Signal, Image and Video Processing (2024) 18:5121–5136
https://doi.org/10.1007/s11760-024-03220-2

ORIGINAL PAPER

YOLO-MTG: a lightweight YOLO model for multi-target garbage detection

Zhongyi Xia[1,2] · Houkui Zhou[1,2] · Huimin Yu[3,4] · Haoji Hu[3] · Guangqun Zhang[1,2] · Junguo Hu[1,2] · Tao He[1,2]

Received: 28 February 2024 / Revised: 25 March 2024 / Accepted: 14 April 2024 / Published online: 20 May 2024

Abstract
With wide adoption of deep learning technology in AI, intelligent garbage detection has become a hot research topic. However, existing datasets currently used for garbage detection rarely involves multi-category and multi-target garbage that are densely accumulated in actual garbage detection scenarios. In addition, many existing garbage detection models have such problems as low detection efficiency and difficulties in integration with resource-constrained devices. To address the above situations, this study proposes a lightweight YOLO model for multi-target garbage detection (YOLO-MTG). This model is designed as follows: firstly, MobileViTv3, a lightweight hybrid network, serves as the feature extraction network to encode global representations, enhancing the model's ability of discriminating dense targets. Secondly, MobileViT block, the feature extraction unit, is optimized with combination of EfficientFormer and dynamic convolution, aiming to enhance the model's feature extraction capability, focusing on essential feature information and reduce the redundancy in useless information. Finally, feature reuse techniques are deployed to reconstruct Neck to minimize the loss of channel information in the feature transmission process, and maintain the strong feature fusion ability of the model. The experimental results on the self-built multi-target garbage (MTG) dataset show that YOLO-MTG achieves 95.4% mean average precision (mAP) with only 3.4 M parameters, and it is superior to other state-of-the-art (SOTA) methods. This work contributes new insights into the field of garbage detection, aiming to advance garbage classification for practical engineering applications.

Keywords Garbage detection · MobileViTv3 · EfficientFormer · Dynamic convolution · Feature reuse techniques

1 Introduction

Due to the increased population and rapid economic development, the garbage production has been growing wildly in a straight upward trend across the world over recent years. Especially, the phenomena of "garbage sieges" and "garbage villages" are common seen in developing countries [1]. If not properly disposed of, such a large amount of garbage will cause waste of huge resources and, more seriously, lead to devastating damage to the ecological environment. Garbage classification is not only an important foundation for realizing the harmlessness, reduction and resource utilization of garbage, but also an inevitable trend of social development [2]. However, at present, garbage classification mainly relies on manual labor, resulting in such problems as low sorting efficiency, high work intensity and poor sanitary conditions. With the rapid development of AI technology, intelligent garbage sorting expects to provide a new and effective solution to the problem of garbage classification. Therefore, it is of great academic value and social significance to seek efficient intelligent garbage classification methods.

Intelligent garbage classification requires efficient visual algorithms as technical support. Traditional visual algorithms manually extract low-level features of targets, which are suitable for single scenarios and exhibit poor robustness. The emergence of deep learning has broken through this limitation. This algorithm solves the problem of traditional

✉ Houkui Zhou
zhouhk@zju.edu.cn

1 School of Mathematics and Computer Science, Zhejiang A & F University, Hangzhou 311300, China

2 Zhejiang Provincial Key Laboratory of Forestry Intelligent Monitoring and Information Technology, Hangzhou 311300, China

3 College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China

4 State Key Laboratory of CAD & CG, Hangzhou 310027, China

feature extractors' inability to obtain high-level abstract representations of targets, making it the mainstream technology for image recognition today. Similarly, deep learning has achieved outstanding results in the field of garbage classification. Mao et al. [3] used genetic algorithm to fine-tune the parameters of the fully connected layer of DenseNet121. The optimized model achieved an accuracy of 99.6% on the TrashNet dataset. Feng et al. [4] combined attention mechanisms ECA and CA to redesign and simplify the structure of the MBConv module in EfficientNet. The improved model achieved classification accuracies of 94.54% and 94.23% on self-built garbage datasets and the TransNet dataset, respectively. Chen et al. [5] proposed a lightweight garbage classification model (GCNet). GCNet efficiently improves ShuffleNetv2 by introducing parallel hybrid attention mechanisms, FReLU activation function, and transfer learning. Experiments show that GCNet achieved 97.9% accuracy with only 1.3 M parameters on a self-built dataset, and the single recognition time on Raspberry Pi was 105 ms.

However, the aforementioned garbage classification models are based on single-object classification methods, exhibiting significant limitations in addressing multi-classification tasks, which restrict their universality in practical applications. In most real-world scenarios, a single image often contains various types of garbage, with each type occupying specific regions within the image. Consequently, more researchers have turned towards object detection research capable of simultaneously locating and identifying various types of garbage in images. Li et al. [6] proposed a PC-Net model for detection of floating garbage based on the Faster R-CNN, which introduced a pyramid anchor generation approach to reduce the interference of background information of garbage images, while the classification maps as feature maps were lead into the ROI pooling stage to enhance the feature information of small-target garbage; although the detection accuracy of the final improved model reached 86.4%, the model was too complex to detect targets in real time. Ma et al. [7] proposed a new feature fusion method for the SSD model, realizing effective fusion of garbage features in different scales, and introduced a focal loss function to solve the problem of imbalanced positive and negative sample proportions. The mAP of the proposed model reached 83.48% in five categories of single-target garbage datasets, but the detection speed was not sufficiently high. Jiang et al. [8] put forward a method for detecting rural household garbage based on YOLOv5s, and this method mainly optimized the detection performance by adding a small-target detection layer and using the attention mechanism; although the final improved model was tested on thirteen categories of household garbage datasets to deliver a detection accuracy of 96.4% and a detection speed of 47.6FPS. To solve the problem of aquatic environment pollution, Tian et al. [9] developed an underwater garbage cleaning robot based on the lightweight YOLOv4 model, realizing such functions as automatic detection, identification and collection of underwater garbage; however, because of few categories of garbage datasets used and monotonous garbage targets in the images, the method lacked the practicality of detection in complex aquatic environments. Luo et al. [10] developed a recyclable garbage detection system by integrating several end devices, an edge server and a cloud center, while designing an effective collaboration mechanism for the end devices to collect garbage images, for the edge server to detect the images, and for the cloud center to recover the images; although experiments showed that the overall accuracy of the system reached 90%, the detection model embedded in the edge server encountered a high inference latency due to heavy computation burdens. To realize automatic detection and recovery of recyclable garbage in the waste-to-energy plant scenarios, Cheng et al. [11] constructed a garbage detection system with a hardware module and then used the system to collect data and create a multi-target recyclable garbage dataset (REG); finally, the improved CenterNet model was used for training and testing on the REG dataset, with good detection results obtained. However, the REG dataset contained only three categories of garbage, thus lacking proper applicability in specific scenarios of waste-to-energy plants.

In summary, although the garbage detection models designed in the above researches have significantly boosted the accuracy and robustness and can be applied in various garbage detection tasks, most garbage datasets used by them were for single-target or less-target garbage and lacked rich categories, so these models could not sufficiently meet the need of actual detection scenarios, where multiple garbage piles are often assembled together, lacking practical application value. Although some researches have transplanted their garbage detection models into certain embedded modules, most models still have a mass of parameters or calculations, requiring a huge amount of computational resources. As a result, they cannot be well applied to resource-constrained devices, thus greatly affecting the detection performance. In addition, some garbage targets in actual garbage detection tasks have the characteristics of inter-class similarity, intra-class difference or mutual occlusion, making it difficult for models to extract useful feature information for discriminating targets in a proper manner; eventually, missed or false detection of garbage targets would be generated during the recognition process.

To solve the above problems, this study proposes a lightweight multi-target garbage detection model (YOLO-MTG) to meet the needs of applications in actual garbage detection scenarios. It mainly makes the following contributions:

1. To effectively improve the recognition accuracy of the model for multi-target garbage images, a lightweight

hybrid network, MobileViTv3, is introduced as the feature extraction network for global modeling, thereby strengthening the model's global representational capability.
2. To enhance the model's feature extraction capacity and capture irregular garbage features, the feature extraction operators of both local and global representation within the MobileViT block are redeployed. For local representation, dynamic convolution ODConv is employed to dynamically generate attention to learn local fine-grained features of targets. For global representation, EfficientFormer block is introduced to replace Transformer to simplify the global feature extraction process and reduce feature information redundancy.
3. To further reduce computational costs and accelerate inference speed, feature reuse techniques are employed to redesign the constituent units of the feature fusion network, Neck. Leveraging the information transmission properties of feature reuse, the model significantly retains the fusion and interaction capabilities of contextual information, thereby preserving detection performance.

2 Material and methods

2.1 MTG dataset

In this study, a multi-target garbage dataset (MTG) was created to evaluate the performance of the proposed model. The MTG mainly consists of two parts: an open-source multi-target garbage detection dataset from Haihua Research Institute, and a self-built garbage dataset, which was collected and screened from the Internet. The MTG contains 6,782 high-definition garbage images, with a total of 204 categories of abundant garbage, all of which are related to common household garbage. The images' resolutions range from 640*384 to 1920*1080, with up to 20 garbage targets per image. In most of the images, garbage targets are densely packed together, with some targets occluded. This pattern closely resembles the real-life garbage detection scenarios. Examples in the MTG are shown in Fig. 1. To rationalize the use of resources, we refer to the garbage classification standards of China [2], and reclassify garbage into nine categories: cardboard, glass, metal, paper, plastic, other recyclable waste, harmful waste, dry trash, and wet trash. For subsequent experiments, the MTG dataset was divided into a training set and a test set at a ratio of 7:3. Detailed information on the details of the self-built dataset is shown in Table 1.

2.2 Overview of YOLOv5

As one-stage detectors, YOLO series [12–14] play an important role in object detection. Especially, YOLOv5 [15] is an excellent object detection model, because it performs better in terms of detection accuracy and speed than the previous YOLO series. Due to small model size, simple structure and easy deployment in hardware devices, YOLOv5s, a lightweight version of YOLOv5, is widely used in the field of object detection. Therefore, YOLOv5s is selected as the benchmark model in this study. After further optimized, it can be efficiently deployed in widespread hardware devices to achieve accurate detection and recognition of multi-target garbage.

As shown in Fig. 2, the architecture of YOLOv5s consists of four parts: Input, Backbone, Neck and Prediction. The Input part mainly preprocesses the input images in the data augmentation method Mosaic, adaptive anchor box calculation and adaptive image scaling. The backbone consists of the feature extraction network CSPDarket53 and the SPP layer to learn more gradient information through the cross-stage hierarchy. Then, the Neck part adopts the combined structure of Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) to strengthen the feature fusion ability of the network for objects with different scaling sizes. Finally, the Prediction part is equipped with three couple heads of different scales, which are optimized by loss functions of the bounding box, confidence and classification to deliver accurate positioning and recognition of objects of different sizes.

2.3 The YOLO-MTG model

To meet the need of multi-category and multi-target garbage detection in real life, this study proposes a YOLO-MTG model by optimizing the architecture of YOLOv5s. The architecture of the proposed YOLO-MTG model is shown in Fig. 3. Firstly, MobileViTv3 [16], a lightweight hybrid network, is adopted as the feature extraction network. This ensures that the model can be embedded in resource-constrained devices in a small volume while fully paying attention to the global features of multi-target garbage. Secondly, based on the MobileViT block, a new feature extraction unit, ED-Mobile block, is proposed. In this block, the Linear Transformer in the global representation module is replaced with the EfficientFormer block [17] to reduce the computational burden of the model while maintaining detection accuracy; the dynamic convolution ODConv [18] is introduced to replace the convolutions in the local representation, focusing on the local fine-grained feature information of targets. Finally, In the feature fusion network, a new Thin-neck is constructed based on two feature reuse techniques, Ghost Module [19] and RepGhost Module [20], so as to further lighten the model, while retaining the fusion and interaction capabilities for context information. In Thin-neck, the Ghost Module based on cheap operation replaces the complex convolution operations in PAN, thus alleviating the problem of excessive computational costs incurred in convolutions. In addition, the RepGhost Module based on

Fig. 1 Examples in the MTG dataset

Table 1 Details of the self-built dataset

	Training			Testing			Total
Size	640 * 384 to 1920 * 1080			640 * 384 to 1920 * 1080			
Number of instances in each category	ca: 969	gl: 1170	me: 3380	ca: 314	gl: 687	me: 1249	40,687
	pa: 1096	pl: 2945	re: 3765	pa: 643	pl: 2129	re: 1602	
	ha: 3614	dr: 4672	we: 4527	ha: 1661	dr: 2895	we: 3369	
	Total: 26,138			Total: 14,549			
Number of images	4649			2133			6782
Strategy	Mosaic, resize (640*640)			Resize (640*640)			

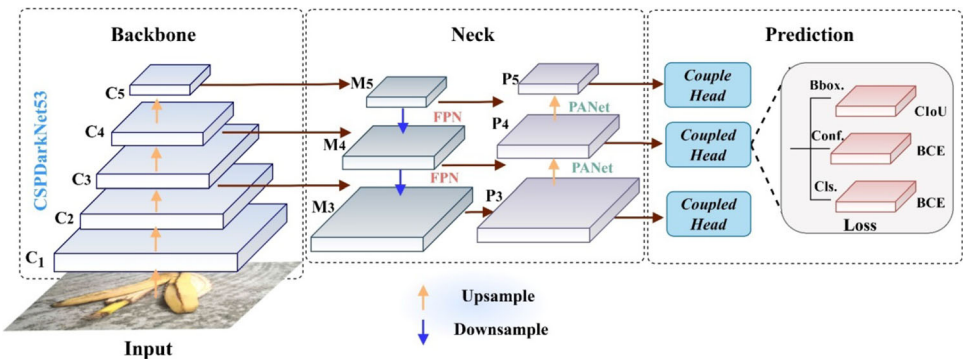

Fig. 2 The overall architecture of YOLOv5s model

structural re-parameterization is deployed to reconstruct C3, obtaining a new RepGhostC3. The aforementioned improvement methods will be detailed in the following sections.

2.4 MobileViTv3

The original YOLOv5s operates convolutions to extract features. Nevertheless, due to the limited receptive field of the convolution, only local feature information of images may be extracted, thus easily leading to incomplete extraction or even loss of target feature information when dealing with multi-target detection tasks. ViTs [21, 22] perform well in intensive multi-target detection tasks by dividing an input image into a sequence of non-overlapping patches, while learning the inter-patch complex dependencies through multi-headed self-attention mechanism (MHSA), so as to obtain global representations. However, the above models all suffer from a common problem: large computational

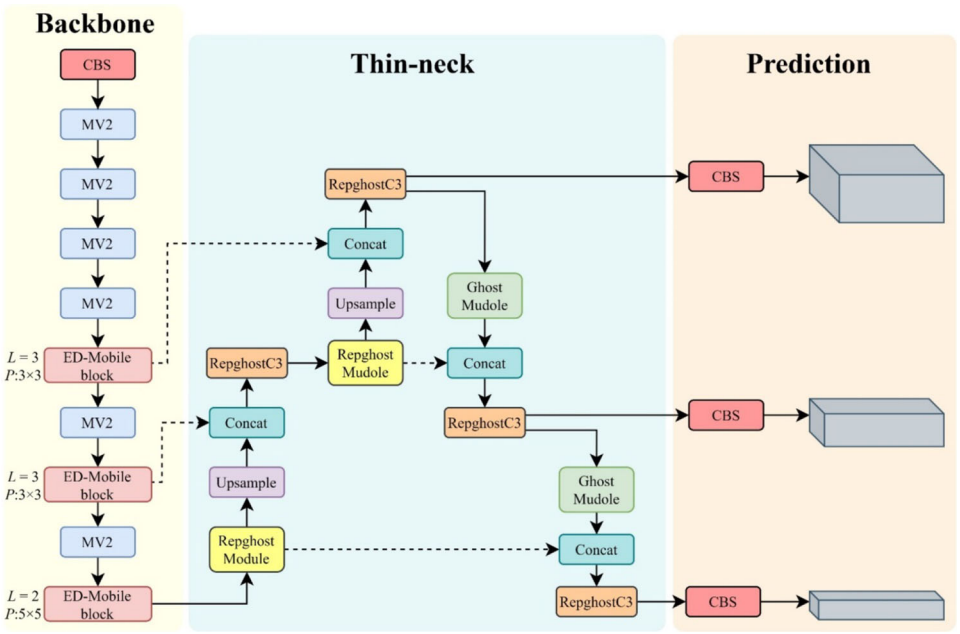

Fig. 3 The architecture of the proposed YOLO-MTG model. Here, CBS denotes a convolution operation combining convolution, batch normalization and activation function SiLU; MV2 denotes the MobileNetv2 block;L denotes the number of stacks of the EfficientFormer block in each ED-Mobile block; P denotes the pool kernel size in the EfficientFormer block

costs, making it difficult for them to make effective inference in resource-constrained devices. In contrast, the MobileViT series [23, 24] are lightweight networks built by combining the advantages of CNNs and ViTs, so they can easily achieve the efficient extraction of both local and global features of images with fewer parameters. As a result, MobileViTs provide a good option for deployment in resource-constrained devices.

This study introduces MobileViTv3 [16], an improvement of MobileViTv2, as the feature extraction network. In architecture, MobileViTv3 is mainly composed of the MobileNetv2 block and the MobileViT block. Among them, the MobileViT block is an important factor in determining the network performance. As shown in Fig. 4a, the architecture of MobileViT block consists of three sub-modules for local representations, global representations and fusion. Specifically, for a given input feature $X \in \mathbb{R}^{H\times W\times C}$, the local spatial information is firstly encoded by a 3×3 Depthwise convolution (DWConv), followed by projecting the feature tensor into the low-dimensional space using point-wise convolution, so as to obtain the local feature $X_{\mathrm{L}} \in \mathbb{R}^{H\times W\times C/2}$. Secondly, X_{L} is unfolded into N non-overlapping flattened patches $X_{\mathrm{U}} \in \mathbb{R}^{C/2\times P\times N}$, where $P = wh$ and $N = HW/P$, w and h are the length and width of each patch; then inter-patch relationships are encoded through Linear Transformer, so as to obtain the global representation $X_{\mathrm{G}} \in \mathbb{R}^{H\times W\times C/2}$. X_{G} can be expressed as:

$$X_{\mathrm{G}} = Linear\ Transformer(X_{\mathrm{L}}) \tag{1}$$

Then, X_{G} is folded to obtain the global feature $X_{\mathrm{F}} \in \mathbb{R}^{H\times W\times C/2}$. Finally, X_{L} is fused with X_{F} through concatenation, so as to obtain the fused feature $X_{\mathrm{FO}} \in \mathbb{R}^{H\times W\times C}$. Afterwards, X_{FO} is fed into 1×1 convolution, and the residual connection is introduced to obtain the final output feature $Y \in \mathbb{R}^{H\times W\times C}$. Y can be expressed as:

$$Y = X + Conv(Cat(X_{\mathrm{F}} + X_{\mathrm{L}})) \tag{2}$$

2.5 ED-Mobile block

The details of the ED-Mobile block are shown in Fig. 4b. The ED-Mobile block proposed retains the design structure of MobileViT block. The ODConv and EfficientFormer blocks replace the elements of the original local and global representation modules, respectively. ED-Mobile block is capable of fully extracting complex garbage features, improving model accuracy, and its simple dimensionally consistent design can speed up model inference.

2.5.1 EfficientFormer

In some recent research, ViT-based models have been extended to the MetaFormer architecture [25, 26]. Metaformer refers to the MHSA module in Transformer as Token Mixer, providing a generic architecture consisting of a Token Mixer and MLP. PoolFormer, as the starting network in the MetaFormer architecture, only replaces MHSA with a simple spatial pooling operator as Token Mixer, but can deliver competitive performance in multiple visual tasks. The

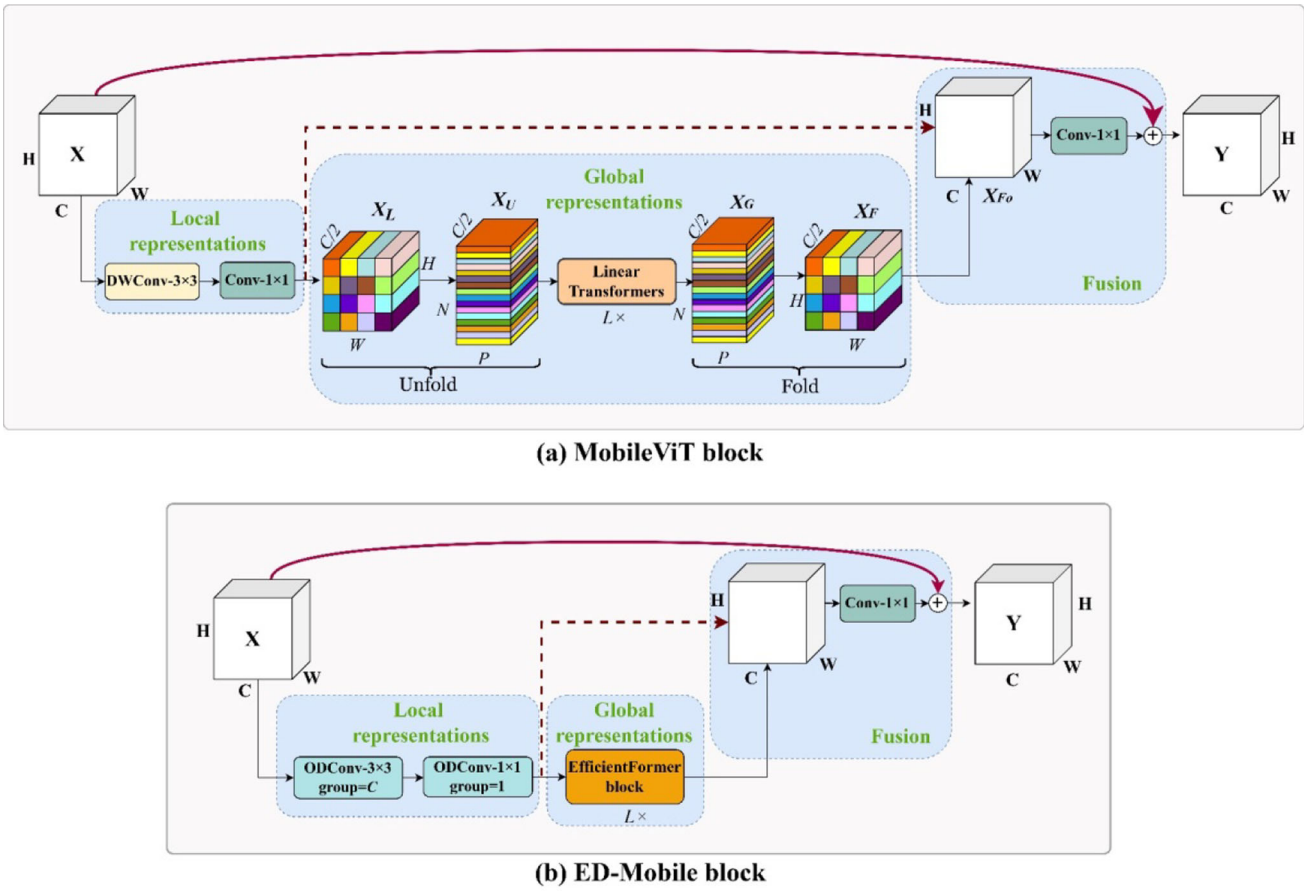

Fig. 4 The architectures of the **a** MobileViT block and the **b** ED-Mobile block. Here,L denotes the number of stacks

architecture of PoolFormer block is shown in Fig. 5a. When replacing the Linear Transformer with the PoolFormer block, we found that the mAP was reduced by only 0.4% with a significant reduction in computational costs. Therefore, PoolFormer provides an effective improvement strategy.

EfficientFormer [17] provides a new way to mitigating Transformers' latency by revisiting the design principles of the structure through a delay analysis of ViT and its variant models. It combines the performance advantages of PoolFormer and ViT, and design an efficient inference architecture with consistent dimensions. The EfficientFormer block proposed therein is an improvement to the PoolFormer block. As shown in Fig. 5b, the EfficientFormer block follows the MetaFormer architecture's design, mainly composed of two sub-blocks: Token Mixer and MLP. The two sub-blocks can be expressed as:

$$Y = Pooling(X) + X \tag{3}$$

$$Z = Conv_{\mathrm{B}}\big(Conv_{\mathrm{B,S}}(Y)\big) + Y \tag{4}$$

where X and Y denote the input and output of Token Mixer, respectively; Z denotes the output of MLP; $Pooling$ denotes a pooling operator as Token Mixer. B and S in $Conv_{\mathrm{B,S}}$ denote whether there is BN and SiLU, respectively, after the convolution operation.

The EfficientFormer block eliminates Layer Normalization (LN) layer in Token Mixer, replaces the original combination of 3D linear projection and LN with the combination of Batch Normalization (BN) and 1×1 convolution in spatial MLP, and removes the complex Reshape operation, while unifying the feature dimensions. Moreover, it takes full advantage of the fusion mechanism of convolution and BN during model inference, so as to accelerate the inference speed.

2.5.2 ODConv

To solve the problem of missed and false detection of irregular garbage targets, this study adopts the dynamic convolution ODConv [18]. Dynamic convolutions [27, 28] can dynamically generate attention weight values and adjust convolution parameters according to different input features, rather than directly calibrating the input features via attention mechanisms. Therefore, this dynamic learning method makes

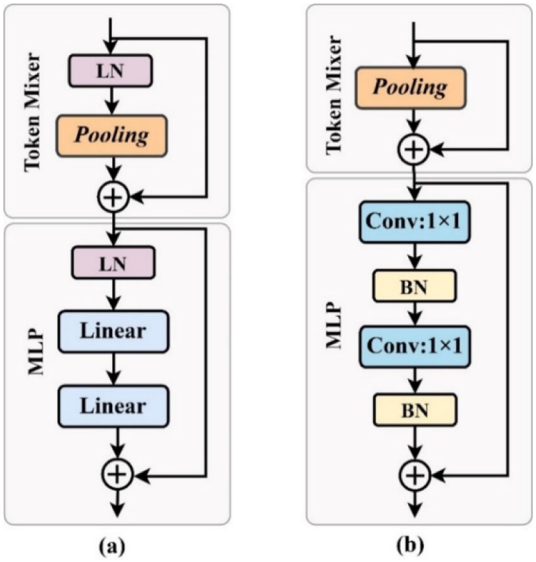

Fig. 5 Two model blocks based on the MetaFormer architecture, **a** PoolFormer block; **b** EfficientFormer block

convolution no longer a simple linear function, but nonlinear reinforcement of the convolution kernel via attentions, so that the network can make full use of the convolution kernel in extracting image features.

As a dynamic convolution based on omni-dimensional attentions, ODConv learns four complementary attentions along the four dimensions of the kernel space. The schematic of ODConv is shown in Fig. 6. Specifically, for an input feature x, it would be first squeezed into a $1 \times 1 \times C_{\text{in}}$ feature vector by Global Average Pooling (GAP), and then the squeezed feature vector is mapped to a low-dimensional space with a dimensionality reduction ratio $r = 1/16$ by using the Fully Connected (FC) layer. Later on, the feature vector is computed in parallel along the spatial dimension, the input channel dimension, the output channel dimension and the kernel dimension of the convolution kernel W_n, i.e., four parallel FC layers, to obtain the output size of $k \times k$, $C_{\text{in}} \times 1$, $C_{\text{out}} \times 1$ and $n \times 1$, respectively; then, the attention weights, α_{sn}, α_{cn}, α_{fn} and α_{wn}, are generated using Sigmoid or Softmax function. Finally, the generated attention weights are weighted and summed with the corresponding convolution kernels to acquire a new convolution, the output features y of ODConv can be obtained by feature extraction of the input features through this convolution. Thus, ODConv can be expressed as:

$$y = (\alpha_{\text{w1}} \odot \alpha_{\text{f1}} \odot \alpha_{\text{c1}} \odot \alpha_{\text{s1}} \odot W_1 + \cdots + \alpha_{\text{wn}} \odot \alpha_{fn} \odot \alpha_{cn} \odot \alpha_{sn} \odot W_n) * x \tag{5}$$

where $\alpha_{sn} \in \mathbb{R}^{k \times k}$, $\alpha_{cn} \in \mathbb{R}^{C_{in}}$, $\alpha_{\text{fn}} \in \mathbb{R}^{C_{out}}$, and $\alpha_{\text{wn}} \in \mathbb{R}$ denote the attention weights computed along the spatial dimension, the input dimension, the output channel dimension, and the kernel dimension of the convolution kernel W_n, respectively; $\odot$ denotes multiplication operations in different dimensions; $+$ denotes an addition operator; $*$ denotes a new convolution formed by aggregating the four attention weights.

2.6 Thin-neck

This study leverages the advantages of feature reuse techniques, Ghost Module and RepGhost Module, to optimize Neck, proposing a new lightweight Thin-neck. The Thin-neck, with fewer parameters and computations, preserves the robust feature fusion capability of the original Neck, achieving efficient detection.

2.6.1 Ghost module

The Ghost Module [19] maintains the network capacity by reusing features via concatenation. As shown in Fig. 7a, the architecture of the Ghost Module integrates the convolution into two sections: The first section uses a standard convolution to reduce dimensions and generate a certain number of feature maps; and the second section completes the missing feature maps through linear computation (DWConv). Then, the feature maps generated in the first section are reused via concatenation; and finally, rich and low-redundancy high-dimensional features are obtained. Specifically, the feature reuse technique of the Ghost Module can be expressed as:

$$y = Cat\left(\left[x, \Phi_1(x), \ldots, \Phi_{n-1}(x)\right]\right) \tag{6}$$

where Cat denotes the concatenation; n denotes a scale factor controlling the dimension size; $y \in \mathbb{R}^{C_{\text{out}} \times H \times W}$ denotes the output feature after feature reuse; $x \in \mathbb{R}^{C_{\text{in}} \times H \times W}$ denotes the input feature to be processed and reused; $\Phi_i(x)$, $\forall i = 1, \cdots, \text{n} - 1$ denotes the linear computation Φ of DWConv, which generates feature tensors with a total dimension size of $((n-1)/n) * C_{\text{out}}$.

Unlike the standard convolution operation, the Ghost Module can eliminate a large number of redundant features and reduce the amounts of parameters and the computational complexity without changing the size of the output feature map, while maintaining proper network performance. Therefore, the Ghost Module is often used to replace some complicated convolution operation. However, its concatenation operation consumes huge memory resources, thus imposing non-negligible computing costs on hardware devices, while affecting the actual inference speed. As a result, it was not used much in the final improved model.

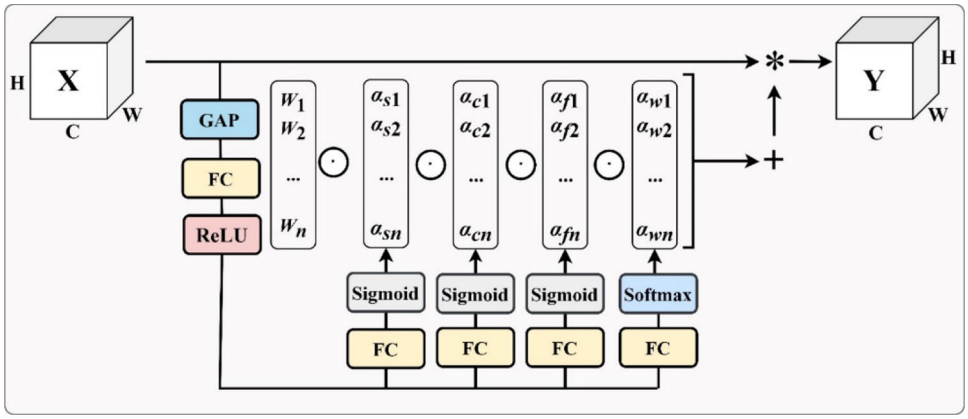

Fig. 6 The schematic of the ODConv

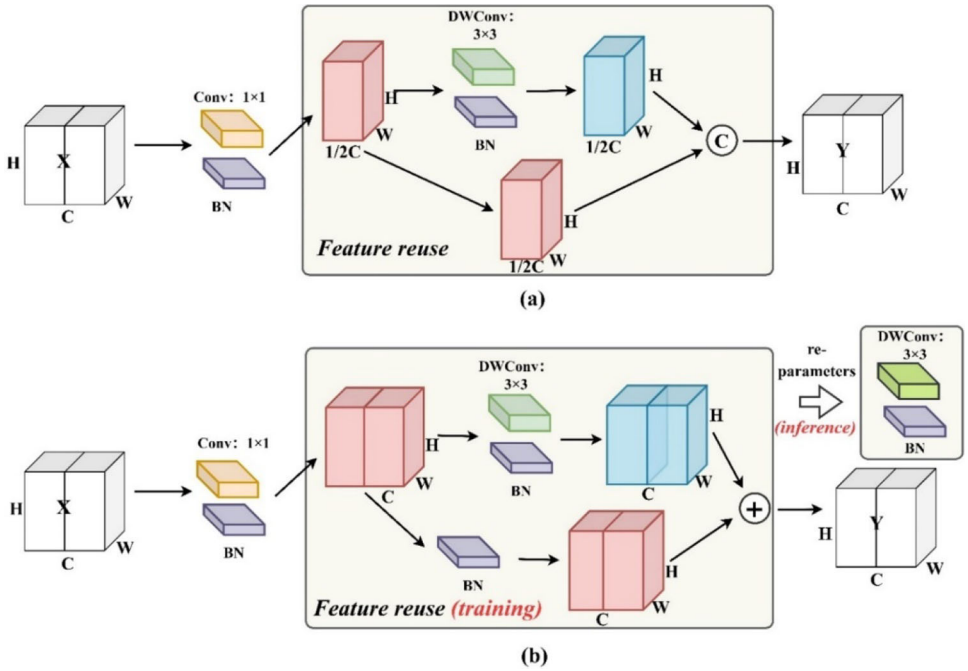

Fig. 7 The architecture of two feature reuse techniques, **a** Ghost Module; **b** RepGhost Module

2.6.2 RepGhost module

The RepGhost Module [20] reuses features efficiently via structural re-parameterization. Its architecture is shown in Fig. 7b. To solve the problem of heavy memory resource consumption in concatenation, the RepGhost Module replaces the hardware-inefficient concatenation with a hardware-efficient addition operator, while implementing structural re-parameterization in the DWConv part to generate different feature mappings. Feature reuse is implicitly achieved by fusing the features acquired from different layers, and the interaction ability for contextual information is thus strengthened. Specifically, the feature reuse technique of the RepGhost Module can be expressed as:

$$y = Add\left(\left[x, \Phi_1(x), \ldots, \Phi_{n-1}(x)\right]\right) = \Phi^*(x) \tag{7}$$

where Add denotes the addition operator; $\Phi_i(x), \forall i = 1, \cdots, \mathrm{n} - 1$ denotes the feature tensor generated by the linear computation Φ at a total dimension size of $n * C_{\text{out}}$. $\Phi_i(x)$ and x will be eventually fused into $\Phi^*(x) \in \mathbb{R}^{C_{\text{out}} \times H \times W}$.

Additionally, the feature weights in the branch (BN layer) are fused into the weights of the main branch (DWConv and BN layers) in the inference process, forming a simple inference block, i.e., a 3×3 DWConv. In this way, the fusion process is transferred from feature space to weight space. Without any time cost, the complex parallel structure during training is transformed into a simple serial structure during inferring, thus facilitating the model to perform efficient inference on common hardware devices.

Therefore, this study uses the RepGhost Module, which has efficient feature reuse capability, to replace the convolution operation in the original Bottleneck and formulates

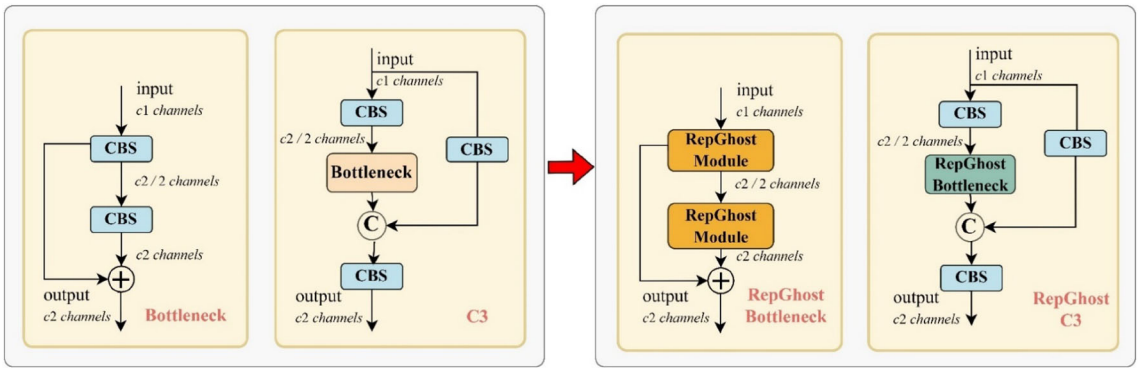

Fig. 8 The architecture of RepGhostBottleneck and RepGhostC3, as improved from the original Bottleneck and C3

Table 2 Experimental environment and Parameter settings

Item	Strategy
CPU	Intel I7-13700K (5.4 GHz)
GPU	1* NVIDIA RTX 3090 (24 GB)
Operate system	Windows 11
Deep learning environment	PyToch 1.10.0, Python 3.8, CUDA 11.3
Optimizer	Stochastic Gradient Descent (SGD)
Momentum	0.937
Weight decay	0.0005
learning rate schedule	Warmup to initial:0.01 (CosineAnnealing)
Training epochs	300
Batch size	8
Data augmentation	Mosaic (random cropping/scaling)
Image size	Resized 640*640 (train & test)

RepGhostBottleneck, resulting in the new RepGhostC3, as shown in Fig. 8. RepGhostC3 preserves the cross-stage hierarchy, and learns rich gradient combination information with fewer parameters.

3 Experiments and results

3.1 Experimental setup

The relevant experiments covered in this paper were conducted in accordance with the experimental settings in Table 2, to ensure the fairness of the experimental sessions.

3.2 Evaluation metrics

In this study, the evaluation metrics for model performance include: precision (P), recall (R), mean average precision (mAP), parameters (Params), floating-point operations per second (FLOPs), and frames per second (FPS). Among them, P indicates the proportion of correctly predicted positive samples to all predicted positive samples, R indicates the proportion of correctly predicted positive samples to all actual positive samples, and mAP shows how well the predicted samples match the true samples. Params, FLOPs and FPS are all important indicators to measure the performance of lightweight models. They indicate a model's space complexity, time complexity and detection speed, respectively. The formulas for P, R, mAP and FPS are as follows:

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

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

$$\mathrm{mAP} = \frac{1}{n}\sum_{i=1}^{n} AP_{\mathrm{i}} \tag{10}$$

$$\mathrm{FPS} = \frac{1}{t} \tag{11}$$

where TP (true positive) denotes the number of samples correctly judged as positive; FP (false positive) denotes the number of samples incorrectly judged as positive; FN (false negative) denotes the number of samples incorrectly judged as negative; AP_i denotes the area of the P–R curve of the ith category; and t denotes the time required for processing an image.

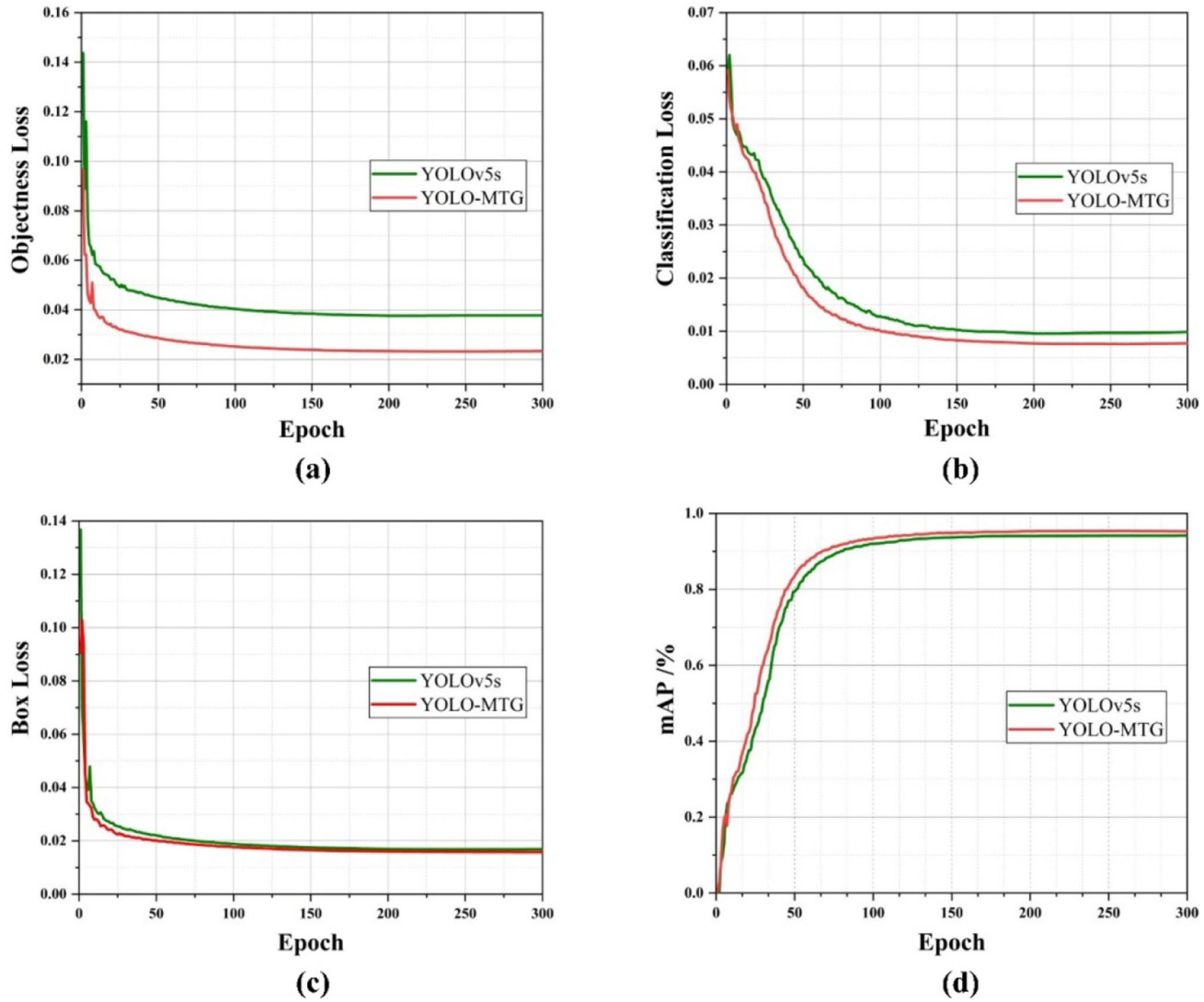

Fig. 9 Changes in loss values and mAP during training, **a** objectness loss; **b** classification loss; **c** bounding box loss; **d** mAP

3.3 Analysis of model training

To verify the correctness and rationality of the designed model, we recorded the loss values and mAP values produced during the training process for both YOLO-MTG and baseline YOLOv5. The relationship between loss values, mAP values, and the number of model iterations is depicted in Fig. 9. Regarding the loss aspect, YOLO-MTG exhibits a faster and more substantial decrease in loss values compared to YOLOv5s. This suggests that the actual detection results of YOLO-MTG are closer to the real labels, showcasing advantages in positioning and identification. Regarding the accuracy aspect, the mAP curve of YOLO-MTG steadily rises, converging rapidly until it approaches fitting, ultimately achieving a higher mAP value than YOLOv5s. This demonstrates that the improved model has stronger learning capabilities.

3.4 Ablation experiment

An ablation experiment was conducted on the MTG dataset to verify the effectiveness of the improved method proposed in this study, with the experiment results illustrated in Table 3. It can be observed that the improvement strategies of all modules in this study have helped boost the detection performance. Model A represents the original YOLOv5s model, which serves as the baseline network for this experiment, with relatively good results achieved on the MTG dataset.

Model B introduces MobileViTv3 model as a feature extraction network. Although the improved model is lightweight, it benefits from the powerful global feature extraction ability of Transformer, with an increase of 0.3%, 0.6% and 0.7% in the values of P, R and mAP, respectively, over those of YOLOv5s. However, this also indirectly leads

Table 3 Ablation experiment

Model	MobileViT	EF block	ODConv	Thin-neck	Params/M	FLOPs/G	*P*/%	*R*/%	mAP/%	FPS
A					7.1	16.4	92.4	90.7	94.2	114
B	√				5.5	20.3	92.7	91.3	94.9	87
C	√	√			4.6	17.5	92.5	91.4	94.9	109
D	√	√	√		4.7	17.1	93.8	92.4	95.5	95
E	√	√	√	√	3.4	14.8	93.7	91.9	95.4	102

to the increase of the model's computation burden, which affects the speed of the model's detection.

Model C introduces EfficientFormer (EF) block on top of Model B. The experimental results show that after introducing the EfficientFormer Block, the model could achieve the same mAP as the original MobileViTv3, with a significant reduction in the Params and FLOPs, while the detection speed is enhanced by 25%.

Model D introduces dynamic convolution ODConv on top of Model C. The experimental results show that the dynamic convolution has increased the mAP from 94.9 to 95.5% and cut a certain amount of FLOPs, while the params and detection speed have not changed much.

Model E, i.e., YOLO-MTG model, replaces the original Neck with a reconstructed Thin-neck. As seen in the experimental results, the method can greatly reduce the amount of parameters as compared with Model D, with a mAP loss of only 0.01%; and the model still maintains strong detection performance. The improved model YOLO-MTG, delivers a 1.2% higher mAP than YOLOv5s on the MTG dataset, with only 3.4 M parameters and slightly reduced FLOPs, which ensures that the improved model can be well deployed in hardware modules. Although the detection speed has dropped by 12FPS, it still meets the requirements of real-time garbage detection.

3.5 Comparative experiment in MobileViTv3 method

To verify the superiority of MobileViTv3 as a feature extraction network, this study uses Model A in Table 3 as the baseline model to compare with other mainstream lightweight CNNs, including ShuffleNetv2 [29], MobileNetv3 [30], GhostNet [19], GhostNetv2 [31], and lightweight hybrid networks, MobileViT [23], MobileViTv2 [24] and ParC-Net [32]. According to the comparison results shown in Table 4, the detection performance of the models using the lightweight CNNs on the MTG dataset is substantially degraded, with their detection speed even not exceeding that of the baseline for fewer params and FLOPs. In contrast, the lightweight hybrid network delivers strong performance; especially, when MobileViTv3 is used as the backbone, the detection accuracy of the model reaches the highest. Except for high FLOPs and slow detection speed, other performance indicators are all ahead of the baseline.

3.6 Comparative experiment in ODConv method

To verify the superiority of introducing ODConv, Model C in Table 3 is used as the baseline model, with other dynamic convolution methods, i.e., CondConv [27] and DyConv [28], selected for comparison. In addition, certain mainstream attention mechanisms, including SE [33], CBAM [34], ECA [35] and CA [36], are added for comparison with ODConv. The comparison results are shown in Table 5, although the models that implement dynamic convolutions demand more params, they are with fewer FLOPs and can deliver better detection accuracy than the model that adds attention mechanisms. Moreover, compared with other models that introduce dynamic convolutions, the model introducing ODConv needs fewer params but delivers higher detection accuracy, so it has more advantages in multi-target garbage detection tasks.

3.7 Comparative experiment in thin-neck method

To verify the effectiveness of the proposed Thin-neck, Model D in Table 3 is used as the baseline model, with two other lightweight Neck methods selected for comparison. One method, indicated as Ghost-neck, replaces the convolutions and C3 in the original Neck with the GhostConv and C3Ghost as proposed in the YOLOv5 project; and the other method is a recently proposed method, Slim-neck [37]. According to the comparison results shown in Table 6, the model introducing Ghost-neck has the fewest params, but its mAP is reduced by 0.4%, and its detection speed is the slowest. Although the model introducing Slim-neck has improved the detection speed, its mAP is the lowest, with a difference of 0.8% from the baseline, and it requires more params than the other two methods. Therefore, the advanced method shows no benefits in multi-target garbage detection tasks. In contrast, the model introducing Thin-neck removes 28% of the parameters, but its mAP is only 0.1% less than that of the baseline, while its detection speed is slightly improved, so its overall detection performance is completely superior to that of the other two models.

Table 4 Comparison of MobileViTv3 with other lightweight networks

Model	Params/M	FLOPs/G	mAP/%	FPS
A (baseline)	7.1	16.4	94.2	114
+ShuffleNetv2	3.6	7.5	91.8	103
+MobileNetv3	5.7	10.1	92.9	102
+GhostNet	5.3	8.6	92.8	87
+GhostNetv2	6.2	9.1	92.9	69
+MobileViTv1	4.6	18.0	93.6	83
+MobileViTv2	5.3	19.9	94.7	90
+ParC-Net	4.6	18.9	94.1	85
+MobileViTv3	5.5	20.3	94.9	87

Table 5 Comparison of ODConv with other dynamic convolutions and attention mechanisms

Model	Params/M	FLOPs/G	mAP/%	FPS
C (baseline)	4.6	17.5	94.9	109
+SE	4.6	17.5	95.2	103
+CBAM	4.6	17.5	95.3	96
+ECA	4.6	17.5	95.1	106
+CA	4.6	17.5	95.2	100
+CondConv	5.6	17.0	95.3	98
+DyConv	5.2	17.1	95.4	96
+ODConv	4.7	17.1	95.5	95

Table 6 Comparison of thin-neck with other lightweight neck methods

Model	Params/M	FLOPs/G	mAP/%	FPS
D (baseline)	4.7	17.1	95.5	95
+Ghost-neck	3.3	14.6	95.1	88
+Slim-neck	4.2	14.4	94.7	99
+Thin-neck	3.4	14.8	95.4	102

3.8 Model performance comparison

To verify the superiority of the YOLO-MTG over other object detection models, this study investigates on the MTG dataset the performance of some currently state-of-the-art single-stage detection algorithms, including YOLOXs [38], YOLOv6s [39] and YOLOv7-tiny [40]. According to the comparison results shown in Table 7, among the five models, YOLO-MTG delivers the highest mAP, reaching 95.4%, and needs the fewest Params, only 3.4 M; its FOLOPs is 14.8G, and its detection speed is 102FPS. Besides being slightly inferior to YOLOv5s and YOLOv7-tiny in terms of detection speed, it is sufficient to meet the requirements of real-time garbage detection. However, YOLOXs and YOLOv6s perform poorly on the MTG dataset, even though they require more params and FLOPs to train the model; and their detection accuracy does not even exceed that of YOLOv5s, and their detection speed is slower. Although the detection speed of YOLOv7-tiny is the fastest, YOLO-MTG showcases more advantages in terms of detection accuracy and params.

Figure 10 visually assesses the performance advantages and disadvantages of each model. It can be seen that the YOLO-MTG proposed is highlighted as more comprehensive compared to other detection models, and it can achieve a proper trade-off between detection accuracy and speed, while the model size is suitable for deployment on hardware modules, thus having practical application significance.

3.8.1 Analysis of model robustness

To better present the practical detection advantages of YOLO-MTG intuitively, this study simulates garbage detection scenarios under indoor conditions with normal, low, and bright conditions levels to examine the detection performance of YOLO-MTG in coping with different complex

Table 7 Comparison of YOLO-MTG with other advanced detection models

Model	Params/M	FLOPs/G	mAP/%	FPS
YOLOv5s	7.1	16.4	94.2	114
YOLOXs	8.9	26.7	93.7	81
YOLOv6s	17.2	44.1	94.1	75
YOLOv7-tiny	6.0	**13.2**	95.2	**131**
YOLO-MTG	**3.4**	14.8	**95.4**	102

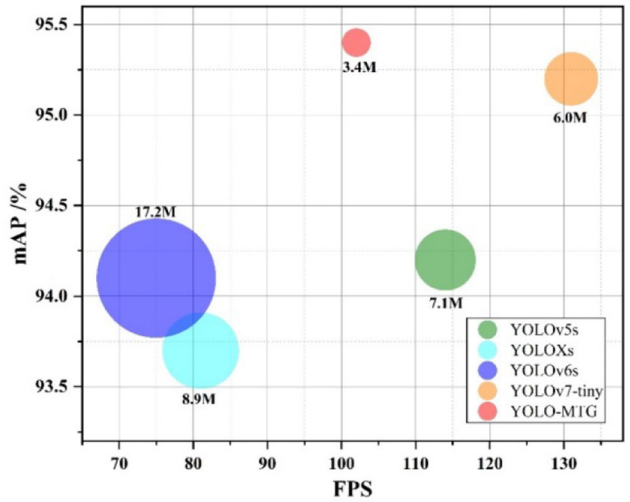

Fig. 10 Performance analysis of YOLO-MTG with other advanced models

conditions, and uses YOLOv5s and YOLOv7-tiny as detection comparison examples. The detection results, as depicted in Fig. 11, indicate instances of wrong detection and missed detection, indicated by yellow arrows. Under normal light conditions, garbage representations in the images are clearly visible, with YOLO-MTG demonstrating robust detection performance. However, both YOLOv5s and YOLOv7-tiny exhibit instances of wrong detection and missed detection, with a more pronounced occurrence of missed detections for occluded targets compared to YOLO-MTG. Under low light conditions, the quality of garbage representations decreases significantly due to reduced illumination, with some garbage features (e.g., transparent garbage) being obscured by low-light conditions. All three models experience a decline in detection performance, but YOLO-MTG maintains relatively good performance compared to the other models. Under bright light conditions, reflections on garbage representations are enhanced, especially the features of light-colored garbage are assimilated by the background, and the shadow phenomenon is also aggravated. YOLOv5s exhibits weak performance in handling bright light conditions, and both YOLOv7-tiny and YOLOv5s, have serious shadow misdetections, whereas YOLO-MTG effectively addresses these challenges. In summary, YOLO-MTG demonstrates strong adaptability and robustness under complex conditions, making it more suitable for deployment and application in practical detection scenarios.

Table 8 Comparison results on public dataset Trash_ICRA19 and Pascal VOC2012

Model	Params/M	mAP/%	
		ICRA19	VOC2012
YOLOv5s	7.1	97.8	62.6
YOLOXs	8.9	98.4	63.7
YOLOv6s	17.2	99.1	65.1
YOLOv7-tiny	6.0	97.3	63.1
YOLO-MTG	3.4	98.0	63.4

3.8.2 Analysis of model generalization

To further analyze whether YOLO-MTG exhibits generalizability in other detection task scenarios, this study conducted comparative experiments on the publicly available underwater garbage detection dataset Trash_ICRA19 [41] and the classic dataset Pascal VOC2012 [42]. The comparative results, as shown in Table 8, indicate that YOLO-MTG achieves mAP values of 98.0% and 63.4% on Trash_ICRA19 and VOC2012, respectively, surpassing both YOLOv5s and YOLOv7-tiny, while the model volume is only 48% and 57% of theirs. Additionally, YOLO-MTG achieves a comparable mAP to the larger YOLOXs model. These results suggest that YOLO-MTG efficiently models image information with fewer parameters, thereby meeting the requirements for portability and real-time performance on embedded devices. Thus, YOLO-MTG is equally applicable to other detection task scenarios and exhibits strong generalizability.

3.9 Garbage detection system

Based on the proposed YOLO-MTG, this study develops a garbage detection application system on a PC platform with the Qt framework. The system is capable of static detection of both single and multiple images in a single folder. For real-time detection, it is equipped with a video detection function

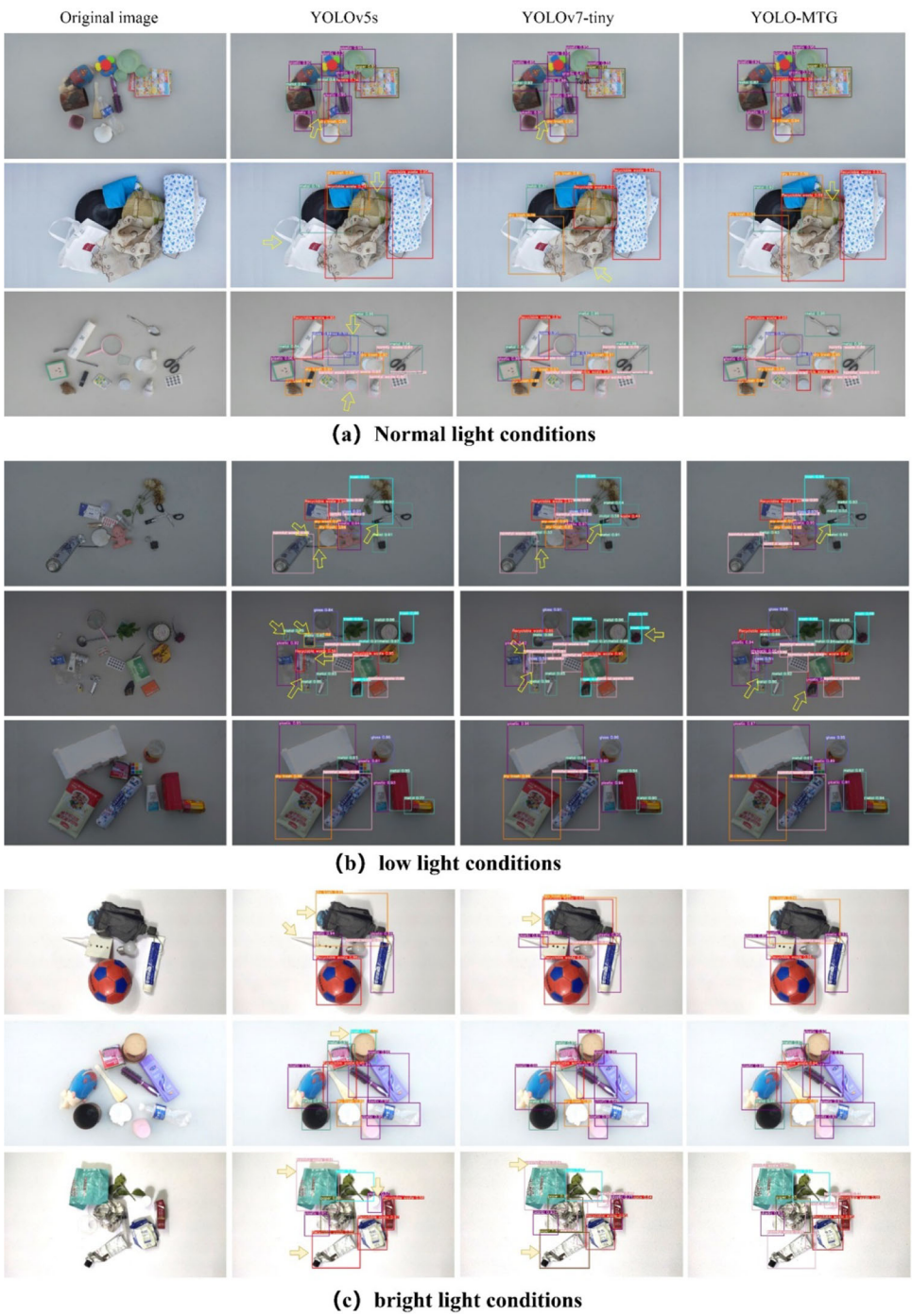

Fig. 11 The detection effect of **a** YOLOv5s, **b** YOLOv7-tiny and **c** YOLO-MTG under different light conditions

to call a camera for visual monitoring of garbage detection in the litter sorting environment. The system's operation interface is shown in Fig. 12, which provides an effective display of the detection process, including the states of litter images before and after detection. As can be seen, the system is able to recognize garbage rapidly and accurately in a manner meeting the needs of practical application.

4 Conclusion

This study put forwards a lightweight model called YOLO-MTG for multi-target garbage detection, which proves outperforming other advanced single-stage detection models. In this work, the feature extraction network MobileViTv3 is employed to encode the global features of multi-target garbage, so that the model proposed can refine the features and accurately locate the target garbage in high-density

Fig. 12 The operation interface of the garbage detection system

scenarios. The ED-Mobile block proposed on the basis of MobileViT block can smoothly process the global feature information of images using the EfficientFormer, while employing dynamic convolution ODConv to focus on learning the pixel features of different garbage targets. In this way, the discrimination and anti-occlusion abilities of the model are enhanced. By adopting feature reuse techniques, the proposed lightweight Thin-neck can maintain the expression and fusion abilities of the model on multi-scale objects at low computational costs. The experimental results show that the P value, R value and mAP value of YOLO-MTG on the MTG dataset reach 93.7%, 91.9% and 95.4%, with an increase of 1.3%, 1.2% and 1.2% over with the original YOLOv5s model, respectively. Moreover, the amount of model parameters is only 3.4 M, while the detection speed reaches 102FPS. Therefore, YOLO-MTG is easy to deploy on resource-constrained devices, and can locate and recognize multi-target garbage quickly and accurately in actual garbage detection scenarios.

It should be noted that there are two deficiencies in this study. For one thing, the proposed YOLO-MTG requires heavier computation than some lightweight detection networks (e.g., YOLOv7-tiny), so as to leads to the slower inference speed. Therefore, future work shall be focused on suitable compression algorithms for the model, so that the proposed model can get better applicability. Moreover, our application work is still in its early stages. For future application research, we will embed YOLO-MTG into certain edge devices (e.g., NVIDIA Jetson series and Raspberry Pi series), and redesign a comprehensive garbage detection system by combining hardware and software to simulate real garbage detection scenarios.

Acknowledgements The authors would like to thank the anonymous reviewers for their constructive comments and suggestions, which significantly contributed to improving the manuscript. This research was supported by Zhejiang Provincial Natural Science Foundation of China under Grant No. LY24F020005.

Author contributions Zhongyi Xia: Conceptualization, Methodology, Software, Formal analysis, Visualization, Writing – original draft. Houkui Zhou: Project administration, Data curation, Writing – review & editing. Huimin Yu: Data curation, Supervision, Haoji Hu: Investigation, Supervision. Guangqun Zhang: Investigation, Funding acquisition. Junguo Hu: Project administration, Funding acquisition. Tao He: Investigation.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Kuang, Y., Lin, B.: Public participation and city sustainability: evidence from Urban Garbage Classification in China. Sustain. Cities Soc. **67**, 102741 (2021). https://doi.org/10.1016/j.scs.2021.102741
2. Tong, Y., Liu, J., Liu, S.: China is implementing "Garbage Classification" action. Environ. Pollut. **259**, 113707 (2020). https://doi.org/10.1016/j.envpol.2019.113707
3. Mao, W.-L., Chen, W.-C., Wang, C.-T., Lin, Y.-H.: Recycling waste classification using optimized convolutional neural network. Resour. Conserv. Recycl. **164**, 105132 (2021). https://doi.org/10.1016/j.resconrec.2020.105132
4. Feng, Z., Yang, J., Chen, L., Chen, Z., Li, L.: An intelligent waste-sorting and recycling device based on improved EfficientNet. IJERPH. **19**, 15987 (2022). https://doi.org/10.3390/ijerph192315987
5. Chen, Z., Yang, J., Chen, L., Jiao, H.: Garbage classification system based on improved ShuffleNet v2. Resour. Conserv. Recycl. **178**, 106 (2022). https://doi.org/10.1016/j.resconrec.2021.106090
6. Li, N., Huang, H., Wang, X., Yuan, B., Liu, Y., Xu, S.: Detection of floating garbage on water surface based on PC-Net. Sustainability **14**, 11729 (2022). https://doi.org/10.3390/su141811729
7. Ma, W., Wang, X., Yu, J.: A lightweight feature fusion single shot multibox detector for garbage detection. IEEE Access. **8**, 188577–188586 (2020). https://doi.org/10.1109/ACCESS.2020.3031990

8. Jiang, X., Hu, H., Qin, Y., Hu, Y., Ding, R.: A real-time rural domestic garbage detection algorithm with an improved YOLOv5s network model. Sci. Rep. **12**, 16802 (2022). https://doi.org/10.1038/s41598-022-20983-1
9. Tian, M., Li, X., Kong, S., Wu, L., Yu, J.: A modified YOLOv4 detection method for a vision-based underwater garbage cleaning robot. Front Inform Technol Electron Eng. **23**, 1217–1228 (2022). https://doi.org/10.1631/FITEE.2100473
10. Luo, Q., Lin, Z., Yang, G., Zhao, X.: DEC: a deep-learning based edge-cloud orchestrated system for recyclable garbage detection. Concurr. Comput. Pract. Exper. (2021). https://doi.org/10.1002/cpe.6661
11. Cheng, X., Hu, F., Song, L., Zhu, J., Ming, Z., Wang, C., Yang, L., Ruan, Y.: A novel recyclable garbage detection system for waste-to-energy based on optimized centernet with feature fusion. J Sign Process Syst. **95**, 67–76 (2023). https://doi.org/10.1007/s11265-022-01811-1
12. Redmon, J., Farhadi, A.: YOLO9000: Better, Faster, Stronger. Presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
13. Redmon, J., Farhadi, A.: YOLOv3: An Incremental Improvement, http://arxiv.org/abs/1804.02767 (2018)
14. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
15. Glenn J., YOLOv5 release v6.0. https://github.com/ultralytics/yolov5/tree/v6.0 (2022)
16. Wadekar, S.N., Chaurasia, A.: Mobilevitv3: Mobile-friendly vision transformer with simple and effective fusion of local, global and input features. arXiv preprint arXiv:2209.15159 (2022)
17. Li, Y., Yuan, G., Wen, Y., Hu, J., Evangelidis, G., Tulyakov, S., Wang, Y., Ren, J.: Efficientformer: vision transformers at mobilenet speed. Adv. Neural. Inf. Process. Syst. **35**, 12934–12949 (2022)
18. Li, C., Zhou, A., Yao, A.: Omni-dimensional dynamic convolution. arXiv preprint arXiv:2209.07947 (2022)
19. Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., Xu, C.: Ghostnet: more features from cheap operations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1580–1589 (2020)
20. Chen, C., Guo, Z., Zeng, H., Xiong, P., Dong, J.: RepGhost: A Hardware-Efficient Ghost Module via Re-parameterization. arXiv preprint arXiv:2211.06088 (2022)
21. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929. (2020)
22. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)
23. Mehta, S., Rastegari, M.: Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. arXiv preprint arXiv:2110.02178 (2021)
24. Mehta, S., Rastegari, M.: Separable self-attention for mobile vision transformers. arXiv preprint arXiv:2206.02680 (2022)
25. Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J., Yan, S.: Metaformer is actually what you need for vision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10819–10829 (2022)
26. Yu, W., Si, C., Zhou, P., Luo, M., Zhou, Y., Feng, J., Yan, S., Wang, X.: Metaformer baselines for vision. arXiv preprint arXiv:2210.13452 (2022)
27. Yang, B., Bender, G., Le, Q.V., Ngiam, J.: Condconv: Conditionally parameterized convolutions for efficient inference. Adv. Neural Inf. Process. Syst. **32** (2019)
28. Chen, Y., Dai, X., Liu, M., Chen, D., Yuan, L., Liu, Z.: Dynamic convolution: Attention over convolution kernels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11030–11039 (2020)
29. Ma, N., Zhang, X., Zheng, H.-T., Sun, J.: Shufflenet v2: Practical Guidelines for Efficient Cnn Architecture Design. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 116–131 (2018)
30. Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V.: Searching for mobilenetv3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1314–1324 (2019)
31. Tang, Y., Han, K., Guo, J., Xu, C., Xu, C., Wang, Y.: GhostNetV2: Enhance Cheap Operation with Long-Range Attention. arXiv preprint arXiv:2211.12905 (2022)
32. Zhang, H., Hu, W., Wang, X.: Parc-net: position aware circular convolution with merits from convnets and transformer. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI. pp. 613–630. Springer (2022)
33. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and pattern recognition. pp. 7132–7141 (2018)
34. Woo, S., Park, J., Lee, J.-Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 3–19 (2018)
35. Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., Hu, Q.: ECA-Net: efficient channel attention for deep convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on cOmputer vision and Pattern Recognition. pp. 11534–11542 (2020)
36. Hou, Q., Zhou, D., Feng, J.: Coordinate attention for efficient mobile network design. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp 13713–13722 (2021)
37. Li, H., Li, J., Wei, H., Liu, Z., Zhan, Z., Ren, Q.: Slim-neck by GSConv: a better design paradigm of detector architectures for autonomous vehicles. arXiv preprint arXiv:2206.02424 (2022)
38. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021)
39. Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W.: YOLOv6: a single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976 (2022)
40. Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M.: YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696 (2022)
41. Fulton, M.S., Hong, J., Sattar, J.: Trash-ICRA19: A Bounding Box Labeled Dataset of Underwater Trash, http://conservancy.umn.edu/handle/11299/214366 (2020)
42. The PASCAL Visual Object Classes Challenge 2012 (VOC2012), http://host.robots.ox.ac.uk/pascal/VOC/voc2012