IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 January 2024, accepted 14 February 2024, date of publication 21 February 2024, date of current version 29 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3368848

RESEARCH ARTICLE

LAYN: Lightweight Multi-Scale Attention YOLOv8 Network for Small Object Detection

SONGZHE MA[1,2,3], **HUIMIN LU**[1,2,3], **JIE LIU**[1], **YUNGANG ZHU**[2], **AND PENGCHENG SANG**[1,2,3]

[1]School of Computer Science and Engineering, Changchun University of Technology, Changchun 130102, China
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun 130012, China
[3]Smart Health Joint Innovation Laboratory for the New Generation of AI, Changchun 130102, China

Corresponding author: Huimin Lu (luhuimin@ccut.edu.cn)

This work was supported in part by the Fundamental Research Funds for the Central Universities, JLU, under Grant 93K172022K15; in part by the Industrial Technology Research and Development Special Project of Jilin Provincial Development and Reform Commission, in 2023, under Grant 2023C042-6; and in part by the Key Project of Science and Technology Research Plan of Jilin Provincial Department of Education, in 2023, under Grant JJKH20230763KJ.

ABSTRACT Currently, with the widespread application of embedded technology and the continuous improvement of computational power in mobile terminals, the efficient deployment of algorithms on embedded devices, while maintaining high accuracy and minimizing model size, has become a research hotspot. This paper addresses the challenges of deploying the YOLOv8 algorithm on embedded devices and proposes a novel lightweight object detection algorithm focusing on small object detection. We optimize the model through two key strategies, aiming to achieve lightweight deployment and improve the accuracy of small object detection. Firstly, GhostNet is introduced as the backbone network for YOLOv8 in order to achieve lightweight deployment. By using some cost effective operations to generate redundant feature maps, we not only reduce the number of model parameters while ensuring better detection results, but also improve the speed of the model. Secondly, a new multi-scale attention module is designed to enhance the network's acquisition of crucial information for small targets, which includes a multi-scale fusion attention mechanism and the Soft-NMS algorithm. The multi-scale fusion attention mechanism captures key features of discriminative small targets in the feature map tensor from both spatial and channel dimensions, suppressing non-key information, reducing the impact of complex and unimportant information in the image, enhancing the network model's learning ability for important features of small targets. The Soft-NMS method improves accuracy by significantly reduces false positives in the detection results. To validate the performance of our proposed method, we conducted validation experiments on the PASCAL VOC dataset and evaluated the model's generalization ability on the MS COCO dataset. The experiments results demonstrate that our model achieves a significant improvement in small object detection, with a 5.41% increase in detection accuracy compared to the existing YOLOv8. Meanwhile, FLOPs are reduced by 49.62%, and the number of model parameters is reduced by 48.66%. These results fully confirm the effectiveness of our innovative method in achieving both lightweight deployment and significant efficacy in small object detection tasks.

INDEX TERMS Object detection, lightweight, attention mechanism, YOLO.

I. INTRODUCTION

In recent years, with the rapid development of computer vision based on deep learning, object detection has gradually become a popular research direction in computer vision, widely applied in various fields such as video surveillance, industrial inspection, and healthcare [1]. The implemention of computer vision to reduce the consumption of manpower and resources has significant practical significance [2]. However, most mainstream object detection frameworks currently lack specific improvements for small targets [3]. In real-world scenarios, small object detection is crucial, such

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

as identifying disaster victims in drone search and rescue, recognizing distant traffic signs and vehicles in autonomous driving [4]. Both the training and testing stages of small object detection are conducted on images with a resolution ranging from 51×72≤size≤4064×6354. According to the image resolution range used in this paper, targets with resolutions of 32×32 and below are generally considered small targets. However, existing object detection algorithms show a certain degree of performance degradation when targets have small absolute and relative sizes [5]. This phenomenon can be attributed to the following reasons [6]:

1. If the scale of the detected target is small, as the network deepens during training, the detected target is prone to losing features such as edge information and grayscale information [7]. Advanced semantic information is also obtained less, and there may be some noise information in the image that misleads the training network to learn incorrect features.

2. The size of the receptive field mapped to the original image also plays a crucial role in the success of target detection [8]. When the receptive field is relatively small, spatial structural features are preserved more, but abstract semantic information may be less. Conversely, when the receptive field is large, relatively richer semantic information is preserved, but there may be a potential loss of spatial structural information for the target.

3. Convolutional Neural Networks (CNNs) implement the extraction of features discretely, making it challenging to attain sub-pixel accuracy [9]. When dealing with small targets, if the neural network lacks one pixel in the deep layers, it may lack 8 pixels or 16 pixels or even more in the shallow layers. This has a minimal impact on large targets but significantly affects small targets. Therefore, successfully detecting small targets and reducing the model's size without compromising accuracy is an urgent problem that needs to be addressed.

To address these issues, many scholars have proposed their solutions. Zhao et al [10]. proposed the BiTNet method, which extracts image features by introducing the Efficient Transformer Block (ETB) and Efficient Convolution Aggregation Block (ECAB). The method reduces computational costs by designing a homogeneous multi-branch. Cui et al [11]. introduced the LC-YOLO method, aiming to integrate prior information into CNN and maximize knowledge sharing within the network to enhance shallow features. Liu et al [12]. presented the YOLO-UAVlite method, which combines the Spatial-Coordinate Self-Attention (SCSA) module with the lightweight backbone SCSAshufflenet to reduce information loss during feature map fusion and decrease model weight. Zhao et al [13]. proposed the MCANet method, specifically utilizing the Hierarchical Cross-Fusion Lightweight Transformer Based on Multi-ConvHead Attention for object detection. Peng et al [14]. replaced the backbone of YOLOv4 with MobileNetV2-CA and added the Squeeze-and-Excitation (SE) module. Gong et al [15]. introduced the reparameterized fusion convolution (RFConv) to leverage the advantages of both methods with the minimum computational cost when detecting edges and small targets.

Despite the achievements made in the above-mentioned studies, challenges persist due to complex backgrounds, considerable spatial resolution variations, the prevalence of small and irregularly arranged objects. Additionally, issues such as excessively large model sizes and slow inference speeds further compound these challenges. Achieving a dual enhancement of detection speed and accuracy, improving algorithm robustness to changes in target scales, and achieving lightweight small object detection remain highly challenging tasks in object detection. Existing algorithms face challenges such as large-scale models and high computational requirements, which hinder the deployment on embedded devices. This paper proposes a lightweight small object detection algorithm with a multi-scale attention mechanism based on YOLOv8. Firstly, we utilize GhostNet as the backbone to "lighten" the algorithm, reducing the model size while ensuring accuracy for lightweight deployment. Secondly, we design a novel multi-scale attention module, including a multi-scale attention mechanism and the Soft-NMS method [16]. This module refines the feature information in the target region and generates effective target features. It allocates more weight to channels in high-level feature maps with richer semantic information, thereby improving the discriminative ability of the algorithm for small targets. Additionally, the Soft-NMS method effectively reduces false positives in detection results, which further enhances detection accuracy. Experimental results demonstrate that our proposed method not only has excellent performance in small object detection but also achieves real-time capabilities while maintaining lightweight deployment. We successfully overcome challenges associated with efficient deployment on embedded devices, achieving a dual enhancement of detection speed and accuracy for lightweight small object detection tasks.

The main contributions of our paper are as follows:

- We propose the Lightweight Multi-Scale Attention YOLOv8 Network (LAYN), which is a novel lightweight small object detection network. It can reduce the model's size without compromising accuracy.
- By utilizing the GhostNet module, we employ some cheap operations to generate redundant feature maps. This approach ensures a good detection performance while reducing the number of model parameters, enhancing the model's speed.
- A new multi-scale mixed attention mechanism has been designed to capture important features of discriminative small targets in feature map tensors from both spatial and channel dimensions. This reduces the impact of complex and unimportant information in the image, enhancing the network model's ability to learn important features of small targets.
- Utilizing Soft-NMS can alleviate the issue of missed detections for closely spaced small targets, effectively

detecting large areas of overlapping targets, thereby improving the detection accuracy of small targets.

The remaining work in this paper is organized as follows: In Section II, we provide a brief introduction to related work. In Section III, we present a detailed explanation of the Lightweight Multi-Scale Attention YOLOv8 network. In Section IV, we conduct experiments and analyze the results. In Section V, we discuss and conclude the paper in the final section.

II. RELATED WORK

At present, the methods for small target detection can be categorized into the following three directions [17].

First, leveraging the concept of an image pyramid [18], the detected input image undergoes scale transformation–scaling up or down. This process forms an image pyramid with progressively increasing or decreasing image scales from top to bottom. Subsequently, the target of interest is detected by sliding a fixed-size window over each layer of the image. For instance, multi-task cascaded convolutional networks (MTCNN) adopts this approach to detect faces at different scales [19]. However, Scale Normalization for Image Pyramids (SNIP), proposed by Singh and Davis [20], also utilizes the image pyramid concept but requires passing images of varying resolutions through CNNs, which results in relatively high computational costs and slower detection speeds.

Second, incorporating the Attention Mechanism (AM) [21]. This mechanism empowers the network model to achieve superior performance by emphasizing essential information while discarding irrelevant data [22]. Initially utilized for machine translation [23], the AM has been experimentally demonstrated by Vaswani et al. to enhance the performance of neural network models in image classification and natural language processing tasks [24]. Xu introduces an attention-based YOLO (You Only Look Once) detection algorithm that addresses the shortcomings of the original YOLO model, such as biased boundary localization and the challenge of distinguishing overlapping objects. This is achieved by integrating channel-domain and spatial-domain attention mechanisms into the YOLO model's feature extraction network. Fu et al. has designed a Recurrent Attentional Convolutional Neural Network (RA-CNN) for fine-grained image recognition [25]. It employs an Attention Proposal Network (APN) to accurately locate regions in an image that contain detailed object information [26]. By focusing on these areas, the model learns rich detail features and enhance fine-grained image recognition accuracy.

Third, adopting a data expansion approach [27], [28]. For example, Kisantal et al [29]. contend that the low accuracy in detecting small targets primarily results from the scarcity of images featuring small objects. Even when such images exist, small objects are rarely present. The authors of this article propose the repeated sampling of images containing small targets and duplicating small targets found within an image to other areas of the same image. This strategy acts as data augmentation, augmenting the weights of corresponding scale objects during training by increasing the number of matching anchor frames.

III. LIGHTWEIGHT MULTI-SCALE ATTENTION YOLOV8 NETWORK FOR SMALL OBJECT DETECTION

A. NETWORK STRUCTURE

The YOLOv8-based target detection model exhibits exceptional accuracy [30], [31], [32], [33], [34], [35], [36]; however, it demands a high-performance GPU for real-time execution. This reliance on substantial computational resources limits its efficiency when used in embedded devices, which are characterized by restricted memory and processing capabilities. Moreover, YOLOv8 proves to be less adept at detecting small targets, which invariably pose a challenge due to their diminutive size and low-resolution nature. The YOLOv8 algorithm adopts Cross Stage Partial Darknet (CSPdarknet) as the foundational framework for feature extraction. Nevertheless, as the network deepens, the receptive field expands, causing a reduction in feature map dimensions. Consequently, feature abstraction intensifies, and semantic features become increasingly prominent. However, this augmentation in semantic abstraction leads to a loss of precise location information, significantly impeding the accurate detection of small targets.

Based on the aforementioned insights, we hereby introduce a novel lightweight architecture derived from YOLOv8 and incorporate a multi-scale attention module to address the challenge of small target detection. The network structure is illustrated in Figure 1.

Figure 1 illustrates the functions of key components within YOLOv8 architecture. The Cross Stage Partial Module (CSPModule) plays a pivotal role in optimizing the network's efficiency by halving the channel count, reducing the number of convolutions and enhancing network speed. On the other hand, the Spatial Pyramid Pooling-Fast (SPPF) Bottleneck facilitates the fusion of feature maps at varying scales by employing downsampling techniques [37]. This fusion process ensures the seamless integration of feature information. Our architecture leverages the Path Aggregation Network (PANet) structure, combining the Feature Pyramid Network (FPN) [38] with the Pixel Aggregation Network (PAN) [39]. This involves introducing a bottom-up feature pyramid alongside the FPN layer and incorporating the Lightweight Multi-Scale Attention (LMA) module during the upsampling phase. Feature maps from each layer are fused through the Concat operation, culminating in the effective transfer of feature information.

B. THE GHOSTNET BACKBONE

In YOLOv8 models, it is common practice to incorporate rich, and at times, redundant feature maps to ensure a comprehensive understanding of the input data. Inexplicably, the issue of feature map redundancy has been somewhat ignored or overlooked in model structure design. To overcome this

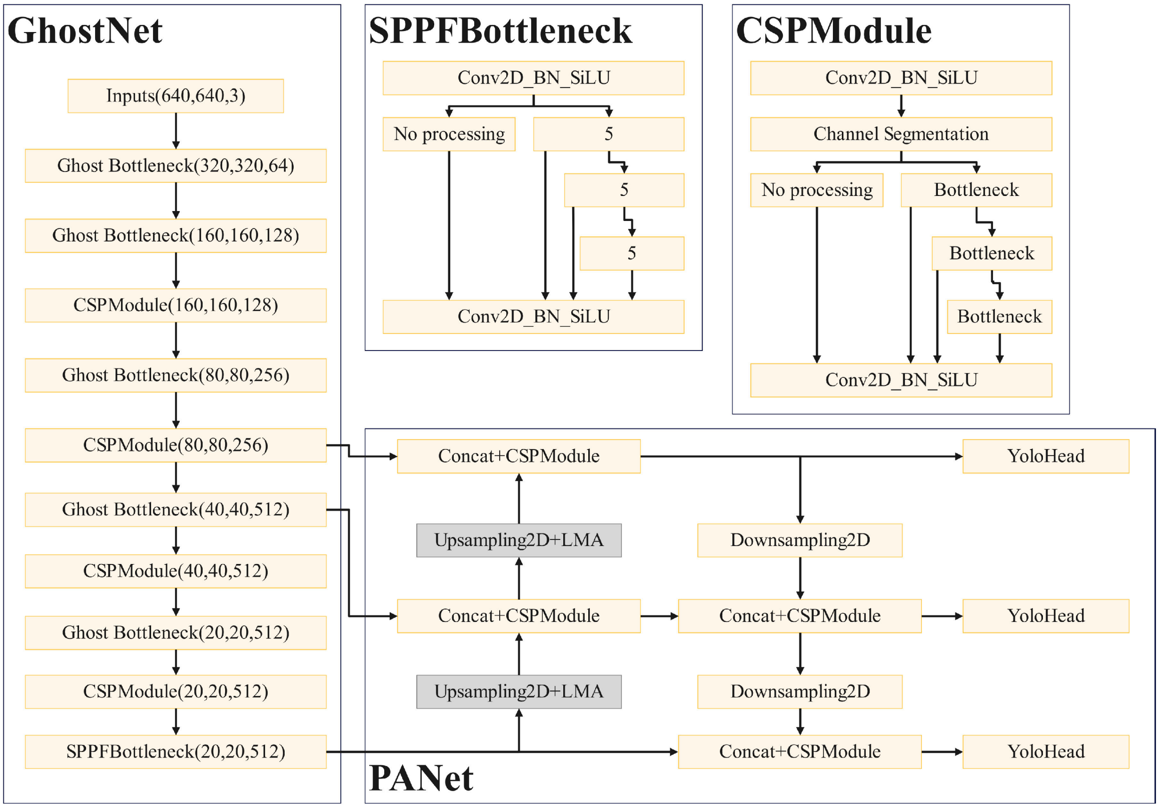

FIGURE 1. LAYN network structure.

shortcoming, we employ cost-effective operations, known as Cheap Operations, to generate these redundant feature maps. This strategic choice reduces the number of model's parameters, consequently enhancing execution speed while ensuring robust detection results.

The core idea of GhostNet [40] is to use some operations of Cheap Operations to generate these redundant feature maps. First, given the input data $X \in \mathbb{R}^{c\times h\times w}$, where c is the number of input channels, and h and w are the height and width of the input data, respectively, then the operation of generating n feature maps for any convolutional layer can be expressed as:

$$Y = X * f + b \tag{1}$$

where $*$ denotes the convolution operation, b is the bias term, $Y \in \mathbb{R}^{h'\times w'\times n}$ is the output feature map of n channels, and $f \in \mathbb{R}^{c\times k\times k\times n}$ is the convolution kernel of this feature layer. In addition h' and w' are the height and width of the output data, respectively, and $k \times k$ is the kernel size of the convolution kernel f. In this convolution process, the number of FLOPs required amounts to $n \times h' \times w' \times c \times k \times k$ because the number of convolution kernels n and the number of channels c are very large. Therefore, there is a module called Ghost Module in GhostNet, which functions as an alternative to normal convolution. It is shown in Figure 2. The Ghost Module divides the normal convolution into two parts, first, a normal 1 × 1 convolution, which is a small amount of convolution. This 1 × 1 convolution works like feature ensembling, generating feature concentration of the input feature layer. It then performs a depth-separable convolution, which is a layer-by-layer convolution or a Cheap Operations, that uses the obtained feature condensation to generate the Ghost feature map.

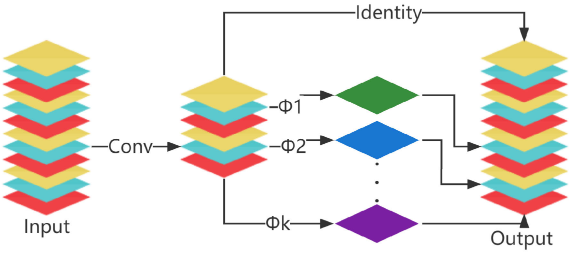

FIGURE 2. The ghost module.

Secondly, we employ the Ghost Module as a replacement for the standard convolution within the bottleneck structure, thereby giving rise to the aptly-named Ghost Bottlenecks. Ghost Bottlenecks comprise two distinct components: the primary section and the residual edge segment. The component that incorporates the Ghost Module is denoted as the primary section. Figure3 illustrates two variations of Ghost Bottlenecks. When there is a need to compress the width and height of the feature layer, we set the stride (of these Ghost Bottlenecks) to 2. In such cases, additional convolutional

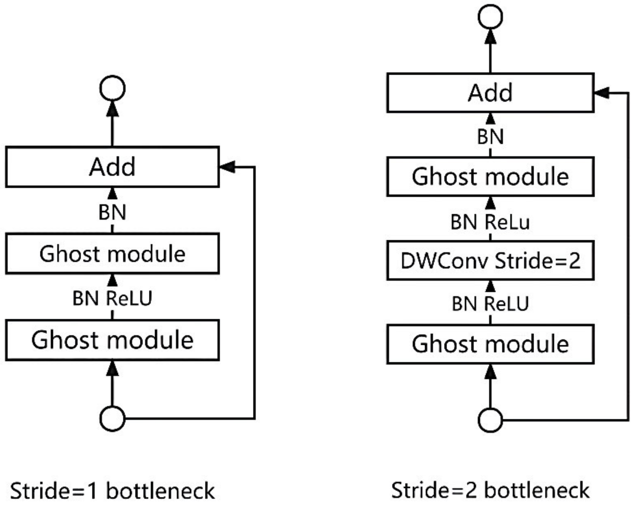

FIGURE 3. Ghost bottleneck.

layers are introduced within the Bottlenecks. In the primary section, we integrate a 2×2 depth-separable convolution with a stride of 2 × 2 into the two Ghost Modules to compress the feature layer's dimensions. In the residual edge segment, we also introduce a 2 × 2 depth-separable convolution and a 1 × 1 standard convolution.

The GhostNet architecture primarily consists of Ghost Bottlenecks. When an image is input into GhostNet, we initially apply a 16-channel standard 1 × 1 convolution block (Conv+BN+activation function). Subsequently, we commence the stacking of Ghost Bottlenecks, which ultimately generates a feature layer of 7 × 7 × 160 (for input images of size 224 × 224 × 3).To adjust the number of channels, we employ a 1 × 1 convolution block, resulting in a 7 × 7 × 960 feature layer. Following this, global average pooling is executed, followed by channel adjustment through a 1 × 1 convolution block, resulting in a 1 × 1 × 1280 feature layer. Lastly, tiling is employed for full concatenation, facilitating the classification process.

To explain further, the reasons why GhostNet can reduce the algorithm's parameter count primarily mainly encompass the following aspects:

1) THE GHOST MODULE

GhostNet introduces the Ghost Module, which is a novel module design that achieves parameter sharing by splitting a convolutional layer into two parts. These two parts are called the ''main branch'' and the ''ghost branch.'' The weights of the main branch are used to compute the output, while the ghost branch weights are a subset of the main branch and are used to generate the main branch weights. This design significantly reduces the number of parameters since the ghost branch has far fewer parameters than the main branch.

2) DEPTHWISE SEPARABLE CONVOLUTION

GhostNet employs Depthwise Separable Convolution, which is a lightweight convolutional operation. Depthwise Separable Convolution breaks down the standard convolution into depth-wise convolution and point-wise convolution, reducing the number of parameters. GhostNet uses Depthwise Separable Convolution in the Ghost Module to reduce the model's complexity.

3) SHUFFLENET-INSPIRED APPROACH

GhostNet borrows the idea from ShuffleNet and uses channel shuffling to further reduce computational complexity. Channel shuffling divides input channels into multiple groups, performs convolution operations within each group, and then combines the results. This helps to reduce the computational cost.

4) WIDTH MULTIPLIER

GhostNet allows the use of a width multiplier to control the network's width, making it easy to reduce the number of parameters. By reducing the number of channels in each layer, the model's parameter count can be significantly reduced.

To restate, GhostNet effectively trims the model's parameter count through techniques like the Ghost Module, Depthwise Separable Convolution, channel shuffling, and the incorporation of a width multiplier to govern network width. This amalgamation of strategies yields a lightweight neural network architecture optimally-suited for mobile devices and embedded systems. It is important to highlight that it attains a formidable classification performance level, establishing itself as a highly parameter-efficient solution.

C. MULTI-SCALE ATTENTION MODULE

1) SPATIAL ATTENTION

The spatial attention mechanism enables a convolutional neural network to effectively discern and learn the areas requiring attention. This process maps the spatial information from the original image to an alternative space, thereby preserving crucial image features. Refer to Figure4 for an illustration of its structure.

The MaxPooling operation assigns higher weights to local features like prominent edge contours in an adaptive manner, whereas the AvgPooling operation gives priority to global features within salient regions. By combining MaxPooling and AvgPooling, the network can adaptively capture both discriminative global and local features.

2) CHANNEL ATTENTION

The channel attention mechanism [48]assesses the significance of each channel's feature map by assigning a weight to the feature maps of the n channels. A higher weight signifies that the channel's feature map contains more critical features, which warrants heightened attention. For a visual representation, please refer to Figure5.

Utilizing two fully connected layers to construct a bottleneck architecture within Channel Attention (CA) offers two significant advantages. First, it enhances CA's capacity for

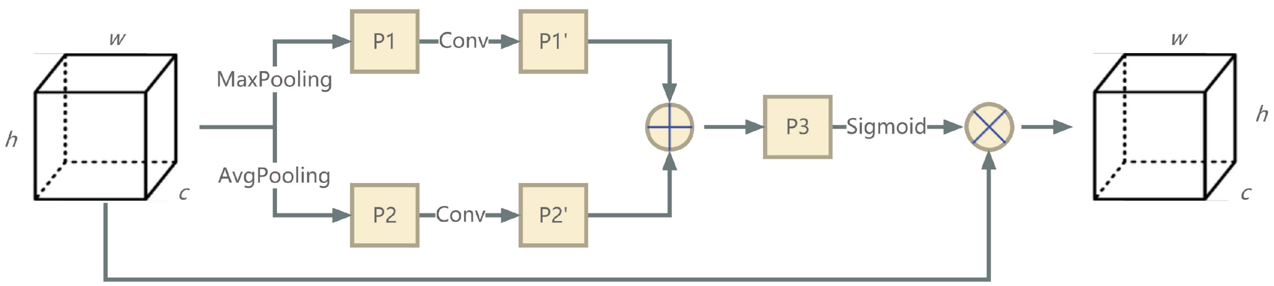

FIGURE 4. Spatial attention module.

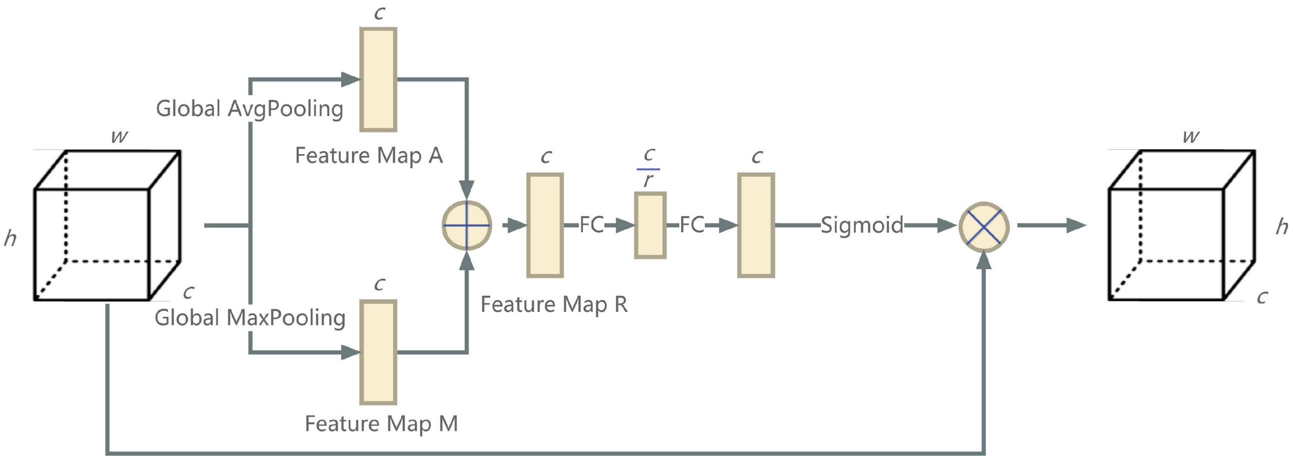

FIGURE 5. Channel attention module.

robust nonlinear learning. Second, it substantially diminishes the parameter count, yielding a more efficient model.

3) MULTI-SCALE ATTENTION

Assuming that the input to the multi-scale attention (MA) mechanism is the feature map tensor $T \in \mathbb{R}^{c\times h\times w}$, the feature map tensor T is first input to the spatial attention (SA) module for computation to add spatial attention to obtain the feature map tensor $S \in \mathbb{R}^{w\times h\times c}$; Then $\frac{c}{4}$ 1×1 convolution kernels are used to convolve the feature map tensor S to obtain the feature map tensor $D \in \mathbb{R}^{w\times h\times \frac{c}{4}}$, which realizes the dimensionality reduction of the feature map tensor Sand reduces the computation of the subsequent operations; Then the multi-scale feature extraction is performed on the feature map tensor D using four branches to obtain the multi-scale feature map tensor $P_1 \in \mathbb{R}^{w\times h\times \frac{c}{4}}$, $P_2 \in \mathbb{R}^{w\times h\times \frac{c}{4}}$, $P_3 \in \mathbb{R}^{w\times h\times \frac{c}{4}}$ and $P_4 \in \mathbb{R}^{w\times h\times \frac{c}{4}}$, and the Concat operation is used to perform feature fusion on the feature map tensor P_1, P_2, P_3 and P_4 for feature fusion to obtain the feature map tensor $Q \in \mathbb{R}^{w\times h\times c}$; then the feature map tensor Q is input to the CA module for calculation to add channel attention to obtain the feature map tensor $C \in \mathbb{R}^{w\times h\times c}$; Finally, the Add operation is used to fuse the features of the feature map tensor S and C to obtain the feature map tensor $H \in \mathbb{R}^{(w\times h\times c)}$ as the output of the multiscale attention mechanism. The structure is shown in Figure6.

The multiscale attention mechanism employs four branches to conduct multiscale feature extraction from the input feature maps. It utilizes two sequential convolution operations with kernels of dimensions 1 × 3 and 3 × 1, as well as two sequential convolution operations with kernels of dimensions 1 × 5 and 5 × 1. This incorporation of cascaded asymmetric convolution operations effectively reduces the network's parameter count while also introducing additional nonlinear activation layers, thereby enhancing its nonlinear learning capability. Furthermore, it facilitates the capture of discriminative features associated with small targets within the feature map tensor, considering both spatial and channel dimensions. This diminishes potential distortion resulting from the impact of extraneous, non-essential information within the image, ultimately fortifying the network model's capacity in learning from small target features.

4) SOFT-NMS

During target detection, multiple bounding boxes with high confidence typically surround the actual target. To address this, the NMS (Non-Maximum Suppression) method is commonly employed to eliminate redundant bounding boxes,

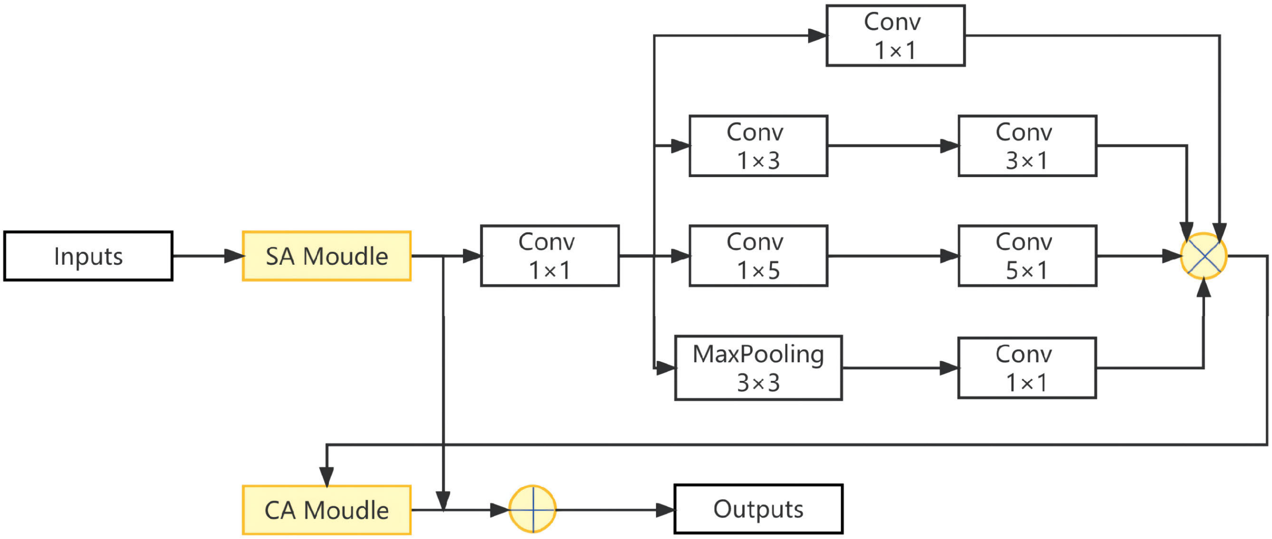

FIGURE 6. **Multi-Scale attention module.**

ensuring that only one bounding box per object is retained. The NMS algorithm proceeds as following steps: 1. Categorize all the boxes to remove background classes. 2. For each target class of bounding boxes, sort them in descending order of classification confidence. 3. Within a given class, select the bounding box with the highest confidence and retain it. 4. Compute the Intersection-over-Union (IOU) between the bounding box with the highest confidence and the remaining boxes individually. Remove any remaining boxes with an IOU value exceeding the threshold. 5. Iterate steps 3 and 4 until the processing for a target class is completed. 6. Repeat steps 2 to 5 until NMS processing for all target classes is finished. 7. Output the final selected bounding boxes.

The mathematical expression governing the suppression of confidence for other bounding boxes in favor of the current bounding box with the highest confidence level in the NMS algorithm is expressed as follows:

$$S_i = \begin{cases} S_i, & iou(M, b_i) < N_t \\ 0, & iou(M, b_i) \geq N_t \end{cases} \tag{2}$$

where S_i is the confidence level of detection frame b_i, N_t is the preset IOU threshold, IOU is the Intersection over union, and M is the bounding box with the highest confidence level.

In dense object placement scenarios, substantial overlap between distinct objects, and the resulting bounding box Intersection over Union (IOU) surpassing the threshold may result in the removal of the bounding box with lower confidence. This outcome carries the risk of failure in detecting the relevant object, which adversely affecting comprehensive detection performance. Although raising the IOU threshold can mitigate the issue of missed detections, it concurrently elevates the likelihood of redundant detections. Despite that, adjusting the IOU threshold in isolation is insufficient to achieve a harmonious balance between recall and accuracy. Although increasing the IOU threshold can mitigate the problem of missed detections, it also increases the probability of redundant detections, rendering it inadequate for achieving a balanced trade-off between recall and accuracy.

To deal with this phenomenon, Soft-NMS offers a solution to these challenges by improving upon NMS without introducing additional complexity. At its core, this method incorporates a decay function designed to diminish the confidence levels of neighboring bounding boxes, while still permitting their retention. Soft-NMS encompasses two distinct forms of attenuation, the first of which is delineated by the following mathematical expression:

$$S_i = \begin{cases} S_i, & iou(M, b_i) < N_t \\ S_i(1 - iou(M, b_i)), & iou(M, b_i) \geq N_t \end{cases} \tag{3}$$

where N_t is the threshold value, if the IOU of bounding box b_i and M is less than this threshold value N_t, then its confidence level remains constant, and when it is greater than N_t, its confidence level S_i decays linearly according to the degree of overlap. When the IOU is just greater than N_t, the confidence S_i changes abruptly, which leads to the instability of the resultant sequence. The second form of attenuation is a continuous decay function. It is represented in the following mathematical expression:

$$S_i = S_i e^{-\frac{iou(M, b_i)^2}{\sigma}} \tag{4}$$

The σ parameter is used to regulate the degree of decay. Soft-NMS multiplies the confidence degree S_i by a Gaussian weighting function related to IOU, and the decay is gentler when the IOU value is low, and the degree of decay is greater when the IOU is closer to 1. This continuous decay function avoids the problem of sudden changes in the confidence degree existing in Equation (3). It also exhibits a favorable level of stability.

Soft-NMS diminishes the confidence levels of bounding boxes that intersect with the currently favored detection frame, rather than their outright removal. This approach partially mitigates the issue of missed detections in dense scenarios and proves effective in discerning objects with substantial overlap, ultimately enhancing the accuracy of detecting small targets.

IV. EXPERIMENTAL

A. EXPERIMENTAL SETUP

1) DATASETS

We evaluated our proposed method using two datasets: the PASCAL VOC dataset [41], consisting of PASCAL VOC2007 and PASCAL VOC2012, which includes 20 target classes, 16551 training images, and 4852 test images, and a custom vehicle dataset derived from the MS COCO dataset [42]. The constructed vehicle dataset encompasses three primary categories: cars, buses, and trucks, comprising a total of 16977 images. To train the vehicle detection model, we partitioned the constructed dataset into three subsets :training, validation, and test, with a ratio of 8:1:1.

2) MODEL TRAINING AND EVALUATION

In this paper, the learning rate is dynamically adjusted using the cosine annealing algorithm, and the network model's weights are updated and optimized through the Adam algorithm during model training. Data augmentation methods such as Mosaic and Mixup are employed to enhance the dataset. The specific parameter settings are as follows: a batch size of 16, a learning rate of 0.01, a weight decay factor of 0.0005, and a training duration of 200 epochs for the entire dataset.

To assess the efficacy of our proposed method, we employ a comprehensive set of metrics such as precision, recall, mAP (Mean Average Precision), parameter count, and model size. Precision is determined by the ratio of correctly predicted positive samples to the total samples predicted as positive. It is mathematically defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

Recall is the calculation of the percentage of all correctly predicted targets. It is defined as follows:

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

In this context, TP represents the count of correctly predicted positive samples, FP represents the count of samples predicted to be positive but are actually negative, and FN represents the count of samples predicted to be negative but are actually positive.

The formula for calculating mAP is as follows:

$$AP = \int_0^1 P(R)dR \tag{7}$$

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{8}$$

TABLE 1. Performance results of different models on VOC07+12.

No.	GhostNet	MA	Soft-NMS	mAP(%)
1				86.54
2	✓			84.14
3	✓	✓		89.69
4	✓		✓	87.43
5	✓	✓	✓	91.95

3) IMPLEMENTATION DETAILS

The hardware used for experimentation consists of an Intel(R) Core(TM) i9-10980HK CPU, a NVIDIA GeForce RTX 3080 graphics card, and a Windows operating system. The software environment incorporates CUDA 11.1 and cuDNN 8.0.4, with experiments conducted within the PyTorch 1.9 deep learning framework.

B. RESULTS

1) ABLATION EXPERIMENT RESULTS

Ablation experiments were conducted to examine how alterations in network structure affect network performance. To comprehensively evaluate performance variations, this study employed two distinct datasets and conducted experiments on five variants: YOLOv8, YOLOv8-GhostNet, YOLOv8-GhostNet-Soft-NMS, YOLOv8-GhostNe-MA, and LAYN. The outcomes of these ablation experiments can be found in Tables 1 and 2.

Table 1 illustrates the validation results using the PASCAL VOC dataset. Replacing the YOLOv8 backbone with GhostNet led to a 2.40% reduction in model accuracy. Conversely, combining the backbone with the Soft-NMS algorithm yielded an accuracy improvement of 0.89%. The inclusion of the MA module alongside GhostNet resulted in a noteworthy 3.15% enhancement in model accuracy. Notably, the LAYN (Lightweight Multi-Scale Attention YOLOv8 Network) method proposed in this paper, integrating GhostNet and LMA modules into the YOLOv8 network, achieved a substantial 5.41% increase in model accuracy.

Table2 displays the model validation results using the vehicle dataset derived from MS COCO. Replacing YOLOv8's backbone with GhostNet led to a reduction in model accuracy by 1.32%. In contrast, substituting the backbone with the Soft-NMS algorithm improved model accuracy by 0.25%. Introducing the MA module alongside GhostNet resulted in a notable 1.29% boost in model accuracy. Also notably, our proposed LAYN method, which integrates GhostNet and LMA modules into the YOLOv8 network, achieved a substantial 2.68% enhancement in model accuracy on the vehicle detection dataset.

Table 3 presents the results achieved by our proposed method on the PASCAL VOC test set. These are contrasted with the single-category outcomes of the benchmark model. These results unequivocally demonstrate that our method yields higher single-class Average Precision (AP) values across all 20 categories, signifying improved detection

TABLE 2. Performance results of cars on MS COCO.

No.	GhostNet	MA	Soft-NMS	mAP(%)
1				95.25
2	✓			93.93
3	✓	✓		96.54
4	✓		✓	95.50
5	✓	✓	✓	97.93

TABLE 3. Test set single-test AP results.

Category	YOLOv8	LAYN
airplane	95.42%	97.01%
bicycle	91.96%	96.14%
bird	92.69%	96.05%
boat	76.23%	85.64%
bottle	73.03%	86.44%
bus	95.90%	97.32%
car	85.53%	95.60%
cat	93.52%	96.48%
chair	73.45%	83.72%
cow	93.32%	95.79%
diningtable	69.13%	86.50%
dog	89.97%	93.10%
horse	95.57%	95.04%
motorbike	92.97%	96.07%
person	91.03%	95.87%
pottedplant	60.02%	73.93%
sheep	94.33%	94.90%
sofa	83.25%	87.52%
train	95.59%	97.47%
tvmonitor	87.82%	88.37%

accuracy compared to YOLOv8. Moreover, when compared to the YOLOv8 approach, our method displays superior capabilities in multi-category target detection, thereby affirming the applicability and suitability of our proposed approach.

Table 4 demonstrates that by substituting YOLOv8's backbone network with GhostNet, FLOPs and model parameters are reduced by 66.70% and 58.57%, respectively, in contrast to YOLOv8. On the basis of replacing the backbone network, after adding the Soft-NMS algorithm, FLOPs and model parameters are reduced by 65.43% and 54.90% respectively compared to YOLOv8. When GhostNet serves as the backbone, accompanied by the MA module, it results in a reduction of 65.43% in FLOPs and 51.10% in model parameters in comparison to YOLOv8. Notably, the LAYN method, introduced in this paper, by involving the amalgamation of GhostNet and LMA modules with the YOLOv8 network, achieves a substantial decrease of 49.62% in FLOPs and 48.66% in model parameters when compared to YOLOv8. This underscores the capability of our proposed method to achieve efficient lightweight deployment.

In Figure 7, we offer a performance comparison between the methodology introduced in this paper and several alternative algorithms across various scenarios. From left to right, we present annotated images, the detection results produced by YOLOv8-GhostNet, the results of YOLOv8-Soft-NMS, and the results generated by our method. Our approach demonstrates accurate object class detection, particularly in scenarios featuring small, blurry targets or densely clustered objects with occlusions in both the foreground and background, where other algorithms falter. These results underscore the superior performance of our approach compared to YOLOv8.

TABLE 4. Parameters amount and FLOPs results of different models.

No.	GhostNet	MA	Soft-NMS	FLOPs(G)	Params
1				23.78	25.292M
2	✓			7.85	10.478M
3	✓	✓		10.23	12.369M
4	✓		✓	8.22	11.406M
5	✓	✓	✓	11.98	12.986M

2) COMPARISON OF EXPERIMENTAL RESULTS

Performance Comparison Using Different Attention Mechanisms

In order to validate the performance of the recently introduced MA module, this research integrated various attention modules, namely the Effective Channel Attention (ECA) module, the Convolutional Block Attention Module (CBAM) module, and the Coordinate Attention (CA) module. Table 5 presents a comparative analysis of network performance across these different attention modules. Our approach demonstrates superior performance in mAP, parameter count, and FLOPs (G) compared to the alternative attention modules on both the Pascal VOC and COCO datasets.

Performance comparison with existing YOLO series lightweight object detection algorithms

To fully evaluate the effectiveness of our approach, we performed a comparative analysis with the well-established YOLO family of lightweight object detection algorithms. The results are shown in Table 6.

According to Table 6, although the FLOPs of YOLOv5n, and the parameters and FLOPs of YOLOX-nano on the PascalVOC dataset are smaller than those of our algorithm, they are significantly lower in terms of accuracy when compared to our proposed algorithm. Our algorithm outperforms in all indexes when compared with YOLOv3-tiny, YOLOv4-tiny, YOLOX-tiny, and YOLOv7-tiny. In comparison to YOLOv4-tiny, which has the lowest mAP index, our algorithm achieves a remarkable 39.45% improvement in accuracy.

Performance Comparison with Existing Mainstream Lightweight Object Detection Algorithms

We compared LAYN with other algorithms on the PASCAL VOC dataset, and the results are shown in Table 7. Table 7 demonstrates that our proposed LAYN surpasses the current SOTA algorithm by Gong et al. on all metrics, with an increase of 1.69% in mAP, and reductions of 26.54% and 72.13% in Params and FLOPs, respectively. For BiTNet and MCANet, although their Params and FLOPs are slightly smaller than LAYN, LAYN significantly outperforms BiTNet

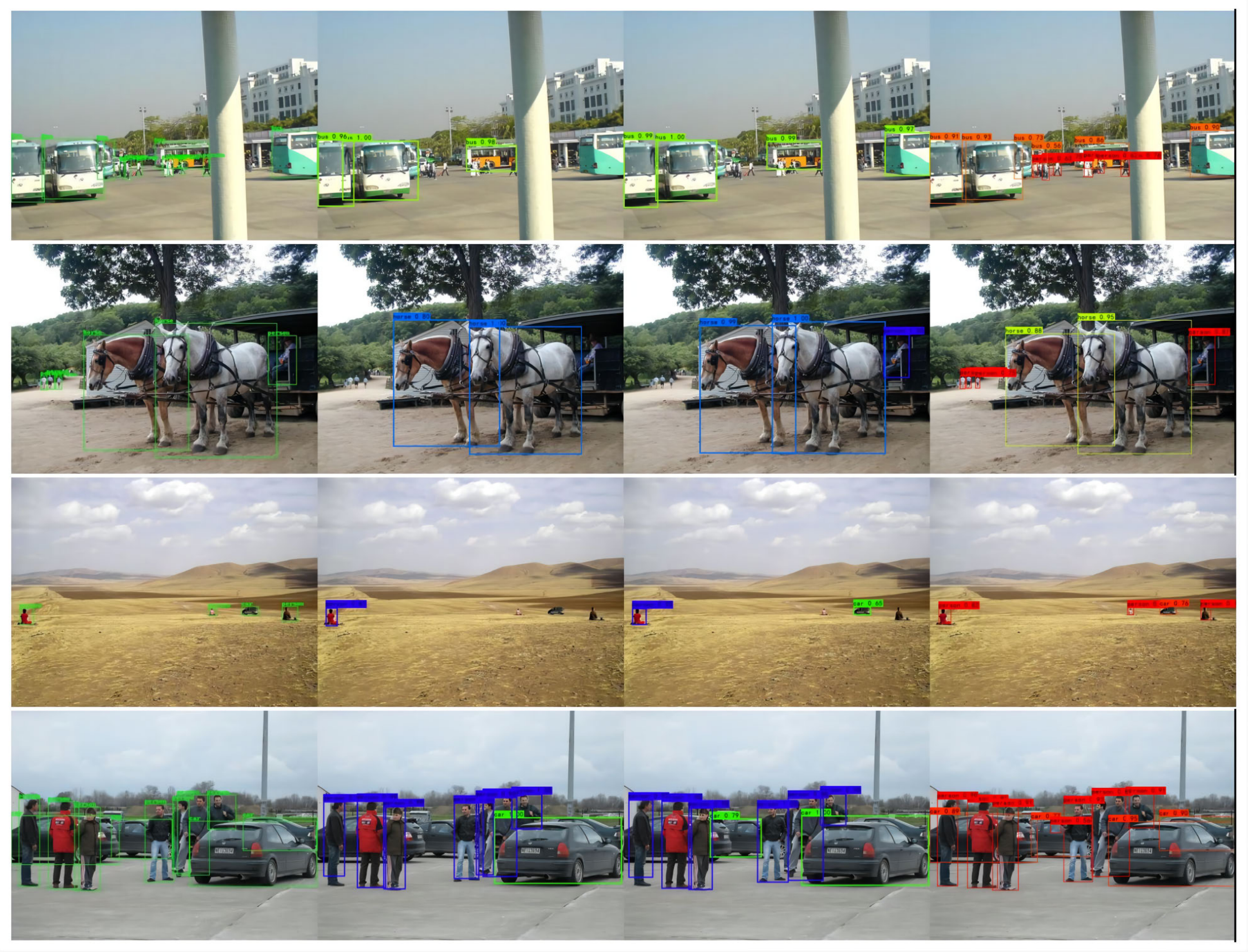

FIGURE 7. Multi-Scale attention module.

TABLE 5. Results of the comparison of different attention mechanisms.

Datasets	Model	mAP(%)	Params	FLOPs(G)
Pascal VOC	ECA	88.91	13.241M	12.18
	CBAM	88.30	13.376M	12.14
	CA	89.01	13.143M	12.11
	MA(Ours)	91.95	12.986M	11.98
COCO	ECA	95.91	13.157M	11.96
	CBAM	96.36	13.317M	11.96
	CA	96.90	13.064M	11.94
	MA(Ours)	97.93	12.916M	11.76

and MCANet in terms of accuracy, with mAP improvements of 18.55% and 21.33%, respectively. This substantial increase in accuracy comes with a slight increase in model size. Compared with the other two methods, LAYN achieves superiority in mAP, Params, and FLOPs, demonstrating that LAYN successfully extracts features of small objects while reducing the model size, significantly improving the accuracy of small object recognition.

V. DISCUSSION

Tables 1, 2, and 4 present the influence of each suggested enhancement method on the algorithm's performance across diverse datasets. Independent assessments were conducted on YOLOv8, YOLOv8-GhostNet, YOLOv8-GhostNet-SoftNMS, YOLOv8-GhostNet-MA, and LAYN. Mean Average Precision (mAP) and other performance metrics reveal that enhancing the backbone network individually

TABLE 6. Comparison results of different lightweight detection models.

Datasets	Model	mAP(%)	Params	FLOPs(G)
Pascal VOC	YOLOv3-tiny [43]	55.9	87.106M	13.0
	YOLOv4-tiny [44]	52.5	59.180M	16.2
	YOLOv5n [45]	70.6	17.862M	4.2
	YOLOX-tiny [46]	79.03	50.384M	15.17
	YOLOX-nano [46]	73.41	9.005M	2.49
	YOLOv7-tiny [47]	75.98	60.654M	13.3
	YOLOv8	86.54	25.292M	23.78
	Ours	91.95	12.986M	11.98
COCO	YOLOv3-tiny	90.97	86.706M	12.9
	YOLOv4-tiny	89.67	58.789M	16.2
	YOLOv5n	90.21	17.822M	4.2
	YOLOX-tiny	94.54	50.344M	15.15
	YOLOX-nano	93.25	8.604M	2.47
	YOLOv7-tiny	94.86	60.614M	13.1
	YOLOv8	95.25	25.253M	23.54
	Ours	97.93	12.916M	11.74

TABLE 7. Comparison results of different mainstream detection models.

Datasets	Model	mAP(%)	Params	FLOPs(G)
Pascal VOC	BiTnet [10]	73.4	10.0	10.5
	MCANet [13]	70.62	-	5.7
	SAI-YOLO [49]	77.59	22.6	-
	Ding [14]	74.73	45.14	-
	Gong [15]	90.26	17.678	42.986
	Ours	91.95	12.986	11.98

effectively diminishes the model's parameter count, thereby reducing computational expenses and enhancing processing speed. Nevertheless, the model's overall performance falls short of that of the baseline model. This discrepancy arises from the parameter reduction in the enhanced model, which leads to inadequate feature extraction by the backbone network, resulting in diminished model accuracy.

The incorporation of Soft-NMS serves to accentuate target semantic information and suppress extraneous details, consequently enhancing model accuracy with only a marginal rise in computational overhead.

Introducing the MA module and enhancing the backbone network leads to an enhancement in model accuracy when compared to the sole enhancement of the backbone network or the addition of Soft-NMS.

LAYN amalgamates the benefits of all three enhancement methods, culminating in an overall superior performance compared to YOLOv8.

Figure 7 illustrates a comparative analysis of detection outcomes between LAYN and various algorithms across diverse scenarios. It is evident that our algorithm demonstrates enhanced detection performance in scenarios featuring heavily occluded and densely populated small targets, outperforming other models. This enhancement can be attributed to the utilization of the LMA module in our methodology. The LMA module prioritizes information pertinent to small target detection, suppresses extraneous details, and efficiently lowers the confidence scores of detection boxes overlapping with the currently top-performing detection boxes. Rather than completely discarding these detections, this approach partly the occurrence of missed detections in dense scenarios, enabling the effective identification of large overlapping objects. Consequently, LAYN surpasses YOLOv8 in terms of detection performance.

Table 5 presents the model's performance after the incorporation of various attention mechanisms. In comparison to ECA, CBAM, and CA, the utilization of MA necessitates fewer parameters and yields heightened network coherence, rendering it better suited for deployment in lightweight networks. The inclusion of MA enables the model to extract more focused features, thereby augmenting its capacity to detect small targets.

Table 6 demonstrates that through the enhancement of YOLOv8's backbone network and the incorporation of the LMA module, our LAYN model achieves a 48.66% reduction in model parameters and a 49.6% decrease in FLOPs when compared to YOLOv8. Notably, the overall performance of this approach surpasses that of YOLOv8. In contrast to lightweight object detection networks such as YOLOv7-tiny, our proposed LAYN strikes a harmonious balance between speed and accuracy. Importantly, it outperforms alternative lightweight object detection algorithms in both performance and accuracy.

Table 7 shows that we compared LAYN with the SOTA algorithms proposed by Gong and others in the past two years on the PASCAL VOC dataset. The results indicate that LAYN surpasses the current optimal algorithm in all metrics, with a 1.69% increase in mAP, and reductions of 26.54% and 72.13% in Params and FLOPs, respectively. Compared to BiTNet and MCANet, although LAYN has a slight increase in Params and FLOPs, its accuracy far exceeds them, with mAP improvements of 18.55% and 21.33%, respectively. This suggests that LAYN successfully extracts features of small objects while reducing the model size, which significantly improves the accuracy of small object recognition. LAYN has achieved superior results compared to other methods in terms of mAP, Params, and FLOPs.

VI. CONCLUSION

In response to the demand for target detection algorithms suitable for embedded devices, we propose a Lightweight Multi-Scale Attention YOLOv8 small object detection algorithm. Firstly, we replace the YOLOv8 backbone network with GhostNet, reducing the parameter count and model size without compromising accuracy. Secondly, to enhance the network's acquisition of crucial information for small objects, we design a multi-scale attention module composed of a multi-scale mixed attention mechanism and the Soft-NMS method. This module effectively extracts finer-grained multi-scale spatial information, selectively choosing information crucial for small object detection, suppressing non-key information, and efficiently reducing false positives in detection results through the Soft-NMS method, thereby improving accuracy. Finally, in addition to using the PASCAL VOC dataset, a vehicle dataset compiled from the MS COCO dataset is also utilized to evaluate the proposed method. Experimental results demonstrate that compared to traditional YOLOv8, our proposed algorithm reduces FLOPs by 49.62%, decreases model parameters count by 48.66%, and increases mAP by 5.41% on the PASCAL VOC dataset, showcasing superior performance on common object detection datasets. On the vehicle dataset compiled from the MS COCO dataset, the mAP increases by 6.96%, confirming the effectiveness and strong generalization ability of the LAYN algorithm. Our proposed algorithm not only achieves lightweight deployment but also enhances detection accuracy, demonstrating its effectiveness in small object detection. Furthermore, compared with other lightweight algorithms, our proposed algorithm strikes a good balance between detection accuracy and lightweight deployment, showing excellent versatility. In contrast to existing models, the method proposed in this paper has lower computational costs, better detection accuracy, reduces the demand for computing power in embedded devices, and can be easily deployed on embedded devices with limited computational resources. Moving forward, our research will focus on further advancements in the following key aspects:

1. Explore self-supervised learning methods by designing appropriate self-supervised tasks, that allow the model to automatically learn useful features and thereby enhance the performance of small object detection.

2. Investigate incremental learning methods to enable the model to effectively learn from new data without having to retrain the entire model.

3. Conduct model quantization by converting model parameters into low-precision representations to reduce the model's size and computational complexity. Additionally, consider model pruning techniques to remove redundant and unnecessary network connections to improve the speed and efficiency of lightweight models.

REFERENCES

[1] X. Li, J. Deng, and Y. Fang, "Few-shot object detection on remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5601614, doi: 10.1109/TGRS.2021.3051383.

[2] Y. Ji, H. Zhang, F. Gao, H. Sun, H. Wei, N. Wang, and B. Yang, "LGCNet: A local-to-global context-aware feature augmentation network for salient object detection," *Inf. Sci.*, vol. 584, pp. 399–416, Jan. 2022, doi: 10.1016/j.ins.2021.10.055.

[3] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9626–9635, doi: 10.1109/ICCV.2019.00972.

[4] W. Han, J. Li, S. Wang, Y. Wang, J. Yan, R. Fan, X. Zhang, and L. Wang, "A context-scale-aware detector and a new benchmark for remote sensing small weak object detection in unmanned aerial vehicle images," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 112, Aug. 2022, Art. no. 102966, doi: 10.1016/j.jag.2022.102966.

[5] J. Zhong, J. Chen, and A. Mian, "DualConv: Dual convolutional kernels for lightweight deep neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 9528–9535, Nov. 2023, doi: 10.1109/TNNLS.2022.3151138.

[6] G. Cheng, X. Yuan, X. Yao, K. Yan, Q. Zeng, and J. Han, "Towards large-scale small object detection: Survey and benchmarks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13467–13488, Nov. 2023, doi: 10.1109/TPAMI.2023.3290594.

[7] B. Bosquet, D. Cores, L. Seidenari, V. M. Brea, M. Mucientes, and A. D. Bimbo, "A full data augmentation pipeline for small object detection based on generative adversarial networks," *Pattern Recognit.*, vol. 133, Jan. 2023, Art. no. 108998, doi: 10.1016/j.patcog.2022.108998.

[8] D. Cores, V. M. Brea, and M. Mucientes, "Spatiotemporal tubelet feature aggregation and object linking for small object detection in videos," *Int. J. Speech Technol.*, vol. 53, no. 1, pp. 1205–1217, Jan. 2023, doi: 10.1007/s10489-022-03529-w.

[9] J. Zhang, J. Lei, W. Xie, Z. Fang, Y. Li, and Q. Du, "SuperYOLO: Super resolution assisted object detection in multimodal remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5605415, doi: 10.1109/TGRS.2023.3258666.

[10] J. Zhao, H. Zhu, and L. Niu, "BiTNet: A lightweight object detection network for real-time classroom behavior recognition with transformer and bi-directional pyramid network," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 8, Sep. 2023, Art. no. 101670, doi: 10.1016/j.jksuci.2023.101670.

[11] M. Cui, G. Gong, G. Chen, H. Wang, M. Jin, W. Mao, and H. Lu, "LC-YOLO: A lightweight model with efficient utilization of limited detail features for small object detection," *Appl. Sci.*, vol. 13, no. 5, p. 3174, Mar. 2023, doi: 10.3390/app13053174.

[12] C. Liu, D. Yang, L. Tang, X. Zhou, and Y. Deng, "A lightweight object detector based on spatial-coordinate self-attention for UAV aerial images," *Remote Sens.*, vol. 15, no. 1, p. 83, Dec. 2022, doi: 10.3390/rs15010083.

[13] Z. Zhao, K. Hao, X. Liu, T. Zheng, J. Xu, S. Cui, C. He, J. Zhou, and G. Zhao, "MCANet: Hierarchical cross-fusion lightweight transformer based on multi-ConvHead attention for object detection," *Image Vis. Comput.*, vol. 136, Aug. 2023, Art. no. 104715, doi: 10.1016/j.imavis.2023.104715.

[14] P. Ding, H. Qian, Y. Zhou, and S. Chu, "Object detection method based on lightweight YOLOv4 and attention mechanism in security scenes," *J. Real-Time Image Process.*, vol. 20, no. 2, p. 34, Mar. 2023, doi: 10.1007/s11554-023-01263-1.

[15] L. Gong, X. Huang, J. Chen, M. Xiao, and Y. Chao, "Multiscale leapfrog structure: An efficient object detector architecture designed for unmanned aerial vehicles," *Eng. Appl. Artif. Intell.*, vol. 127, Jan. 2024, Art. no. 107270, doi: 10.1016/j.engappai.2023.107270.

[16] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS—Improving object detection with one line of code," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5562–5570, doi: 10.1109/ICCV.2017.593.

[17] X. Zhang, J. Wu, Z. Peng, and M. Meng, "SODNet: Small object detection using deconvolutional neural network," *IET Image Process.*, vol. 14, no. 8, pp. 1662–1669, Jun. 2020, doi: 10.1049/iet-ipr.2019.0833.

[18] L. Xiao, B. Wu, and Y. Hu, "Surface defect detection using image pyramid," *IEEE Sensors J.*, vol. 20, no. 13, pp. 7181–7188, Jul. 2020, doi: 10.1109/JSEN.2020.2977366.

[19] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016, doi: 10.1109/LSP.2016.2603342.

[20] B. Singh and L. S. Davis, "An analysis of scale invariance in object detection—SNIP," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 3578–3587, doi: 10.1109/CVPR.2018.00377.

[21] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)* (Lecture Notes in Computer Science), 2018, pp. 3–19.

[22] H. Fukui, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Attention branch network: Learning of attention mechanism for visual explanation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10697–10706.

[23] Y. Zhang, Y. Chen, C. Huang, and M. Gao, "Object detection network based on feature fusion and attention mechanism," *Future Internet*, vol. 11, no. 1, p. 9, Jan. 2019, doi: 10.3390/fi11010009.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jun. 2017, pp. 1–11.

[25] J. Fu, H. Zheng, and T. Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4438–4446.

[26] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6450–6458, doi: 10.1109/CVPR.2017.683.

[27] B. Zoph, E. D. Cubuk, G. Ghiasi, T.-Y. Lin, J. Shlens, and Q. V. Le, "Learning data augmentation strategies for object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 566–583.

[28] 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.

[29] M. Kisantal, Z. Wojna, J. Murawski, J. Naruniec, and K. Cho, "Augmentation for small object detection," 2019, *arXiv:1902.07296*.

[30] 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.*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[31] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, Jul. 2017, pp. 7263–7271, doi: 10.1109/CVPR.2017.690.

[32] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Apr. 2018, pp. 1–6.

[33] Q. Chen, Y. Wang, T. Yang, X. Zhang, J. Cheng, and J. Sun, "You only look one-level feature," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Nashville, TN, USA, Jun. 2021, pp. 13039–13048, doi: 10.1109/CVPR46437.2021.01284.

[34] P. Hurtik, V. Molek, J. Hula, M. Vajgl, P. Vlasanek, and T. Nejezchleba, "Poly-YOLO: Higher speed, more precise detection and instance segmentation for YOLOv3," *Neural Comput. Appl.*, vol. 34, no. 10, pp. 8275–8290, May 2022, doi: 10.1007/s00521-021-05978-9.

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

[36] J. Terven and D. Cordova-Esparza, "A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Apr. 2023, pp. 1–36.

[37] 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.

[38] 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)*, Honolulu, HI, USA, Jul. 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.

[39] 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.*, Salt Lake City, UT, Jun. 2018, pp. 8759–8768, doi: 10.1109/CVPR.2018.00913.

[40] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, Jun. 2020, doi: 10.1109/CVPR42600.2020.00165.

[41] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010, doi: 10.1007/s11263-009-0275-4.

[42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.

[43] P. Adarsh, P. Rathi, and M. Kumar, "YOLO v3-tiny: Object detection and recognition using one stage improved model," in *Proc. 6th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Coimbatore, India, Mar. 2020, pp. 687–694, doi: 10.1109/ICACCS48705.2020.9074315.

[44] C.-Y. Wang, A. Bochkovskiy, and H. M. Liao, "Scaled-YOLOv4: Scaling cross stage partial network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, Jun. 2021, pp. 13024–13033, doi: 10.1109/CVPR46437.2021.01283.

[45] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, Y. Kwon, J. Fang, K. Michael, D. Montes, J. Nadar, and P. Skalski, "Ultralytics/YOLOv5: V6. 1-TensorRT, TensorFlow edge TPU and OpenVINO export and inference," Zenodo, Tech. Rep., 2022, doi: 10.5281/zenodo.3908559.

[46] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," in *Proc. Workshop IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Aug. 2021, pp. 1–7.

[47] 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.

[48] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020, doi: 10.1109/TPAMI.2019.2913372.

[49] Z. Zhao, K. Hao, X. Ma, X. Liu, T. Zheng, J. Xu, and S. Cui, "SAI-YOLO: A lightweight network for real-time detection of driver mask-wearing specification on resource-constrained devices," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–15, Nov. 2021, doi: 10.1155/2021/4529107.

SONGZHE MA received the B.Eng. degree from Changchun University of Technology, China, in 2020, where he is currently pursuing the M.S. degree in computer technology. His research interests include computer vision and intelligent information processing.

HUIMIN LU received the degree from Xi'an Jiaotong University, and the master's and Ph.D. degrees in computer science and technology, in 2005 and 2010, respectively. She completed her research in computer science and technology with the Postdoctoral Program, Jilin University, in 2014. She was a Visiting Scholar with the University of Missouri, Columbia, USA, from 2016 to 2018. She is currently a Full Professor with the School of Computer Science and Engineering, Changchun University of Technology, China. She is a Ph.D. Supervisor in statistics, data science, and artificial intelligence, and a Master's Supervisor in computer science and technology, and electronic information. Her research interests include artificial intelligence and application, biometric recognition, computer vision, data analysis and mining, and other fields.

JIE LIU received the M.S. degree in computer technology from Jilin University, in 2006. His research interests include artificial intelligence and image processing.

YUNGANG ZHU received the Ph.D. degree in computer science from Jilin University, China, in 2012. He was a Visiting Research Fellow or a Postdoctoral Fellow with Vienna University of Technology, Austria, Dresden University of Technology, Germany, and the University of Trento, Italy. He is currently an Assistant Professor with the College of Computer Science and Technology, Jilin University. He has authored or coauthored more than ten papers on international journals or conferences. His current research interests include probabilistic graphical models, information fusion, statistical machine learning, and data mining, with applications to knowledge engineering. He is a Committee Member of the CCF Computer Applications Technical Committee and the CAAI Intelligent Service Technical Committee. He served on the program committee for several IEEE international conferences. He serves as an Associate Editor for IEEE Canadian Journal of Electrical and Computer Engineering.

PENGCHENG SANG received the B.S. degree in communication engineering from Shenyang Institute of Engineering, in 2020. He is currently pursuing the M.S. degree in electronic information and engineering with Changchun University of Technology. His research interests include pattern recognition and biometrics.