassland terrains," *Appl. Sci.*, vol. 14, no. 1, p. 291, Dec. 2023.
[24] L. Zhu, X. Wang, Z. Ke, W. Zhang, and R. Lau, "BiFormer: Vision transformer with bi-level routing attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 10323–10333.
[25] S. Ren, D. Zhou, S. He, J. Feng, and X. Wang, "Shunted self-attention via multi-scale token aggregation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10843–10852.
[26] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
[27] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 3–19.
[28] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11531–11539.
[29] C. Hou, Q. Sun, W. Wang, and J. Zhang, "Shuffle attention multiple instances learning for breast cancer whole slide image classification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2022, pp. 466–470.
[30] S. Ma and Y. Xu, "MPDIoU: A loss for efficient and accurate bounding box regression," 2023, *arXiv:2307.07662*.

TIANSHUO XIE received the bachelor's degree in computer science and technology from Inner Mongolia Agricultural University, in 2021. She is currently pursuing the master's degree in computer science and technology. Her research interest includes computer vision.

XIAOLING LUO received the M.S. degree in electronics and communication engineering from Beijing University of Aeronautics and Astronautics, Beijing, China, in 2005, and the Ph.D. degree in agricultural information technology from Inner Mongolia Agricultural University, Inner Mongolia. She is currently a Professor with the School of Computer and Information Engineering, Inner Mongolia Agricultural University. Her main research interest includes image processing.

XIN PAN received the Ph.D. degree in information and signal processing from Beijing Jiaotong University, Beijing, China, in 2005. She was in pratacultural science with the Postdoctoral Mobile Station of Grassland Research Institute, Chinese Academy of Agricultural Sciences, from 2010 to 2014. She is currently a Ph.D. Supervisor with the School of Computer and Information Engineering, Inner Mongolia Agricultural University. Her main research interests include computer image processing and computer vision technology.