Heliyon 10 (2024) e24143

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

YOLO-SK: A lightweight multiscale object detection algorithm

Shihang Wang, Xiaoli Hao *

College of Computer Science and Technology (College of Big Data), Taiyuan University of Technology, Jinzhong 030600, China

ARTICLE INFO

Keywords:
Object detection
YOLOv5
Attention mechanism
Weighted feature fusion
Ghost convolution

ABSTRACT

YOLOv5 is an excellent object-detection model. However, it fails to fully use multiscale information when detecting objects with significant scale variations. It might use irrelevant contextual information, leading to incorrect predictions, particularly for low-performance devices. In this study, we selected lightweight YOLOv5s as the baseline model and proposed an improved model called YOLO-SK to overcome this limitation. YOLO-SK introduced several key improvements, the most important being the collaborative work of the weighted dense feature fusion network and SK attention prediction head. The proposed weighted dense feature fusion network could dynamically fuse features at different scales using autonomous learning parameters and cross-layer fusion capabilities. This enabled a balanced feature fusion ability in the output feature maps of different scales, thereby enhancing the richness of the effective information in the fused feature maps. The prediction head equipped with the SK attention mechanism broadened the scope of the model's receptive field and sharpened the focus on the target characteristics. This made it possible to glean more information about the target from the feature map output by employing a weighted dense feature fusion network. In addition, in order to improve the model's performance in terms of both accuracy and volume, we implemented the SIoU loss function and the Ghost Conv. The use of the model allowed for a more precise and in-depth comprehension of the event, which was made possible by all of these various methods of improvement. Extensive testing done on the PASCAL VOC 2007 and 2012 datasets showed that YOLO-SK was able to achieve considerable gains in prediction accuracy when compared with the baseline model (YOLOv5s), all while keeping the same level of model complexity. To be more specific, mAP@.5 increased by 2.6 %, and mAP@.5:.95 increased by 4.8 %. The advancements that were made and detailed in this paper could serve as a springboard for additional research that aims to improve the precision of multiscale object identification models for low-performance devices.

1. Introduction

Object detection is a crucial component of computer vision research that uses the computational capabilities of computers to simulate human vision, recognize object categories, and annotate their positions. In recent years, object detection algorithms using deep convolutional neural networks (CNN) have made significant progress, gradually replacing traditional object detection algorithms [1]. In addition, benchmark datasets such as MS COCO [2] and PASCAL VOC [3] have driven research on object detection technology.

The application scenarios of object-detection technology are diverse and include drone detection [4], autonomous driving [5], and factory product quality inspection [6]. Compared with humans, computers have a significant speed advantage in handling object

* Corresponding author.
E-mail address: hao.xiaoli880126@outlook.com (X. Hao).

https://doi.org/10.1016/j.heliyon.2024.e24143
Received 8 June 2023; Received in revised form 7 December 2023; Accepted 4 January 2024
Available online 7 January 2024

detection tasks, greatly enhancing productivity. However, owing to the variety of application scenarios and differences in the computational device performance for executing detection tasks, there are significant variations in the scale of the same type of object in different scenes, which limits the performance of the detection models. This study's objective is to investigate general methods for enhancing the performance of multiscale object detection algorithms in scenarios with high real-time requirements but limited computing power.

The YOLO series plays a crucial role in single-stage object detection. In this study, we propose an improved model called YOLO-SK based on the lightweight model YOLOv5s (You Only Look Once Version 5) [7]. The objective was to address the issue of accuracy in multiscale object detection for lightweight models and provide new insights for similar research efforts.

This study focused on enhancing the richness of effective information in feature maps and using these features effectively to extract the target information. To achieve this, a weighted dense feature fusion network and detection head combined with the SK attention mechanism are proposed. This study's objectives were to improve the model's detection accuracy and to control the model complexity, making it suitable for running on devices with weak performance. The specific contributions of this study are as follows.

(1) A new weighted dense feature fusion network (WD_FPN-PAN) is proposed. This structure has autonomously learnable weight parameters, could dynamically fuse feature information at different levels, minimizes the introduction of redundant information, and maximizes the enhancement of valid information in fused feature maps.
(2) By integrating the selective kernel attention mechanism into the prediction head (SK_PH), its multibranch convolution structure provides a larger receptive field to the detection head, thereby enhancing the detection head's attention to the target information. This could improve the model's attention towards valid information in the fused feature maps without increasing the number of detection heads.
(3) A more comprehensive judgment standard, the SIoU loss function, was introduced to reduce the positioning error between the prediction and real boxes, ensuring that targets of different scales could be located more accurately.
(4) From the perspective of generating redundant features, GhostConv replaces the original Conv. With its structural characteristics, it reduces the learning cost of non-key features. Thus, reducing the overall computation of the model, keeping the model lightweight and reliable.

YOLO-SK is a lightweight object identification technique, which means that in order to reduce the size of the model, it has lost some accuracy in order to get a smaller overall size. As a consequence of this, the accuracy of the YOLO-SK algorithm could not be on par with that of other object identification algorithms, such as the Faster R–CNN or the Mask R–CNN.

Because it is a multiscale object identification method, YOLO-SK is able to identify a wide variety of different sized items. However, in comparison to other object identification algorithms, such as RetinaNet and EfficientNet, YOLO-SK does not scale up nearly as well. Because of this, it is possible that YOLO-SK will not be able to detect items as accurately in situations that contain a high number of objects in a complicated arrangement.

Because it is a deep learning method, YOLO-SK is extremely sensitive to any noise that may be present in the input image. As a direct consequence of this, YOLO-SK might not be able to recognize objects as precisely in images that contain a lot of noise.

2. Related work

Deep learning-based object detection algorithms [7–10] have shown excellent performance and have been widely applied. Each coin has two sides. In addition, these algorithms have two sides and significant limitations. Complex network structures could achieve higher detection accuracy, but at the cost of slower inference speed and increased reliance on computational resources. However, lightweight models often offer faster inference speeds but struggle to achieve high accuracy. In this section, we provide a comprehensive review of the relevant literature from two perspectives: improving detection accuracy and reducing model complexity. Inspired by these studies, we propose improvements in this study.

Various methods have been proposed to improve the accuracy of object-detection algorithms for multiscale targets. Zhao et al. [11] combined transformer and CNN structures in the backbone network of a model to better use the global and local information of the image for feature extraction. The outputs of the two structures were adaptively fused to enhance the detection ability of the model for small targets. Chen et al. [12] designed a channel-spatial attention mechanism for the FPN (Feature Pyramid Network) structure in Faster R–CNN, which reduced the background noise introduced in the feature fusion process from both the channel and spatial dimensions. This mechanism retained additional key features and improved the detection accuracy of the model for small targets. Gong et al. [13] deepened the original YOLOV3-tiny network structure by adding a series of 3×3 and 1×1 convolutional layer. This enabled the model to better extract the features of vehicles from thermal images and accurately detect multiscale targets. Yu et al. [14] proposed dilated convolutions to integrate the contextual information. The size of the receptive field was controlled by adjusting the dilation rate to adapt to object-detection tasks at different scales. To increase the detection accuracy of the model, Tan et al. [15] proposed a weighted bidirectional feature pyramid network (BiFPN) that fused more target features with effective bidirectional cross-scale connections and weighted feature fusion. The interpretation of MRI brain images, which can include the detection of brain tumors, is a difficult undertaking. Multimodal medical image processing has garnered increased attention in recent years, and MRI scans themselves are multimodal. It has been suggested that the information transfer across and among modalities could be used to overcome this difficulty in MRI brain picture segmentation [26]. In recent years, there has been a rise in interest in methods that are based on deep learning and can predict drug-target interactions (DTI). In drug target interaction, the data relating to drugs and targets might come in a variety of modalities; as a result, researchers are forced to use multimodal techniques. It has been demonstrated that

the most important factor in multimodal DTI prediction is the presence of a discriminative feature representation of the drug-target pair [27].

These studies inspired the present study. In this study, a cross-layer feature fusion method was employed to combine shallow positional and deep semantic features across different layers. The introduction of redundant information is reduced by using learnable weight parameters, resulting in the fusion of feature maps that provide richer information to the prediction head. In addition, an attention mechanism was incorporated into the prediction head to enhance its scale awareness and improve its focus on target information. This allowed for the extraction of more effective target information from the enhanced fusion feature maps.

Extensive studies conducted in this area have reduced the size and complexity of the model and improved the inference speed. Howard et al. [17] suggested depth wise separable convolution as a replacement for conventional convolution. Consequently, the computational and parameter complexities of the model were reduced. The Ghost convolution method was suggested by Han et al. [18], which generated a portion of the feature maps via conventional convolution before enriching the feature maps using low-cost linear operations. Consequently, the computational complexity and parameter count of the model were considerably reduced. Li et al. [19] and Li et al. [20] used model compression techniques based on channel pruning to create lightweight networks. The number of model parameters were reduced by pruning the less important channels according to certain rules. However, this method could not guarantee the stable detection accuracy of the model.

This study used a more efficient Ghost convolution to replace the conventional convolution to optimize the model structure. This made the model lightweight while maintaining detection accuracy.

3. Methods

In this study, a lightweight multiscale object detection algorithm, YOLO-SK, was proposed based on the YOLOv5s model. Fig. 1 illustrates the architecture. After an image was preprocessed and input into the model, it first passed through the backbone network, where its features were extracted, forming feature maps of multiple levels and scales. Then, these feature maps were sent to the feature fusion network (neck) for feature fusion. Using a weighted dense feature fusion network, deep semantic features were fused with shallow positional features, enhancing the expressive power of the output features. The fused feature maps were processed by the detection layer, with three detection heads of different sizes performing detection and outputting the prediction boxes and category information of the targets.

The principles and motivations behind network optimization were elaborated in the following sections. The focus was on how SK_PH worked in tandem with WD_FPN-PAN to enhance the detection accuracy of multiscale targets. In addition, we discuss how the SIoU loss function improved the accuracy of multiscale target prediction boxes. Finally, the compression effect of GhostConv on the model is presented.

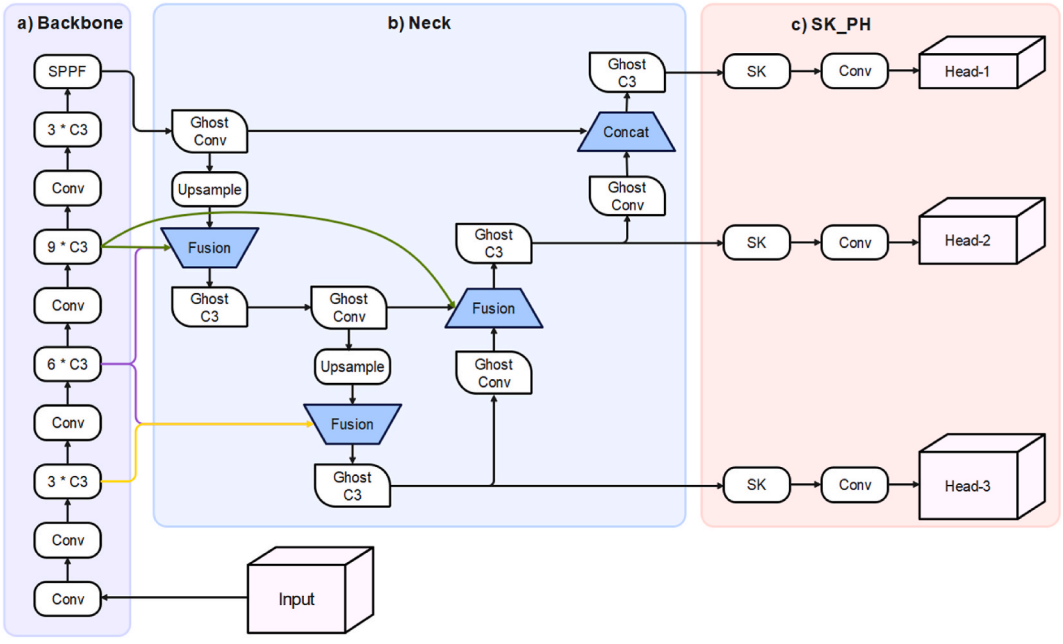


Fig. 1. Structure of YOLO-SK. a) CSPDarknet53 is used as the backbone. b) The neck uses the WD_FPN-PAN structure introduced in GhostConv. c) Three SK_PHs use feature maps from the neck.

The step wise procedure for the proposed algorithm is as follows:

 # Input: Image
 # Output: Bounding boxes of detected objects
Step 1: The image is preprocessed by being resized and having its pixel values normalized.
Step 2: Using a convolutional neural network (CNN), extract features from the image. For the lightweight_head(features parameter, concatenate the features that were extracted from the various levels of the CNN.
Step 3: Apply a layer that is entirely related to the concatenated features.
Step 4: A sigmoid activation function is applied to the output of the fully connected layer, and the output of the sigmoid activation function is then returned.
Step 5: Use a lightweight head to make predictions about the bounding boxes and classes of items in the image.
Step 6: Eliminate Predictions in Which You Have Low Confidence
Step 7: Give back the bounding boxes that correspond to the objects that have been detected.

Because of YOLO-SK's lightweight head, the algorithm is able to achieve great accuracy while yet keeping a compact footprint. This makes the lightweight head an essential component. The pixel values of the input image are first brought to a consistent level, and then the image is shrunk down to a specific size that has been determined in advance. This approach is carried out in order to make it simpler for the Convolutional Neural Network to extract relevant attributes from the image being processed. In CNN, the process of extracting features from an image involves applying a series of convolutional and pooling layers in sequence. It is the responsibility of a neural network's convolutional layers to teach the network how to recognize and differentiate between various spatial patterns included within an image. The pooling layers, on the other hand, have the function of reducing the size of the feature maps and, as a result, increasing their resistance to noise. The lightweight module makes predictions about the bounding boxes and classes of items present in the image by using the attributes that were collected by the Convolutional Neural Network (CNN). The term "lightweight head" refers to a condensed neural network architecture that has been meticulously developed to achieve the highest possible levels of both accuracy and productivity. In order to get rid of predictions that have a low level of confidence, the approach uses a threshold that is applied to the output of the lightweight head. This measure contributes to the reduction of the number of incorrectly positive detections that take place.

3.1. WD_FPN-PAN

In a CNN, each feature map contains distinct target feature information. The feature maps obtained from the shallow layers of the network had a higher resolution and captured the target's position information precisely, thereby aiding the network in accurately regressing the target boundary. In contrast, feature maps obtained from the deep layers of the network had a lower resolution and focused more on extracting advanced semantic information relevant to the target, which aided the network in accurately detecting the target. The original YOLOv5 model's feature fusion network uses a combination of a top-down feature pyramid network (FPN) [22] and bottom-up path aggregation network (PAN) [23] to merge the extracted semantic and position features.

Multiple convolution operations leave feature maps with semantic information but without the precise location features of the targets, the model's detection accuracy was limited. This study proposed a weighted and dense feature pyramid network-path aggregation network (WD_FPN-PAN) for feature fusion to further improve the model's attention toward shallow information. The WD_FPN-PAN structure, depicted in Fig. 2, incorporated low-level feature maps C1 and C3 as input feature maps to the FPN fusion node and added the same level feature map C3 as the input feature map to the PAN fusion node to fully fuse the detailed position information of the targets.

In the feature fusion process, different input feature maps often contribute to a fused feature map with varying degrees of importance. However, the improved fusion node had three input edges with different feature information foci, which could introduce

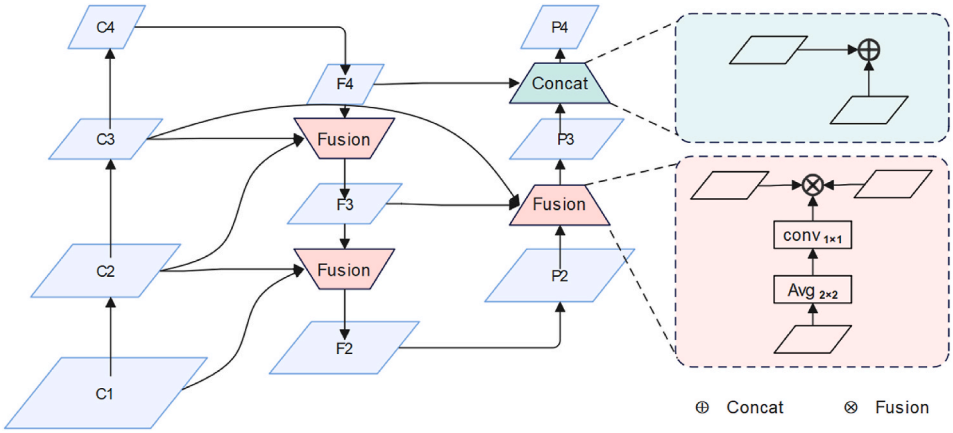


Fig. 2. Structure of WD_FPN-PAN (weighted and dense_feature pyramid network-path aggregation network). Note: WD_FPN-PAN uses the feature maps of the three backbones as additional input feature maps for the neck fusion nodes. The quadrilateral size represents the feature-map size.

redundant information if direct fusion was applied. To overcome this problem, this study proposed a weighted fusion method that employed learnable parameters w to determine the relative importance of each input feature branch. With this method, the accuracy of the fused feature map increased, and the integration of false feature information decreased. Eq. (1) shows the calculation formula for the weighted fusion method, which describes how learnable parameters were used to calculate the weighted sum of the input feature maps.

$$P_{out} = \sum_i \frac{w_j \times P_{in}}{\varepsilon + \sum_j w_j} \tag{1}$$

In Eq. (1), each input feature map that had to be fused was represented by P_{in}, and the weight coefficient for each input feature map was represented by w_j. Weight coefficient w_j represented the weight assigned to the input branch feature map and could be updated through self-learning. The initial value was set to one, indicating the direct fusion of different branch feature maps. In this study, we proposed normalizing the weight coefficient to a range of zero to one. This approach enhanced the training speed and mitigated the occurrence of training instability. The small value ε, fixed at 0.001, was used to prevent numerical instability. By adjusting the weight coefficients of each input feature map, the weighted fusion method assigned different levels of importance to the different feature maps, reducing the impact of redundant or invalid information and improving the overall accuracy of the fused feature map.

The input feature maps of the top-level nodes in the PAN structure underwent fewer convolutions before fusion and retained detailed information. To decrease the model's complexity, feature fusion of nodes with only two input edges was performed using the Concat operation, which concatenated feature maps along the channel dimension. The weighted fusion method was used for other fusion nodes with three input edges, and the weight parameters were updated during model training through adaptive learning. This approach ensured that the fused features contain crucial information and improved the accuracy of model detection.

3.2. SK_PH

The detection head of the YOLOv5 network was responsible for performing both classification and localization tasks. To detect targets of different sizes, the network employed three detection heads that use feature maps downsampled from 8×, 16×, and 32× of the input image. These detection heads identified the targets in the original image based on their predicted bounding boxes and classifications. However, it was recommended that SK attention be added to the original detection head to further increase detection accuracy. This enhanced the model's ability to perceive scales and consequently improve its detection accuracy.

The core concept behind SK attention [21] was the use of convolution kernels of various sizes to extract feature information from an input image, predict the significance of various feature maps for the detection task, and select the feature map data from various receptive fields based on channel attention weights. This approach aimed to highlight informative features and suppress irrelevant features to adapt them to targets at different scales. Fig. 3 shows the split, fused, and selected components of the SK attention structure. The split component uses multi-branch convolution to extract the features of different receptive fields using 3×3 and 5×5 convolution kernels, resulting in feature maps U1 and U2. In this study, a large 5×5 convolution kernel was replaced with a dilated convolution [14], which used a dilation rate of two and a kernel size of 3×3, thus reducing the number of module parameters while retaining the same receptive field size. The fuse component first performed element-wise addition on feature maps U1 and U2 to create the fused feature map U. Next, the feature map U was subjected to global average pooling to produce the feature vector S, which provided the global information of each channel in the feature map. Finally, to acquire the channel weights Z, a fully connected layer was employed to aggregate and enlarge the channels. The selected component aggregated the feature maps of different kernel sizes based on the weights of the different channels. Specifically, feature maps U1 and U2 were multiplied by the channel weights and then added element-wise to strengthen the key target features and reduce irrelevant features.

The split component of the SK attention module used two convolution operations to expand the perceptual field, which led to an increase in computation and memory overhead, while simultaneously increasing the detection accuracy of the model. In a CNN model, the detection head is situated at the end of the network, and the resolution of the feature map decreases as it approaches the end of the model. Therefore, incorporating SK attention into the detection head could enhance the model's ability to perceive multiple scales while minimizing the computational resources required.

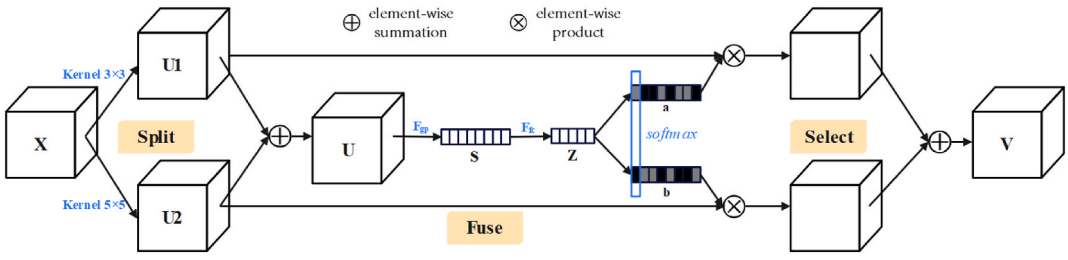


Fig. 3. Illustration of SK attentional structure consisting of three parts: split, fuse, and select.

3.3. SIoU loss

The localization, confidence, and class losses were three separate parts of the YOLOv5 loss function defined in Eq. (2).

$$Loss_{object} = Loss_{loc} + Loss_{conf} + Loss_{cls} \tag{2}$$

To calculate the localization loss of the target box, YOLOv5 used the GIOU [24] loss function, which considered the intersection over union and overlapping area between the ground truth box G and the predicted box P. This approach provided an accurate measurement of the direction of the mismatch between the predicted and ground-truth boxes. Eq. (3) presents the corresponding calculation formula, where C represents the minimum bounding rectangle of the predicted and ground-truth boxes.

$$Loss_{GIoU} = 1 - IoU + \frac{C - (G \cup P)}{C} \tag{3}$$

However, the GIOU loss function did not consider the possibility of a directional mismatch between the expected ground-truth box and the predicted box. During the training process, the predicted box might drift randomly, leading to slow convergence and ultimately producing a suboptimal model.

In this study, localization loss was computed using the SIoU [16] loss function. In contrast to the GIOU loss function, SIoU considered the orientation angle between the ground-truth box and the predicted box. This approach results in a more stable target-box regression process and higher convergence accuracy. The SIoU loss function, which comprised four functions (angle, distance, shape, and IoU cost), redefined the penalty metric. Eq. (4) shows the formula.

$$Loss_{SIoU} = 1 - IoU + \frac{\Delta + \Omega}{2} \tag{4}$$

By incorporating the angle cost Λ, the SIoU loss function encouraged the predicted box to align with the angle of the ground truth box, resulting in more accurate and stable target box regression. The distance cost Δ penalized the mismatch between the predicted and ground truth boxes in terms of their distances. The shape cost Ω penalized the difference between the predicted and ground truth boxes in terms of their aspect ratios and areas.

Eq. (5) shows the calculation formula for the orientation cost in the SIoU loss function, where $x = \sin\alpha$ and α represented the angle between the line connecting the centers of the boxes and the X-axis. Specifically, it measured the difference between the sine of angle α and its optimal value, which was either zero or one, depending on whether the two boxes had the same or opposite orientation.

$$\Lambda = 1 - sin^2\left(\arcsin(x) - \frac{\pi}{4}\right) \tag{5}$$

Eq. (6) shows the calculation formula for the distance cost in the SIoU loss function, where $\gamma = 2 - \Lambda$.

$$\Delta = \sum\nolimits_{t=x,y} (1 - e^{-\gamma\rho_t}) \tag{6}$$

Eq. (7) illustrates the calculation formula for the shape cost.

$$\Omega = \sum_{t=w,h} (1 - e^{-\omega_t})^{\theta} \tag{7}$$

3.4. Model compression

As can be seen in Fig. 4, the YOLOv5 model has a composite convolutional layer for its convolutional structure. It was composed of a typical convolution layer, a batch normalization (BN) layer, and a SiLU activation function, and it served as the essential building block upon which other structures were constructed. However, in conventional convolution, there are typically excessively large numbers of convolutional kernels and channels, making it difficult to extract features in a complete manner and leading to significant computing costs.

In order to construct feature maps based on the similarities between certain feature maps, the ghost convolution approach was suggested by Han et al. [18]. This action was taken because it was necessary to solve the posed issue. This strategy builds feature maps via the time-tested convolutional method, enhances them via fewer parameters, and decreases the processing load with linear operations that are computationally cheap. Ghost convolution uses shared similarities between feature maps to reduce the amount of convolutional kernels and channels required for feature extraction. This significantly reduces the computing burden without sacrificing model fidelity.

The procedure of conventional convolution is shown in Fig. 5(a) and Ghost convolution is depicted in Fig. 5(b). Ghost convolution begins by employing traditional convolution in order to build intrinsic feature maps with a reduced number of channels. Next, it

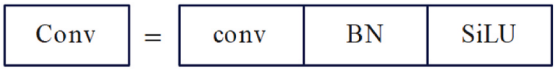


Fig. 4. Illustration of the structural components of Conv in YOLOv5s.

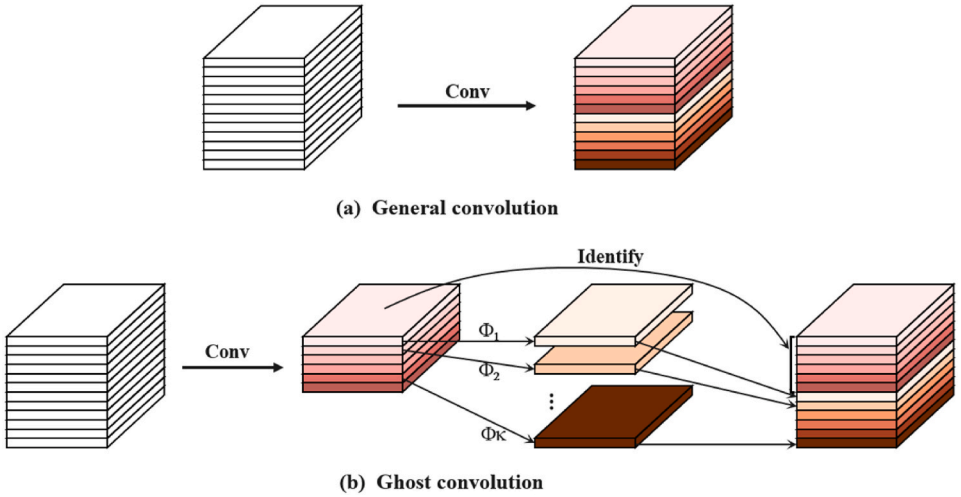


Fig. 5. Demonstration of the general convolutional and the Ghost convolutional for producing the same quantity of feature maps (Φ stands for the inexpensive operation) (a) Convolution operation (b) Ghost convolution.

employs a more cost-effective linear operation in order to construct Ghost feature maps. The linear operation Φ operated on each channel, resulting in lower computational cost compared to conventional convolution. Finally, the final output was created by concatenating the intrinsic feature maps with the ghost feature maps.

Compared with conventional convolution, Ghost convolution used fewer parameters and had a simpler computational structure while retaining the same output feature map size. By leveraging the feature map similarities and reducing the number of required convolutional kernels and channels, Ghost convolution enables efficient feature extraction and enhanced the model's overall performance.

Although Ghost convolution had the advantage of reducing the computational complexity and parameter costs of convolutional layers, it also required segmentation and concatenation operations on input feature maps, which could increase memory usage during model training and impact the model's overall performance.

Consequently, this study substituted all conventional convolutions in the feature fusion network and left only the conventional convolution in the backbone network. This approach solved the problem of increased computational and parameter costs caused by the SK_PH detection head and WD_FPN-PAN structure without sacrificing the model's overall performance. By selectively applying Ghost convolution to certain layers, the model achieved a balance between computational efficiency and accuracy, resulting in an efficient and effective object-detection system.

4. Experiments

This section demonstrates the effectiveness of the proposed improvements from an experimental perspective and highlights the real improvement results. The experiments comprised three parts: a comparative experiment with other detection models, an ablation experiment for each improvement point, and an analysis of the model's overall performance. Experiments were conducted following the concept of controlled variables to ensure rigor and accuracy. The experimental platform and dataset used in each group of experiments and the model parameters were consistent. The data displayed were the results obtained by conducting experiments with reference to the relevant literature. The experiments were conducted on a computer running Ubuntu 20.04, equipped with 32 GB RAM, Intel i7 9700F CPU, and an Nvidia RTX 3080Ti GPU. The YOLO-SK model was implemented using Python and trained and tested using the PyTorch 1.11.0 framework. During the experiments, the initial learning rate of all models was set to 0.01, the final learning rate to 0.2, and the momentum parameter to 0.937. A total of 300 training epochs were used.

4.1. Experimental dataset

The datasets used in the experiment were PASCAL VOC 2007 and PASCAL VOC 2012 [3]. These datasets are widely used worldwide in the field of computer vision and aim to provide standard benchmark test sets to promote the development and comparison of computer vision algorithms. The datasets contained 20 common object categories, including people, dogs, chairs, and cars, with approximately 100 to 2000 annotated samples per category. For the experiment, the training and validation parts of both datasets were merged to form a training set consisting of 16,551 images. The test set consisted of 4952 images from the test part of VOC 2007. All data formats were converted according to the YOLO requirements.

4.2. Evaluation metrics

In this study, mAP@.5, mAP@.5:.95, computational cost (FLOPs), and parameter count (Params) were used as the evaluation metrics for the object detection models. Two mAPs [2] at different thresholds were used to measure the model's accuracy, where mAP@0.5 represented the average AP at an IoU threshold of 0.5, and mAP@0.5:0.95 represented the average of all mAPs at IoU thresholds ranging from 0.5 to 0.95, with a step size of 0.05. The computational cost and parameter count were used to measure the complexity of the model from temporal and spatial perspectives, respectively. The computational cost referred to the number of floating-point operations required by the model, whereas the parameter count referred to the number of parameters that must be trained by the model.

4.3. Contrast experiments and analysis

To validate the accuracy and complexity of YOLO-SK, contrast experiments were conducted with baseline models, YOLO series lightweight models, and YOLO series complex models, while ensuring consistent experimental platforms, datasets, and model parameters. Table 1 lists the experimental data.

4.3.1. Comparison with the baseline model

From the perspective of model complexity, YOLO-SK and YOLOv5s exhibited similar GFLOPs indicators. YOLO-SK had fewer parameters than the baseline model, indicating comparable model complexity. However, YOLO-SK outperformed the baseline model in terms of performance, with a 2.6 % increase in mAP@.5 and a 4.8 % increase in mAP@.5:.95. Although the addition of the SK attention mechanism to the detection head and weighted dense feature fusion network improved the detection accuracy of the model, the complexity remained unchanged. This indicated that Ghost Conv was an effective improvement and played a crucial role in model compression.

4.3.2. Comparison with YOLO lightweight models

YOLO-SK exhibited similar complexity to YOLOv4-tiny and YOLOv7-tiny. However, YOLO-SK achieved the highest precision, with improvements of 19.9 % and 1.3 % in mAP@.5, compared to YOLOv4-tiny and YOLOv7-tiny, respectively. In addition, mAP@.5:.95, increased by 26.5 % and 2.7 % compared to YOLOv4-tiny and YOLOv7-tiny, respectively.

4.3.3. Comparison with YOLO complex models

YOLO-SK had a similar precision to YOLOv4 and YOLOv5m. mAP@.5 was comparable to that of YOLOv4, but 2.4 % lower than that of YOLOv5m. The mAP@.5:.95 was 2.5 % lower than that of YOLOv4 and YOLOv5m by 2.3 %. YOLO-SK had significant advantages in terms of complexity. Compared with YOLOv5m, the computational complexity and number of parameters were reduced by 66.7 % and 67.0 %, respectively. Significant reductions, in comparison to YOLOv4, were seen in both the computational complexity and the number of parameters. These changes resulted in a decrease of 86.6 % and 87.0 %, respectively.

In conclusion, the weighted dense feature fusion network that was proposed and the SK attention detection head were both contributors to the improvement in the detection accuracy of the lightweight models. Through the utilization of Ghost Conv, model compression was made possible, which helped the model to keep its low weight.

4.4. Ablation experiments and analysis

In order to give more data on the effectiveness of the suggested improvements, five ablation experiments were carried out on YOLOv5s utilizing the exact same hyperparameters as those that were utilized during training. The findings are presented in Table 2, where the presence of each improvement is denoted by "$\surd$".

In the second trial, the findings showed that switching to the SK_PH detection head from the original detection head resulted in a 2.1 % improvement in mAP@.5 and a 4.3 % improvement in mAP@.5:.95. These results demonstrated that this module had the most impact on enhancing the model accuracy. However, this led to an increase in the complexity of the model, which in turn demanded an increase in the amount of resources available on the hardware. In the third trial, the incorporation of the weighted dense feature fusion module WD_FPN-PAN based on the utilization of the SK_PH detecting head enhanced mAP@.5 by 0.7 % and mAP@.5:.95 by 0.6 %. These results indicated that weighted learning was advantageous for the process of fusing distinct input features. This module did not

Table 1
Results of contrast experiments.

Model	mAP@.5(%)	mAP@.5:.95(%)	FLOPs(G)	Params(M)
YOLOv5s [7]	76.5	50.2	16	7.1
YOLOv5m [7]	**81.5**	**57.5**	48.3	20.9
YOLOv4 [9]	79.0	57.3	120	52.9
YOLOv4-tiny [25]	59.2	28.5	16.2	**5.9**
YOLOv7-tiny [10]	72.7	47.5	**13.3**	6.1
YOLO-SK	79.1	55.0	16.1	6.9

Table 2
Results of ablation experiments.

YOLOv5s	SK_PH	WD_FPN-PAN	SIoU	GhostConv	mAP@.5 (%)	mAP@.5:.95 (%)	FLOPs (G)	Params (M)
✓	–	–	–	–	76.5	50.2	16.0	7.1
✓	✓	–	–	–	78.6	54.5	69.0	36.1
✓	✓	✓	–	–	79.3	55.1	69.9	36.3
✓	✓	✓	✓	–	79.6	55.3	69.9	36.3
✓	✓	✓	✓	✓	79.1	55.0	16.1	6.9

result in an increase in either the computational cost or the number of parameters, despite the fact that it did not help to improve the accuracy of the model. In the fourth experiment, the introduction of the SIoU loss function to calculate the localization error based on the third experiment improved mAP@.5 by 0.3 % and mAP@.5:.95 by 0.2 %, making the model more accurate in regressing the target boundaries. Finally, in the final experiment, using GhostConv to replace conventional convolutions in the feature fusion network of the improved model reduced the computational cost by 80.0 % and the parameter count by 81.0 %, which was an effective method for lightweighting the model, although the detection accuracy decreased.

In summary, the introduction of SK_PH, WD_FPN-PAN, and the SIoU loss function could improve the detection accuracy of the model, whereas GhostConv could effectively reduce the computational cost and parameter count. The improved model had a complexity similar to that of YOLOv5s, and mAP@.5 and mAP@.5:.95, improved by 2.6 % and 4.8 %, respectively. The model's ability to classify and locate targets was further strengthened, and it required fewer hardware resources.

4.5. Comprehensive model performance validation

4.5.1. Classification accuracy

To assess the classification accuracy of the YOLO-SK model for multiscale targets, the AP@.5:.95 and AR@.5:.95 of all small, medium, and large targets in the test set were calculated. Small targets were defined as those with a resolution of less than 32×32 pixels, large targets were defined as those with a resolution of $>96 \times 96$ pixels, and medium targets were defined as the remaining targets based on the absolute pixel size of the target.

The results in Table 3 show that the YOLO-SK model had a higher average precision and average recall for detecting targets of different scales than the YOLOv5s model. Specifically, the AP@.5:.95 values of YOLO-SK for the small, medium, and large targets were 20.1 %, 42.2 %, and 65.1 %, respectively, which were 1.1 %, 3 %, and 5.5 % higher than those of YOLOv5s. Moreover, the YOLO-SK model had AR@.5:.95 values for small, medium, and large targets that were 2.4 %, 3.1 %, and 4.2 % higher, respectively, than those of YOLOv5s.

In summary, the proposed improvements could improve the classification accuracy of the model for multiscale targets and address the issues of false positives and negatives. The YOLO-SK model outperformed the YOLOv5s model in detecting small, medium, and large targets, thereby demonstrating the effectiveness and versatility of the proposed method.

4.5.2. Convergence performance

A high accuracy and fast convergence are important for the robustness and stability of a model in practical applications. Therefore, the YOLOv5s and YOLO-SK models were trained and tested for 300 epochs, and their convergence curves were compared. Fig. 6 shows the convergence curves of the localization (box_loss) as shown in Fig. 6(a), confidence (obj_loss) as shown in Fig. 6(b), and classification losses (cls_loss) as shown in Fig. 6(c) for YOLOv5s and YOLO-SK. The vertical axis represented the various loss values during network training, and the horizontal axis represented the iteration rounds of the network.

The experimental results revealed that the loss values of both models were high at the beginning of the training process. Within the first 50 epochs, the loss values of both models exhibited decreasing and converging trends. During the training process, the loss values of the network continued to decrease as the number of iterations increased, indicating that the network fit the training data. Overall, the YOLO-SK model always maintained the loss values at a lower level while ensuring convergence speed, and its convergence performance was better, with better robustness and stability.

In summary, the experimental results confirmed the effectiveness of the proposed improvements in the YOLO-SK model, which enhanced its detection accuracy, particularly for multiscale targets, while maintaining similar computational costs and parameter values. In addition, the model had a better convergence performance, making it more robust and stable for practical applications.

Table 3
Detection results for multiscale objects.

Model	AP@.5:.95(%)			AR@.5:.95(%)		
	small	medium	large	small	medium	large
YOLOv5s	19.0	39.2	59.6	35.2	58.3	73.5
YOLO-SK	**20.1**	**42.2**	**65.1**	**37.6**	**61.4**	**77.7**

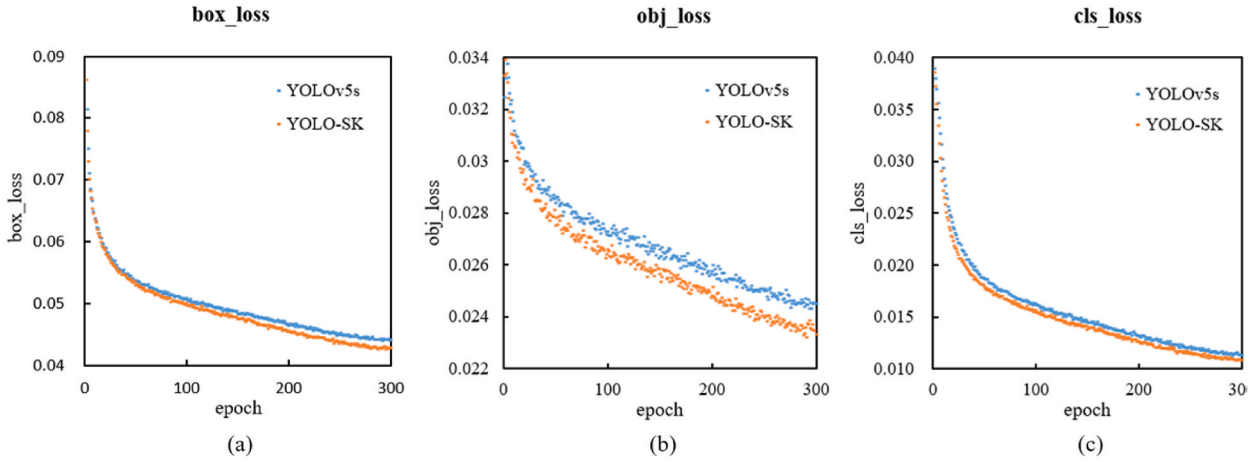


Fig. 6. Convergence curves for YOLOv5s and YOLO-SK (a) Box loss (box_loss) (b) Confidence loss (obj_loss) (c) Classification loss (cls_loss).

4.5.3. Detection effect

We chose three groups of photographs to validate the detection effect in order to evaluate the performance of the YOLOv5s and YOLO-SK algorithms on the VOC 2007 test set images as shown in Fig. 7(a) and (b) respectively. These images were taken under a variety of conditions.

The object scale was on the smaller side in the initial set of photos. YOLOv5s had a significant number of false positives, whereas YOLO-SK accurately detected a greater number of targets and had a greater sensitivity to objects of smaller sizes. This indicated that the suggested enhancements to YOLO-SK increased the model's capacity to recognize low-density targets. The targets in the second batch of experimental photographs were dense, with large-scale changes and severe occlusions in their appearance. YOLO-SK shown superior detection abilities when compared to YOLOv5s, detected a greater number of targets, and raised the confidence of the predicted boxes to varying degrees. This revealed that the proposed enhancements could potentially enhance the model's capacity to deal with difficult scenarios that involve dense targets and occlusions. Although YOLOv5s did not miss any detections in the third set of testing, there were some false-positive boxes, but YOLO-SK detected the categories accurately and improved detection accuracy. The findings of the experiments demonstrated that the proposed modifications provided an efficient solution to the issue of erroneous detections.

In conclusion, the proposed adjustments effectively improved the detection performance of the YOLO-SK model, and they successfully minimized the issues of missing and erroneous detections. The usefulness and applicability of the proposed changes was demonstrated by the fact that the YOLO-SK algorithm exhibited improved detection ability, stronger sensitivity to small objects, and better performance in complicated circumstances.

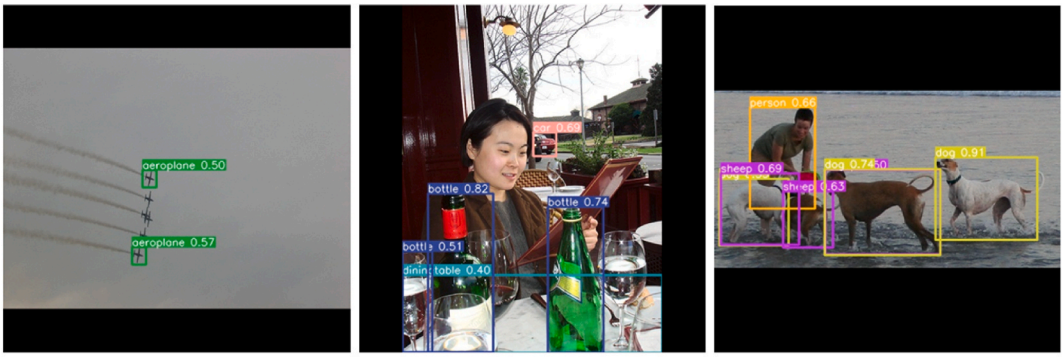
5. Conclusion

To enhance the precision with which multiscale objects may be detected, the authors of this study presented the YOLO-SK model, which involved modifying the head and feature fusion network of YOLOv5 with novel methods. Cross-level fusing of shallow and deep feature maps, making use of both granular and semantic data, was made possible by the incorporation of a weighted feature fusion network. The model's ability to zero in on the desired characteristics was improved by the detecting head's SK attention mechanism, and data was gleaned from the combined feature maps. The model parameters and complexity were unaltered from the baseline model (YOLOv5s) after the conventional convolutions were swapped out for ghost convolutions. A 2.6 % rise in mAP@.5 and a 4.8 % increase in mAP@.5:.95 indicate that the detection accuracy has improved. This is a promising step toward bettering lightweight models' multiscale item identification accuracy. This study has the potential to improve the performance of low-powered devices, such drones and industrial quality inspection instruments, by allowing them to detect multi-scale objects with more precision.

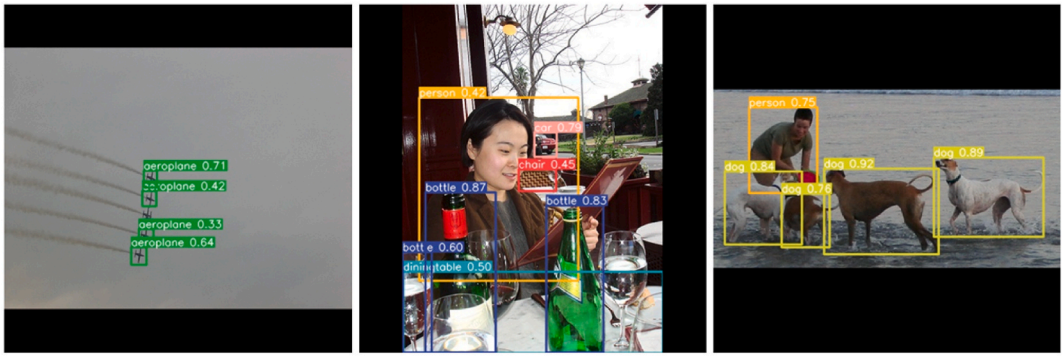
This work's findings can be applied to situations where limited-performance devices are used for multi-scale object detection. As a result, it is important to keep the model's complexity under wraps while also enhancing its detection accuracy. Future work could expand on this study by developing more effective model compression algorithms that safeguard and improve target information while it is compressed. This line of inquiry may open up further opportunities for increasing detection accuracy while decreasing complexity, paving the way for more widespread use of object detection algorithms across a broader range of devices and boosting productivity at a reduced cost.

Data availability statement

Data associated with this study has been deposited at https://github.com/shihanghoney97/YOLO-SK.



(a) YOLOv5s



(b) YOLO-SK

Fig. 7. Visualization results on the test set with bounding boxes of varying colors for each category (a) Using YOLOv5s (b) Using YOLO-SK. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

CRediT authorship contribution statement

Shihang Wang: Writing - review & editing, Writing – original draft, Visualization, Validation, Methodology, Funding acquisition, Formal analysis, Data curation. **Xiaoli Hao:** Writing – review & editing, Validation, Resources, Project administration, Methodology, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Y.Q. Zhao, Y. Rao, S.P. Dong, et al., Survey on deep learning object detection, Journal of Image and Graphics 25 (4) (2020) 629–654.
[2] T.Y. Lin, M. Maire, S. Belongie, et al., Microsoft coco: common objects in context, in: European Conference on Computer Vision, 2014, pp. 740–755.
[3] M. Everingham, L. Van Gool, C.K.I. Williams, et al., The pascal visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.
[4] X.K. Zhu, S.C. Lyu, X. Wang, et al., TPH-YOLOv5: improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios, in: Proceedings of 2021 IEEE/CVF International Conference on Computer Vision Workshops, IEEE Press, Washington D. C. USA, 2021, pp. 2778–2788.
[5] L. Neumann, A. Vedaldi, Pedestrian and ego-vehicle trajectory prediction from monocular camera, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2021, pp. 10199–10207, https://doi.org/10.1109/CVPR46437.2021.01007.
[6] S.D. Anton, S. Sinh, H.D. Schotten, Anomaly based intrusion detection in industrial data with SVM and random forests, in: 2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), IEEE, 2019, pp. 1–6.
[7] G. Jocher, Yolov5. https://github.com/ultralytics/yolov5, 2020.
[8] S. Ren, K. He, R. Girshick, et al., Faster R-CNN: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2017) 1137–1149.
[9] A. Bochkovskiy, C.Y. Wang, H. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection, 2020.
[10] C.Y. Wang, A. Bochkovskiy, H.Y.M. Liao, YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 2023, pp. 7464–7475.
[11] L. Zhao, S.P. Liu, Small object detection algorithm based on adaptive fusion of global and local image features, Control Decis. 38 (4) (2023) 935–943.
[12] H.Y. Chen, X.J. Zhen, T.T. Zhao, Small object detection model based on feature fusion of attention mechanism, J. Huazhong Univ. Sci. Technol. (Nat. Sci. Ed.) 51 (3) (2023) 60–66.

[13] J. Gong, J. Zhao, F. Li, et al., Vehicle detection in thermal images with an improved yolov3-tiny, in: 2020 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS), 2020, pp. 253–256.
[14] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, ICLR (2016), https://doi.org/10.48550/arXiv.1511.07122.
[15] M. Tan, R. Pang, V. Le Q, EfficientDet: Scalable and Efficient Object Detection// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 10778–10787.
[16] Z. Gevorgyan, SIoU Loss: More Powerful Learning for Bounding Box Regression, 2022 arXiv preprint arXiv: 2205.12740.
[17] A.G. Howard, M.L. Zhu, B. Chen, et al., MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, 2017 arXiv preprint arXiv: 1704.04861.
[18] K. Han, Y.H. Wang, Q. Tian, et al., Ghostnet: more features from cheap operations, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 1580–1589. Seattle.
[19] Y.S. Li, C.Y. Zhang, Y.K. Zhao, et al., Research on lightweight obstacle detection model based on model compression, Laser J. 43 (9) (2022) 38–43.
[20] A. Li, S.J. Sun, C.Y. Zhang, et al., Research on lightweight of improved YOLOv5s track obstacle detection model, Computer Engineering and Applications 59 (4) (2023) 197–207.
[21] X. Li, W. Wang, X. Hu, et al., Selective kernel networks, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 510–519.
[22] T.Y. Lin, P. Dollar, R. Girshick, et al., Feature pyramid networks for object detection, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 936–944.
[23] S. Liu, L. Qi, H. Qin, et al., Path aggregation network for instance segmentation, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 8759–8768.
[24] H. Rezatofighi, N. Tsoi, J.Y. Gwak, et al., Generalized intersection over union: a metric and a loss for bounding box regression, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 658–666.
[25] C.Y. Wang, A. Bochkovskiy, H.Y.M. Liao, Scaled-YOLOv4: scaling cross stage partial network, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 13024–13033.
[26] P. Razzaghi, K. Abbasi, M. Shirazi, S. Rashidi, Multimodal brain tumor detection using multimodal deep transfer learning, Appl. Soft Comput. (2022) 129.
[27] A. Dehghan, P. Razzaghi, K. Abbasi, S. Gharaghani, TripletMultiDTI: multimodal representation learning in drug-target interaction prediction with triplet loss function, Expert Syst. Appl. (2023) 232.