ITC 2/53
Information Technology and Control
Vol. 53 / No. 2 / 2024
pp. 601-618
DOI 10.5755/j01.itc.53.2.36336

Elderly Fall Detection Algorithm Based on Improved YOLOv5s

Received 2024/02/12 | Accepted after revision 2024/04/29

HOW TO CITE: Luo, Z., Jia, S., Niu, H., Zhao, Y., Zeng, X., Dong, G. (2024). Elderly Fall Detection Algorithm Based on Improved YOLOv5s. *Information Technology and Control*, 53(2), 601-618. https://doi.org/10.5755/j01.itc.53.2.36336

Elderly Fall Detection Algorithm Based on Improved YOLOv5s

Zhongze Luo, Siying Jia, Hongjun Niu, Yifu Zhao, Xiaoyu Zeng, Guanghui Dong
College of computer and control engineering, Northeast Forestry University, Harbin, 150040, China

Corresponding author: dghRobert@126.com

The indoor fall detection for the elderly can effectively help the treatment after falling, but many existing detection methods have the problems of inconvenient use, high misjudgement rate and slow speed. Using deep learning methods can effectively solve these problems, and YOLOv5s is a kind of deep learning algorithm that can perform real-time fall detection. In order to achieve a more lightweight and higher detection accuracy, this paper proposes a fall detection algorithm for the elderly based on improved YOLOv5s, called YOLOv5s-GCC. Firstly, the original Conv and C3 structures are replaced by GhostConv and C3GhostV2 structures in backbone to achieve model lightweight, which reduces model computation and improves accuracy. Secondly, the lightweight upsampling operator CARAFE is introduced to expand the receptive field for data feature fusion and reduce the loss of feature information in upsampling. Finally, the deepest C3 is integrated with CBAM attention mechanism in the neck, because the deepest neck receives more abundant feature information, and CBAM can increase the efficiency of the algorithm in extracting important information from the feature map. Experimental results show that YOLOv5s-GCC has increased by 1.2% to 0.935 on the hybrid open source fall dataset mAP@0.5; FLOPs decreased by 29.1%. Params are reduced by 27.5% and have obvious advantages over similar object detection algorithms.

KEYWORDS: Fall detection, Improved YOLOv5s, GhostNetV2, CARAFE, CBAM attention mechanism.

1. Introduction

According to the seventh national census data of the National Bureau of Statistics of China, China's population is projected to surpass 1.4 billion in 2020, exhibiting an average annual growth rate of 0.53%. Among this demographic, individuals aged 60 and above are expected to exceed 260 million, constituting approximately 18.7% of the national population [2]. Given the progress in economy and society, ensuring the

well-being of empty nesters has become increasingly imperative. Data from China's disease surveillance system reveals that falls remain a predominant cause of injury among elderly individuals nationwide. In fact, over 20% of such incidents result in severe injuries for older adults in China; even healthy seniors face a staggering probability (17%) of becoming serious patients due to falls [4]. Consequently, timely detection and notification regarding falls play a pivotal role in safeguarding the safety and security of empty nesters.

At present, the real-time fall detection is mainly divided into three types: wearable fall detection, environmental fall detection and computer vision fall detection. The main problem of the first method is that the elderly may forget to wear the device in daily life, and the main problem of the second method is that the layout cost of the environmental equipment is very high, and the misjudgement rate of these two methods is generally high.

Fall detection based on computer vision has the characteristics of convenient use, low misjudgement rate, good real-time performance, and many applicable scenarios. The feature extraction algorithms can be mainly divided into three types: threshold analysis, detection algorithm based on machine learning, and detection algorithm based on deep learning [28].

2. Related Work

Since 2012, when Krzhevsky et al. [11] proposed the AlexNet model, the deep learning model using convolutional neural network (CNN) for feature extraction has gradually become popular, and has been widely used in fall detection. Nowadays, the object detection algorithm based on deep learning can be divided into one-stage and two-stage. The two-stage algorithms include R-CNN [6], Fast RCNN [5], Faster RCNN [18], and R-FCN [3], but the processing process of these algorithms is relatively cumbersome, and the detection speed is slow. The representative of one-stage algorithm is SSD [13] algorithm and YOLO [1], [15-17] series, which have the characteristics of fast detection speed, good real-time performance, and still maintain good accuracy. For example, Wang et al. [24] used YOLOv3 algorithm and introduced anchor point parameters to detect human falls. Panigrahi et al. [21] proposed an improved lightweight MS-ML-SNYOLOv3 network to obtain better detection results by increasing the receptive field. Li et al. [12] improved the YOLOv5 network by embedding the SE (Squeeze-and-Excitation, SE [10]) channel attention mechanism. SE pools the global average channel information of the input feature map, and then normalizes the compressed information and multiplies it on the input feature map, enhancing the model's ability to capture the information of the object of interest and improving the detection accuracy. Shen et al. [20] proposed a reparameterized backbone network, in which the Conv module was replaced by DBBConv and DBBC3 modules, and proposed a new feature enhancement module (FEM) to enhance the feature representation and feature fusion of the region of interest (ROI), and added FEM to the feature pyramid network (FPN) to improve detection accuracy. Yang et al. [25] proposed MSF-YOLO to fuse multi-scale features of images. Compared with the original ResNet unit, the single convolution scale is increased to four convolution scales, and the features under each different perception field are fused to obtain rich hierarchical information from the image. Zhao et al. [27] proposed a novel attention module SDI based on coordinate attention and aliasing attention. The module enhances the feature extraction ability of detection targets. They proposed a novel convolutional neural network model for fall detection in open space, named YOLO-Fall. The above methods have achieved good results in fall detection, but there are still problems such as large model volume, complex parameter quantity, poor model feature fusion ability, and insufficient attention mechanism introduced. If the algorithm has a large number of parameters, it is difficult to deploy to other mobile devices for real-time fall detection.

In view of the above problems, this paper proposes an elderly fall detection algorithm based on improved YOLOv5s, which achieves the balance between fast detection and high accuracy, and meets the conditions for deployment on hardware platforms. The main contents are as follows: (1) Design a lightweight backbone network, use the GhostNetV2 network idea, replace the bottleneck structure in the original backbone network C3 module with the GhostNetV2 bottleneck module in the GhostNetV2 network, build the C3GhostV2 module, and use it with Ghostconv

convolution to effectively reduce the number of model parameters and reduce the calculation cost. (2) Introduce the CARAFE upsampling operator in the feature extraction network, so that the model can obtain more detailed information in the upsampling process, and effectively reduce the information loss caused by upsampling operation. (3) The convolutional attention model (CBAM) is introduced into the C3 module of the feature extraction network, which is helpful for the reasonable positioning of the bounding box and the solution of the gradient disappearance problem.

3. YOLOv5 Detection Algorithm

YOLOv5 is a classic algorithm of the YOLO series, including YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x four models. All models are composed of four parts: input, backbone network, neck network, and output detection end. The input is responsible for receiving image data and preprocessing it to adapt to the input requirements of the model, such as adjusting the image size to the size required by the model, normalizing the image, etc. The backbone network is the core of the entire model and is responsible for extracting the feature information of the input image. The neck network is located between the backbone network and the output detection end and is responsible for further extracting and integrating the feature information extracted by the backbone network to better adapt to the specific object detection task. The output detection end is the last part of the model and is responsible for outputting the category, location, confidence and other information of the object according to the feature information passed by the neck network and the requirements of the predefined object detection task, so as to complete the object detection task. The input part contains data preprocessing operations such as Mosaic image enhancement, adaptive anchor calculation, and adaptive image scaling, which increases the diversity of the dataset, avoids the inaccuracy of manually setting anchor parameters, solves the problem of inconsistent detection target size, and improves the detection accuracy and robustness. To be specific, Mosaic image augmentation introduces pixelation in certain areas of the training data, creating visual distortions and confusion. This simulates the real-world appearance of objects in complex scenes, helping the model adapt better to complex environments and improve its generalization ability and accuracy. Adaptive anchor calculation dynamically adjusts the sizes and ratios of anchor boxes in the model based on the actual sizes and proportions of objects in the training dataset. This ensures that the model has good adaptability when detecting objects of different sizes and proportions, thereby enhancing accuracy when dealing with objects of different scales. Adaptive image scaling dynamically adjusts the size of images based on the distribution of object sizes and proportions in the training data. This allows the model to encounter a variety of object sizes during training, helping it learn to adapt to different scale objects and improve detection capabilities and accuracy. The backbone network is mainly divided into three parts: Conv module, CSPDarkNet53 backbone network, and SPPF structure. In YOLOv5-6.0 version, the Focus module in the old version is replaced by a convolution layer with a size of 6×6, a stride of 2, and a padding of 2, which helps to improve the model efficiency. This is because the Focus module used in earlier versions of YOLOv5 to reduce the size of the input image and increase the number of channels, which is achieved through slicing operations and channel stacking, a process that, while effective, has computational efficiency limitations. Using a single convolutional layer directly instead of the Focus module simplifies the overall structure of the model, reduces the complexity of the model, and makes the model more lightweight. This simplification helps in training and deployment of the model, especially on resource-limited devices. CSPDarkNet53 backbone network contains multiple CSP modules. In YOLOv5-6.0 version, the BottleneckCSP module in the old version is replaced by a more streamlined C3 module. The C3 module transforms the input feature map with two 1×1 convolutions, one way into the Bottleneck module, through two convolution layers with a size of 1×1 and 3×3, extracts features and performs feature fusion, and the other way is directly passed down to the Bottleneck module output feature map for Concat channel stacking, which obtains rich gradient combination information and faster inference speed. The C3 module optimizes the process of feature extraction by introducing more convolutional layers and carefully configuring the parameters of these convolutional layers. This not only enhances the feature extraction ability of the module, but also improves the adaptability and

Figure 1
Model structure of YOLOv5s-v6.0

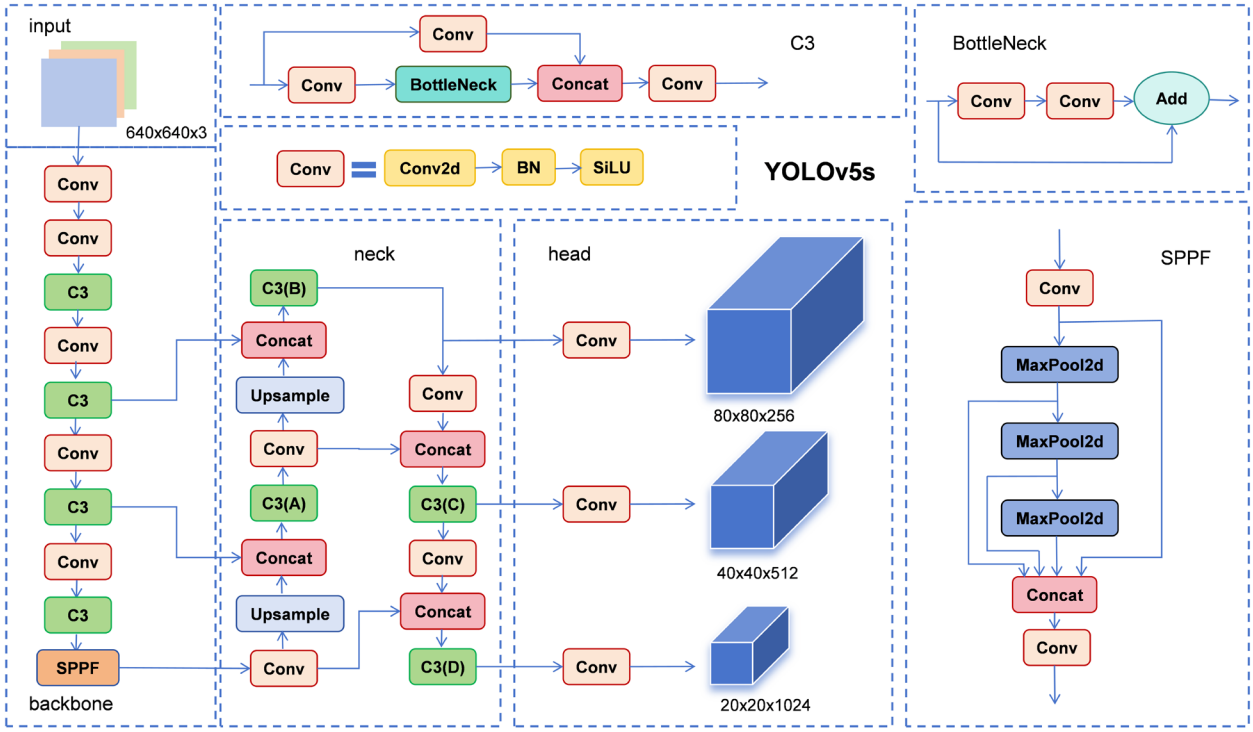
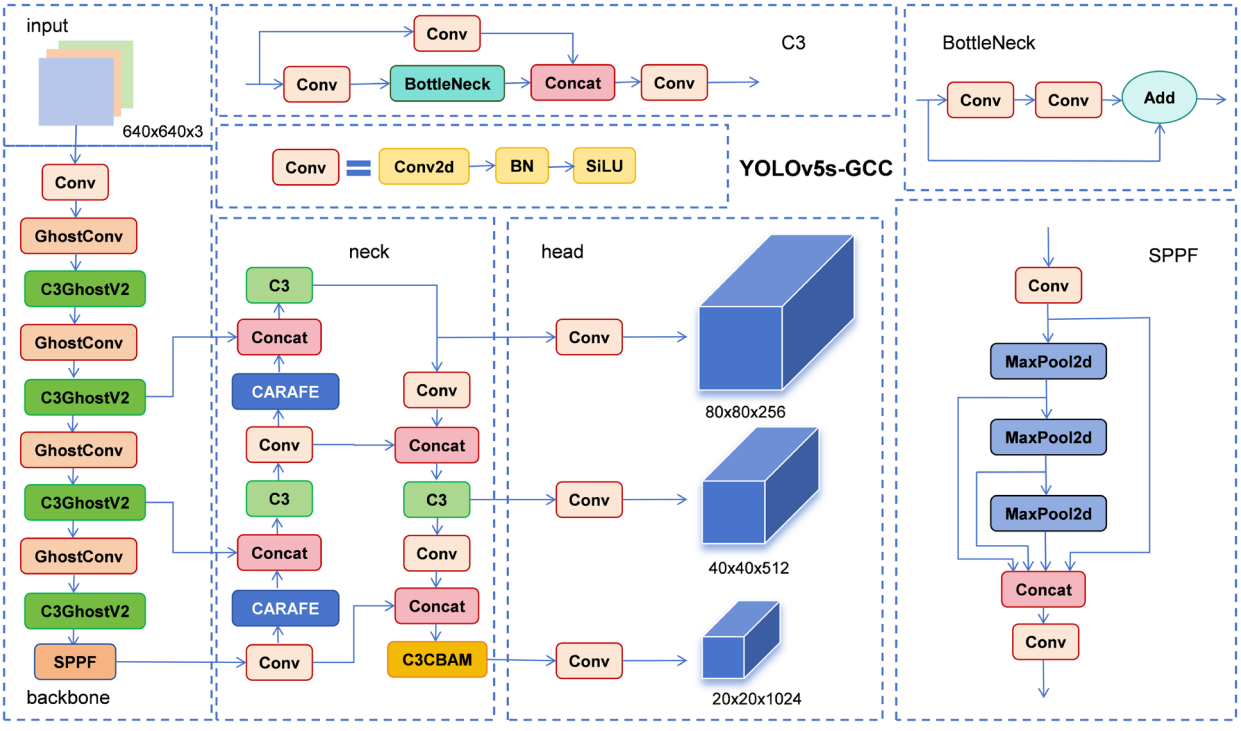


Figure 2
Model structure of YOLOv5s-GCC



recognition accuracy of the model to complex scenes through finer-grained feature processing. In addition, the C3 module improves the parameter efficiency of the network through this optimized convolutional layer design, that is, while maintaining or improving the feature extraction ability, the number of parameters and computational complexity are minimized. In YOLOv5-6.0 version, the SPP structure is replaced by SPPF structure, which uses multiple small-sized pooled kernel cascades to fuse feature maps of different receptive fields at a faster running speed. The neck network adopts FPN+PANet feature pyramid. FPN transfers rich semantic information from deep to shallow for feature fusion from top to bottom, and PANet transfers stronger position information from shallow to deep for feature fusion from bottom to top. The output detection end contains three detection layers of different sizes, corresponding to three different size feature maps in the neck network. CIOU_Loss is used as the loss function to measure the difference between predicted frames and real frames, and non-maximum suppression NMS is introduced to filter repetitive predicted frames, improving the detection efficiency. Because YOLOv5s is the model with the smallest convolution depth and feature map width in YOLOv5, with the smallest amount of calculation and parameters, it is suitable for deployment to other mobile devices, which is in line with the research direction of this paper, YOLOv5s-6.0 version is selected as the improvement object. The model structure of YOLOv5s-v6.0 is shown in Figure 1.

In order to improve the detection performance of the model, this paper designed the object detection model YOLOv5s-GCC based on YOLOv5s as the benchmark model. The improved model is shown in Figure 2.

4. Improved YOLOv5s Model

4.1. Lightweight Backbone Design Based on GhostNetV2

Traditional feature networks have the characteristics of redundant feature information and large amount of parameters. In order to solve these problems, the mainstream lightweight networks at present mainly include MobileNet [9], ShuffleNet [26] and GhostNet [7]. The MobileNet model divides the standard volume into depthwise convolution and pointwise convolution, inverted residuals are introduced in MobileNetV2 [19], and SE attention mechanism module is introduced in MobileNetV3 [8]. In ShuffleNet, channel shuffle operation is used to help information flow in feature channels, and channel split operation is proposed in ShuffleNetV2 [14] to reduce memory access cost. The effect on the ImageNet dataset shows that GhostNetV2 is ahead of other networks in terms of classification accuracy, parameter number and detection speed [22]. Therefore, this paper introduces GhostNetV2 into YOLOv5s and proposes the C3GhostV2 module, which has obvious advantages compared with the traditional C3 module and is used together with GhostConv in GhostNet.

GhostNet is a lightweight network designed by Huawei Noah Ark Laboratory in 2020. It can make full use of limited computing resources to extract redundant feature information, and maintain the performance of the network model while reducing the number of model parameters. GhostConv is a convolution module in GhostNet. It can generate enough feature information at the lowest cost through a series of simple linear operations, so as to improve the network's ability to mine original information, and can replace ordinary convolution. However, GhostNet has certain limitations. The convolution and point-by-point convolution in linear transformation have no information exchange with other pixels, and the ability to capture spatial information is weak. In order to improve this shortcoming, Huawei Noah Ark Laboratory launched a new lightweight network structure GhostNetV2 in 2022. The convolution method of decoupled full-connected DFC attention mechanism is added to the Ghost convolution method in parallel, which obtains better accuracy performance while making the network model lightweight.

Figure 3
Comparison between Ghost module and traditional convolution

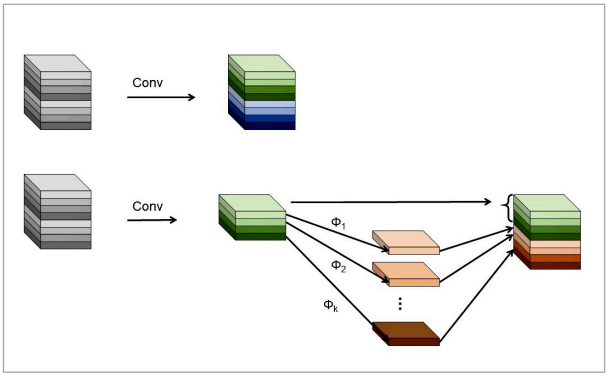
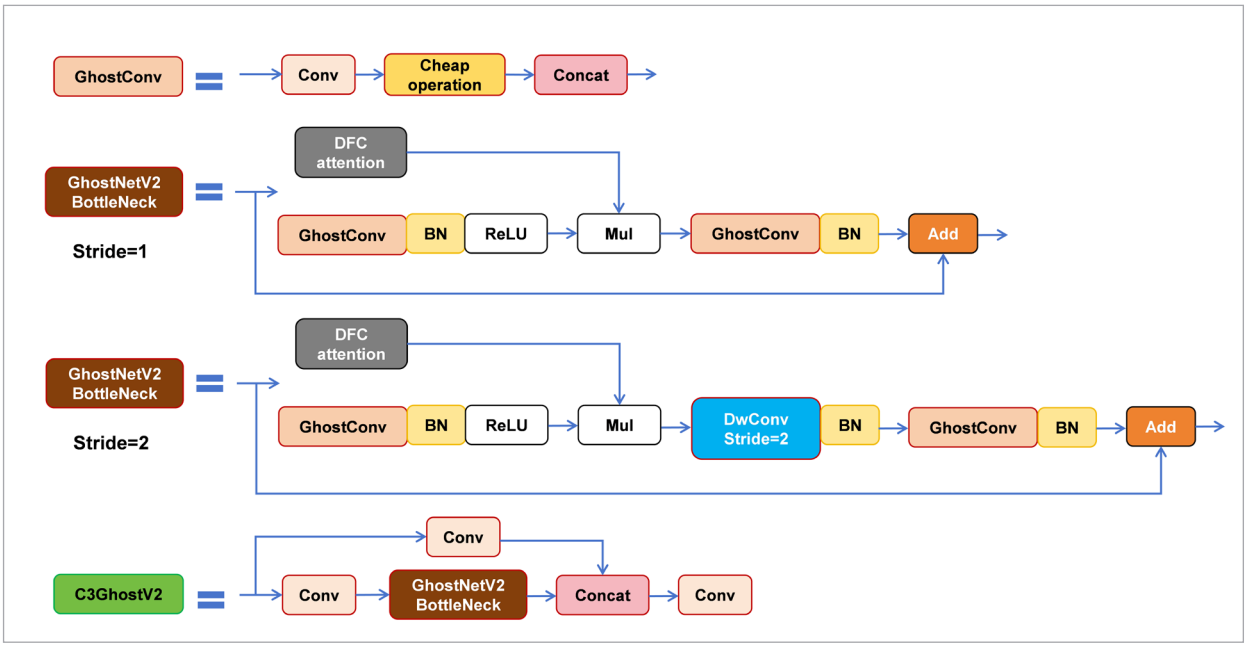


Figure 4
Structure of GhostNetV2 Bottleneck and C3GhostV2



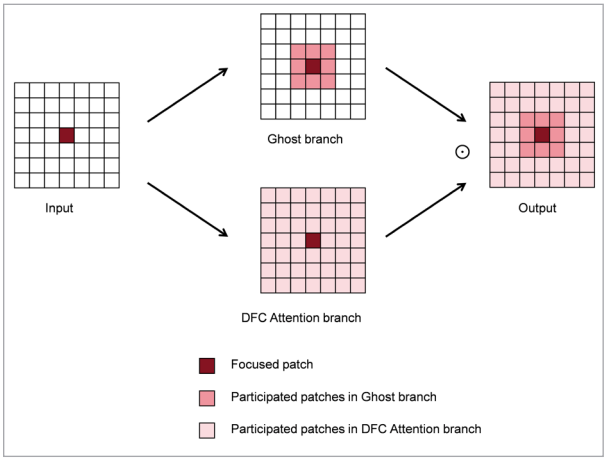
Compared with the traditional convolution method, for a given input feature X ∈ H×W×C (H, W, and C are the height, width, and channel number of the feature map respectively), the Ghost module divides the output channels into two parts. The first part is conventional convolution, but strictly controls the number of convolution output layers to generate part of the feature map. The second part generates some other feature maps through linear transformation with low computational cost, and finally stitches the two parts of the feature map to generate the final feature map, so as to eliminate the redundancy of the feature map and obtain a better lightweight model. In deep learning models, many feature maps are similar, and therefore redundant. By reducing this redundancy, the Ghost module is able to reduce the amount of computation and the number of parameters.

In GhostNetV2, the decoupled fully connected attention mechanism DFC is introduced, that is, in the low-rank feature graph, the horizontal and vertical fully connected layers are used to realize the attention graph with a global receptive field. That is, the input feature X is sent to two branches, one is the Ghost branch, which gets the output feature Y; the other is the DFC branch, which gets the attention matrix A. Finally, the two branches are dot multiplied. Formula 1 illustrates this process.

$$O = sigmoid(A) \odot v(X). \quad (1)$$

DFC enables the network to better focus and process spatial information by introducing fully connected attention layers between feature maps. This approach

Figure 5
The module of GhostNetV2



allows the model to capture spatial relationships on a global scale, rather than just local regions. This helps the model to more effectively understand and process spatial structure and content in images.

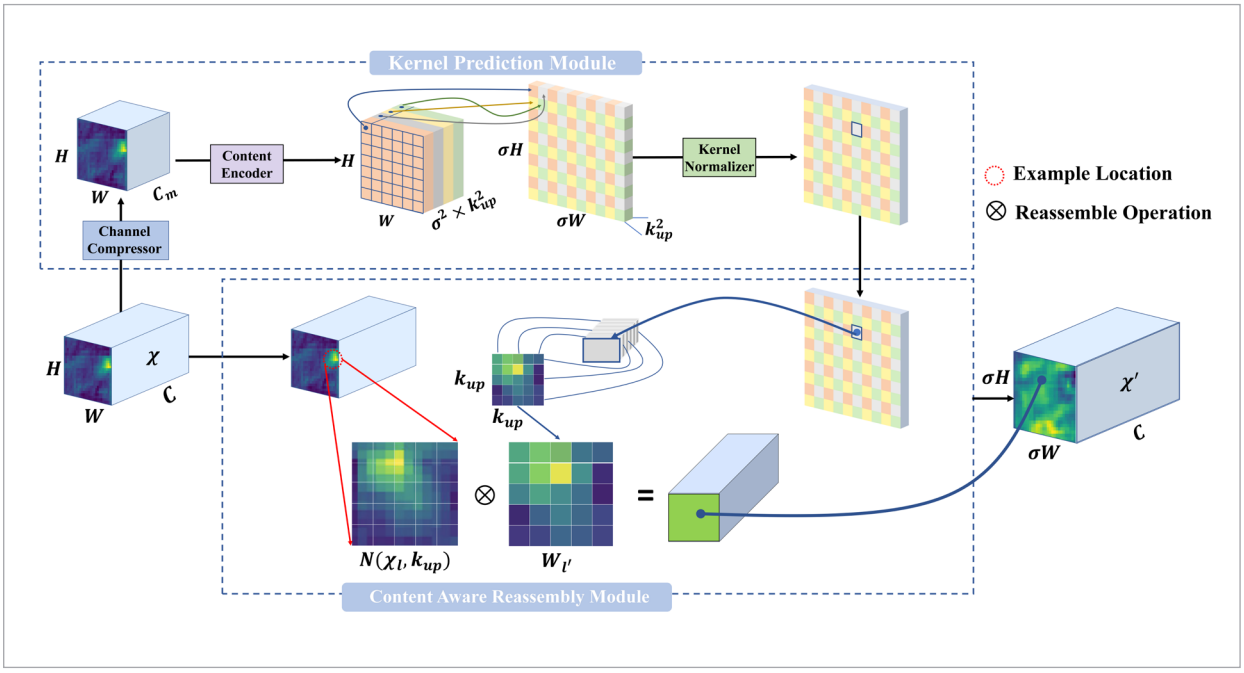
GhostNetV2 parallelizes network computing by grouping channels, which can adapt to input data of different sizes and have less computing overhead. In addition, the use of low-rank decomposition technology can reduce the number of redundant parameters while ensuring model accuracy. In the backbone network, we use GhostConv and C3GhostV2 structures to ensure accuracy without loss while being lightweight.

4.2. Upsampling Operator CARAFE

The nearest neighbor interpolation is used by default in YOLOv5, which determines the upsampling kernel by the spatial position of the pixel point. The semantic information of the feature map is not used, which affects the positioning and recognition of the defect target, cannot achieve the optimal detection effect, and has a small receptive field. In view of these problems, this paper uses the lightweight and efficient upsampling operator CARAFE [23], which maintains lightweight functions with a small number of parameters and calculations. Compared with the nearest neighbor interpolation, it has three main advantages: 1. Large receptive field, able to aggregate context information; 2. Good content awareness ability, dynamically generates adaptive kernels, rather than using a fixed kernel for all samples (e.g. deconvolution), supporting instance-specific content awareness processing; 3. Light weight, fast calculation speed. CARAFE introduces little computing overhead and can be easily integrated into modern network architectures.

CARAFE is divided into two main modules, namely the upsampling kernel prediction module and the feature reorganization module. Assuming the upsampling rate is σ, given an input feature map with shape H×W×C, in the upsampling prediction module, for the input feature image X, the channel compression is first performed through the ordinary convolution operation to generate the compressed image Y with size H×W×C_m, which reduces the network calculation. Set the size of the upsampling kernel as k_{up}×k_{up}. Combined with the input image size and the upsampling rate σ, the predicted size of the upsampling kernel is obtained through the convolution operation, and the

Figure 6
Structure of CARAFE



size of the upsampling kernel is σH×σW×k_{up}×k_{up}. Finally, the softmax algorithm is used to normalize the sampling kernel, so that the sum of the weights of the convolution kernel is 1. For any target position l'(i',j') in the new output graph X', there is a l(i,j) corresponding to it in the original feature graph, and the mapping relationship is i=[i'/σ], j=[j'/σ]. N(X_l,k) is represented as the k×k interval of X with l as the center, and the upsampling kernel prediction module ψ predicts the position kernel $W_{l'}$ of each position l' according to X_l. Formula 2 illustrates this process.

$$W_{l'} = \psi(N(X_l, k_{encoder})). \quad (2)$$

In the feature recombination module, for the obtained compressed image Y, the feature map with the corresponding size of k_{up}×k_{up} is taken at the center of the feature map Y and the prediction of the upsampling kernel is performed to perform convolution operation. Finally, the output feature map X' with the size of σH×σW×C is obtained. Formula 3 illustrates this process.

$$X'_{l'} = \sum_{n=-r}^{r} \sum_{m=-r}^{r} W_{l'(n,m)} \bullet X_{(i+n,\, j+m)}. \quad (3)$$

$X'_{l'}$ represents the l'(i',j') position of the new feature map $X_{l'}$, W represents the convolution kernel, and X represents the original feature map. The original feature map is the region from −r to r with (i,j) as the center.

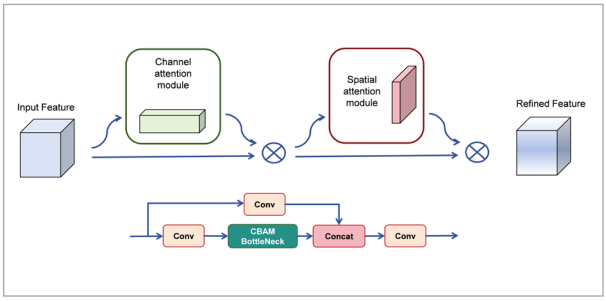
In summary, CARAFE offers a more advanced upsampling method compared to the default nearest neighbor interpolation in YOLOv5s. Its main advantages are the ability to generate smoother feature maps with richer semantic information, which helps improve the detection accuracy of small objects and the precision of boundary localization. Additionally, CARAFE employs content-aware weighting, allowing the reconstructed feature maps to more accurately reflect the detailed characteristics of the input data.

4.3. C3CBAM Attention Mechanism

The attention mechanism module can effectively improve the efficiency of the network by selecting information features, estimating the importance of different information, weakening useless information and strengthening important information. Due to the factors of image occlusion, complex background, and small proportion of distant target image, the detection accuracy will be seriously affected by the real situation. In order to solve these problems, we introduce the attention mechanism. By weighting important features, attention mechanisms can identify and emphasize target objects in images, even when these objects are partially occluded or highly fused with the surrounding environment. This enhances the model's ability to capture details, especially when dealing with images containing complex scenes or multiple overlapping targets, and can effectively distinguish foreground targets from background noise, significantly improving the quality and reliability of detection results. Therefore, the application of attention mechanisms in advanced object detection models such as YOLOv5s is crucial to improve performance in complex visual environments.

CBAM (Convolutional Block Attention Module) contains two submodules, CAM (Channel Attention Module) and SAM (Spartial Attention Module), which are respectively channel and spatial attention. In the channel attention module, the channel dimension is kept constant, the spatial dimension is compressed, and the meaningful information in the input image is focused on. In the spatial attention module, the spatial dimension is kept constant, the channel dimension is compressed, and the focus is on the location information of the target.

Figure 7
Structure of CBAM attention and C3CBAM module



In this paper, CBAM module is introduced into C3 module to form C3CBAM module, which improves the network's ability to extract the characteristics of the detection target.

5. Experiment and Analysis

5.1. Dataset and Evaluation Index

The experimental data set is a mixture of three public datasets: UR Fall Detection Dataset, Fall Detection Dataset (2017 IAPR MVA Conference), and Multiple Cameras Fall Dataset, with a total of 3502 images. Since there are few falls covered in the dataset, in order to enrich the fall backgrounds under different conditions, 801 falls in COCO dataset are selected, a total of 4303 images, which are divided into training set, validation set, and test set according to 8:1:1. Labeling software is used for data labeling, and the labels of the dataset are Fall, Stand, and Sit. The labeling of some datasets is shown in Figure 8.

UR Fall Detection Dataset: This dataset was developed by the University of Rochester to support research in fall detection and everyday behavior recognition. It contains videos from different angles and in different lighting environments, covering a variety of fall and non-fall scenarios. Individuals in the videos perform various activities such as walking, running, sitting, and falling, to provide diverse data for model training and testing.

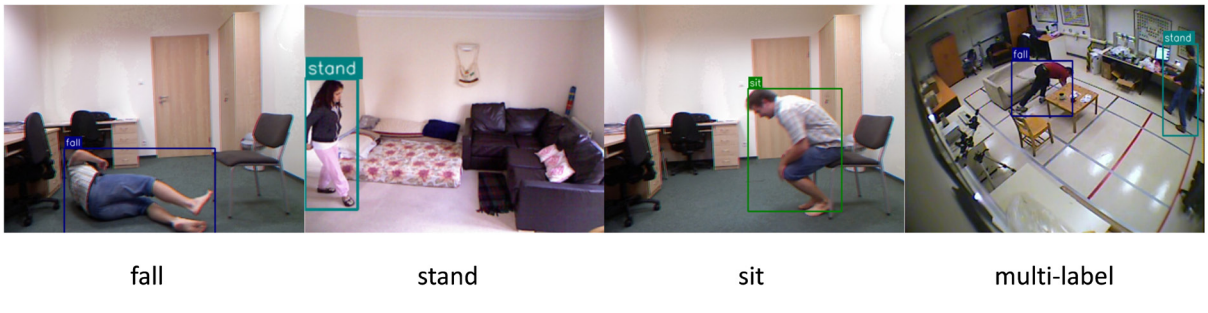
Fall Detection Dataset (2017 IAPR MVA Conference): This dataset was presented at the 15th IAPR International Conference on Machine Vision Applications in 2017 and was designed specifically for fall detection research. The images in the dataset are recorded in 5 different rooms which consist of 8 different view angles. There are 5 different participants out of which there are two male participants of age 32 and 50 and three female participants of age 19, 28 and 40. All the activities of the participants represent 5 different categories of poses that are standing, sitting, lying, bending and crawling. There is only one participant in each image.

Multiple Cameras Fall Dataset: This dataset contains 24 scenes recorded using 8 IP cameras. The dataset focuses on using a multi-camera system to improve the accuracy of fall detection. By collecting data from different angles, the dataset aims to address the perspective limitations that a single camera may encounter. This setup helps to generate a more comprehensive view that can provide more details about fall events, allowing fall detection algorithms to work more accurately in complex environments.

COCO Dataset: This is a widely used large-scale image dataset, which contains more than 200,000 images, dedicated to computer vision tasks such as object detection, human key-point detection, and image description. The dataset was released by the Microsoft team to promote research and development of scene understanding technology. The COCO Dataset provides images of objects in their natural environment, emphasizing the interaction between different objects and the overall understanding of the scene.

By combining these datasets, the strength and robustness of the fall detection system can be significantly improved. Here are some key advantages: 1. These datasets contain data collected from different environments, which improves the generalization ability of the model in various scenarios. In particular, the Multiple Cameras Fall Dataset increases the perspective diversity of the data by providing multiple camera views. In addition, these datasets may involve differ-

Figure 8
Sample images of the dataset



ent populations, including different ages, genders, body types and behavior patterns, which improves the model's ability to deal with the diversity of human behavior. 2. Combining different datasets can provide more comprehensive and integrated data, so that the model can learn more complex features and patterns, improving the robustness and accuracy under complex real-world conditions. 3. Using data from multiple sources increases the size and diversity of the training set, helping to reduce model overfitting, making the model more able to generalize to unseen data.

5.2. Experimental Environment

The experimental computer processor is Intel(R) Core(TM) i7-12650H CPU @ 2.30 GHz, the GPU is NVIDIA A100 80GB PCIe, the operating system is Ubuntu 20.04.2, the deep learning framework is PyTorch 2.0.1, the Python version is 3.8.0, and the CUDA version is 11.7.

The model training parameters are set as follows: the batch size is 32, the epoch is 100, the imgz is 640, the initial learning rate is 0.01, the weight attenuation coefficient is 0.0005, and the momentum is 0.937. SGD is used as the optimizer for iterative training.

5.3. Experimental Evaluation Index

In this paper, average precision (AP) is used as the evaluation index for each defect class, and mean average precision (mAP) is used to evaluate the performance of the entire network model. AP (Average Precision) refers to the area under the PR curve (Precision-Recall Curve), which is the average of the accuracy at different recall points. mAP@0.5 refers to the average of the AP values of all classes when the IoU value is equal to 0.5. mAP@0.5:0.95 represents the average mAP at different IoU thresholds (from 0.5 to 0.95, step size 0.05), which takes into account the accuracy P and recall R of object detection. Precision refers to the proportion of all results predicted as positive samples that are correctly predicted, and recall refers to the proportion of all positive samples that are correctly predicted as positive samples. The number of parameters (Params) of the model is used to evaluate the complexity of the model. The smaller the number of parameters, the lighter the model is. The FLOPs index is used to evaluate the computational efficiency of the model. The lower the FLOPs, the higher the computational efficiency of the model is.

$$P = \frac{P_T}{P_T + P_F} \tag{4}$$

$$R = \frac{P_T}{P_T + N_F} \tag{5}$$

$$AP = \int_0^1 P(R)dR \tag{6}$$

$$mAP = \frac{1}{m}\sum_{i=1}^{m} P_{Ai} \tag{7}$$

P_T is the number of positive samples that are correctly predicted; P_F is the number of positive samples that are wrongly predicted; N_F is the number of negative samples that are wrongly predicted; m is the number of classes that are detected.

5.4. Analysis of Experimental Results

5.4.1. Experimental Analysis of Lightweight Improvement

In this paper, the original YOLOv5s model is improved by replacing the ordinary convolution and C3 module with GhostConv and C3GhostV2 modules. In order to verify whether different replacement parts can effectively reduce the amount of model parameters and explore the impact of different replacement parts on the model accuracy, four sets of experiments are designed, which are YOLOv5s, YOLOv5s-all-Ghost, YOLOv5s-backbone-Ghost and YOLOv5s-neck-Ghost. The model in which all the Conv and C3 are replaced by GhostConv and C3GhostV2 modules is named YOLOv5s-all-Ghost. The model in which all the Conv and C3 in the backbone are replaced by GhostConv and C3GhostV2 modules is named YOLOv5s-backbone-Ghost. The model in which all the Conv and C3 in the neck are replaced by GhostConv and C3GhostV2 modules is named YOLOv5s-neck-Ghost. The comparison results are shown in Table 1.

As can be seen from Table 1, after the model replaces GhostConv and C3GhostV2 modules completely, the parameter amount and FLOPs of the improved model are reduced by 42.74% and 43.04% respectively, the detection accuracy is improved by 0.3%, and the generalization ability of the model is good. After the model backbone replaces GhostConv and C3GhostV2

Table 1
Comparison of lightweight improvement effect in different positions

Model	mAP@0.5	Params(M)	FLOPs(G)
YOLOv5s	0.923	7.02	15.8
YOLOv5s-all-Ghost	0.926	4.02	9.0
YOLOv5s-Backbone-Ghost	0.929	5.27	11.1
YOLOv5s-Neck-Ghost	0.919	5.77	13.6

modules, the parameter amount and FLOPs of the improved model are reduced by 24.93% and 29.75%, respectively, the detection accuracy is improved by 0.6%, and the model has the best generalization ability. After the YOLOv5s model neck replaces GhostConv and C3GhostV2 modules, the parameter amount and FLOPs of the improved model are reduced by 17.81% and 13.92% respectively, the detection accuracy is reduced by 0.4%, the detection accuracy is reduced by much, and the generalization ability is poor. In order to obtain the best detection accuracy and the best generalization ability of the model, this paper determines the experimental scheme of the YOLOv5s model backbone network replacing C3GhostV2 and Ghost modules, which can also reduce a certain amount of model parameters, and the model is named YOLOv5s-G.

5.4.2. Experimental Analysis of the Improvement of the Upsampling Operator

In order to verify the effectiveness of the upsampling operator CARAFE, the algorithm using the upsampling operator CARAFE on the basis of the algorithm YOLOv5s-G is called YOLOv5s-GC. The upsampling operator CARAFE is used to replace the nearest neighbor interpolation of the original neck network to obtain a higher quality upsampling feature map. In the paper [23], Wang et al. believe that $k_{encoder} = k_{up} - 2$ can achieve the best performance under the condition of a similar amount of calculation, and only by increasing the two at the same time can the performance be improved, but it will also increase the amount of calculation. Therefore, we set the CARAFE operator of $k_{encoder} = 3$, $k_{up} = 5$ to seek to improve the model accuracy as much as possible under the condition of a certain amount of calculation. The comparison results are shown in Table 2.

As can be seen from Table 2, after using CARAFE operator, mAP@0.5 and mAP@0.5:0.95 are improved compared with the basic model using the nearest neighbor interpolation, mAP@0.5 is improved by 0.1%, and mAP@0.5:0.95 is improved by 1.1%, indicating that the average mAP of the model at different IoU thresholds is improved, and Precision is improved by 3%, indicating that the accuracy of the model is improved to some extent. The experimental results show that the CARAFE operator can better capture the spatial relationship between features, which can make the model more accurate.

5.4.3. Experimental Analysis of C3CBAM Improvement

5.4.3.1. Analysis of C3CBAM Replacement Positions

In order to verify the effectiveness of the C3CBAM module with improved attention mechanism, the algorithm using C3CBAM on the basis of YOLOv5s-GC is called YOLOv5s-GCC. In addition, in order to explore the optimal position of C3CBAM embedding, this paper fused CBAM at the four C3 modules of the neck network, which were labeled as CBAM_A, CBAM_B, CBAM_C and CBAM_D from shallow to deep. The other parts remained unchanged, and the comparison experiment was conducted with YOLOv5s-GC algorithm. The comparison results are shown in Table 3, where "√" indicates the use of an improved method.

Table 2
Comparison results of upsampling operator

Upsampling Operator	Precision	Recall	mAP@0.5	mAP@0.5:0.95	FLOPs(G)	Params(M)
nearest_neighbor	0.913	0.89	0.929	0.67	11.1	5.27
CARAFE	0.943	0.88	0.93	0.681	11.4	5.41

Table 3
Comparison of improvement effects of C3CBAM modules in different positions

Model	A	B	C	D	mAP@0.5
YOLOv5s-GC					0.93
YOLOv5s-GC-CBAM_A	√				0.927
YOLOv5s-GC-CBAM_B		√			0.931
YOLOv5s-GC-CBAM_C			√		0.926
YOLOv5s-GC-CBAM_D				√	0.935
YOLOv5sGC-CBAM_ABCD	√	√	√	√	0.93

As can be seen from Table 3, not all fusions of CBAM modules at all positions can improve the detection effect. After the fusion of the deepest CBAM_D part, mAP@0.5 is increased by 0.5%, which is the best effect, and the fusion of the rest parts is not good.

The fusion works best when CBAM is integrated at the top of the PANet, i.e. in the case of CBAM_D. We analyze this result for two main reasons: 1. The lateral connections of the PANet allow for the fusion of feature information from different layers of the Backbone. These connections enhance the flow of information and provide a more detailed and comprehensive feature representation by combining high-resolution but semantically shallow low-level features with low-resolution but semantically rich deep-level features. The fusion of CBAM, especially at the top layer after these lateral connections from the Backbone, allows the model to further refine and focus on this fused information; 2. PANet is known for its ability to enhance feature hierarchy through bottom-up paths and horizontal connections. By placing CBAM at the top of the PANet, you effectively take advantage of the rich hierarchical information provided by the bottom layer. CBAM's spatial and channel attention mechanisms refine this information, allowing the network to more effectively encapsulate it into the fusion features for prediction.

5.4.3.2. Comparison to Other Attention Mechanisms

In order to evaluate the improvement effect of the CBAM attention mechanism module selected in this paper, we fused the same C3 position of C3CBAM in YOLOv5s-GC and fused different attention mechanisms of SE and ECA. The SE mechanism focuses on feature weighting between feature channels, learns the importance of each channel, adjusts the weight of the channel according to the importance to enhance important features and suppress features that are not important for the current task. The two fully connected layers of the SE mechanism will reduce the channel size and there is a problem of channel feature loss. In the ECA mechanism, the fully connected layer is removed and one-dimensional convolution is used to complete the information interaction between channels. However, these two attention mechanisms only focus on channel information, while CBAM introduces two analysis dimensions of spatial attention and channel attention at the same time. It not only processes the

Table 4
Comparison of the improvement effect of different attention mechanism modules

Model	mAP@0.5	Precision/%	Recall/%	FLOPs(G)	Params(M)
YOLOv5s-GC	0.93	0.943	0.88	11.4	5.41
YOLOv5s-GC+C3SE	0.929	0.926	0.874	11.4	5.42
YOLOv5s-GC+C3ECA	0.934	0.922	0.895	11.4	5.41
YOLOv5s-GC+C3CBAM	0.935	0.912	0.911	11.2	5.09

allocation of feature map channels through channel attention, but also pays more attention to the pixel area in the image that plays a decisive role in classification and ignores the irrelevant area through spatial attention. The comparison results are shown in Table 4.

As can be seen from Table 4, the maximum mAP can be obtained by integrating the CBAM attention mechanism at the same position, which is 0.5% higher than YOLOv5s. Moreover, compared with other attention mechanisms, the model calculation is smaller and the number of parameters is greatly reduced while improving more accuracy, indicating that CBAM better improves the model performance through the attention mechanism in the two dimensions of space and channel.

6. Contrastive Analysis

6.1. Ablation Experiments

In this paper, improvements are made based on the YOLOv5s model, which are lightweight backbone improvement, upsampling operator CARAFE improvement, and C3 module fusion CBAM attention mechanism. In order to fully verify the effectiveness of the improvements proposed in this paper, ablation experiments are conducted on mixed datasets to verify the importance of each improvement. Each improvement is embedded into the YOLOv5s model in turn, and the same training parameters and environmental conditions are used in each set of experiments. The experimental results are shown in Table 5. "√" indicates that a certain improvement method is used.

It can be seen from Table 5 that the detection performance of YOLOv5s model is low. After the improvement of lightweight backbone, FLOPs and Params are greatly reduced, and mAP@0.5 is improved to a certain extent, up 0.6%. We analyze that the reason for the improvement in accuracy while being lightweight is due to the introduction of the DFC attention mechanism in GhostNetV2. The DFC attention mechanism focuses on both global and local information by dynamically adjusting the weights of the convolution kernel to gather information from different locations. This enables the network to capture feature representations with rich contextual information very efficiently without significantly increasing the number of parameters and computational complexity, thereby improving detection accuracy. On this basis, after the improvement of the upsampling operator CARAFE, although the CARAFE operator involves the reorganization and weighting operation of features, resulting in the increase of the amount of calculation and parameter of the model, mAP@0.5 and mAP@0.5:0.95 are improved by 0.1% and 1.1% respectively. Finally, the CBAM attention mechanism is fused to obtain the highest AP value of fall, reaching 0.846. At this time, although mAP@0.5:0.95 is reduced, mAP@0.5 reaches the highest 0.935. It can be seen from the experimental results that after the fusion of CBAM, the amount of calculation and parameter of the model are reduced compared with those before the fusion, and the Params after fusion reach the lowest 5.10M, and the FLOPs are only 11.2G, which achieves the purpose of high-precision fall detection in complex environments with the best effect. These data illustrate that the CBAM module can help reduce the number of parameters by adaptively recalibrating the feature map so that the model can focus on important features and reduce redundancy. By integrating the attention mechanism, it also allows the model to efficiently allocate computing resources by focusing on the rel-

Table 5
Comparison of ablation experiments

Ghost	CARAFE	C3CBAM	AP			mAP@0.5	mAP@0.5:0.95	FLOPs (G)	Params (M)
			stand	fall	sit				
			0.989	0.823	0.958	0.923	0.676	15.8	7.02
√			0.99	0.824	0.973	0.929	0.67	11.1	5.27
√	√		0.99	0.82	0.979	0.93	0.681	11.4	5.41
√	√	√	0.987	0.846	0.972	0.935	0.676	11.2	5.09

evant parts of the input, which can save computing resources during inference without compromising detection performance.

The training visualization parameters of YOLO-GCC are shown in Figure 9. Box Loss (box_loss) is used to measure the position accuracy of the target box prediction, that is, the difference between the position of the target box and the position of the real box. Objectness Loss (obj_loss) is used to measure whether the target is correctly detected, that is, whether the target exists and its confidence. Class Loss (cls_loss) is used to measure the class classification of the target, that is, the category to which the target belongs. Train/loss means the mean loss in the training set, and val/loss means the mean loss in the validation set. Ideally, we hope that the training loss and validation loss are relatively small, and the difference between them is small, which means that the model can not only fit the training data, but also have good generalization ability.

6.2. Visual Comparison of Detection Effect of Three Types of Tags Before and After Improvement

It can be seen from Table 5 that, compared with the AP data of each label of YOLOv5s and YOLOv5s-GCC before and after the improvement, the AP of stand label decreased by 0.2%, but still reached 0.987, indicating that the accuracy of stand recognition before and after the improvement was very high, with little difference. For the slight drop in accuracy on the stand label caused by our improvement, our analysis is that the addition of attention mechanism often improves the overall performance of the model, but in some specific cases, it may have a slight impact on one or several labels. This is because for labels that already have a high accuracy, the model may have learned enough features for this particular task without using attention. Introducing attention risks making the model focus too much on certain features and ignoring other equally important information. Since our fall detection is not particularly strict for standing, the impact on this non-critical task may be negligible, so this slight drop in accuracy may not have a significant negative impact on the final actual results. However, the AP of fall label and sit label increased by 2.3% and 1.4% before and after the improvement, respectively, indicating that the model's recognition ability of fall and sit before and after the improvement was greatly improved. The comparison of the detection results is as follows:

As can be seen from Figure 10, in different scene environments, the improved model YOLOv5s-GCC has al-

Figure 9

The training visualization parameters of YOLO-GCC

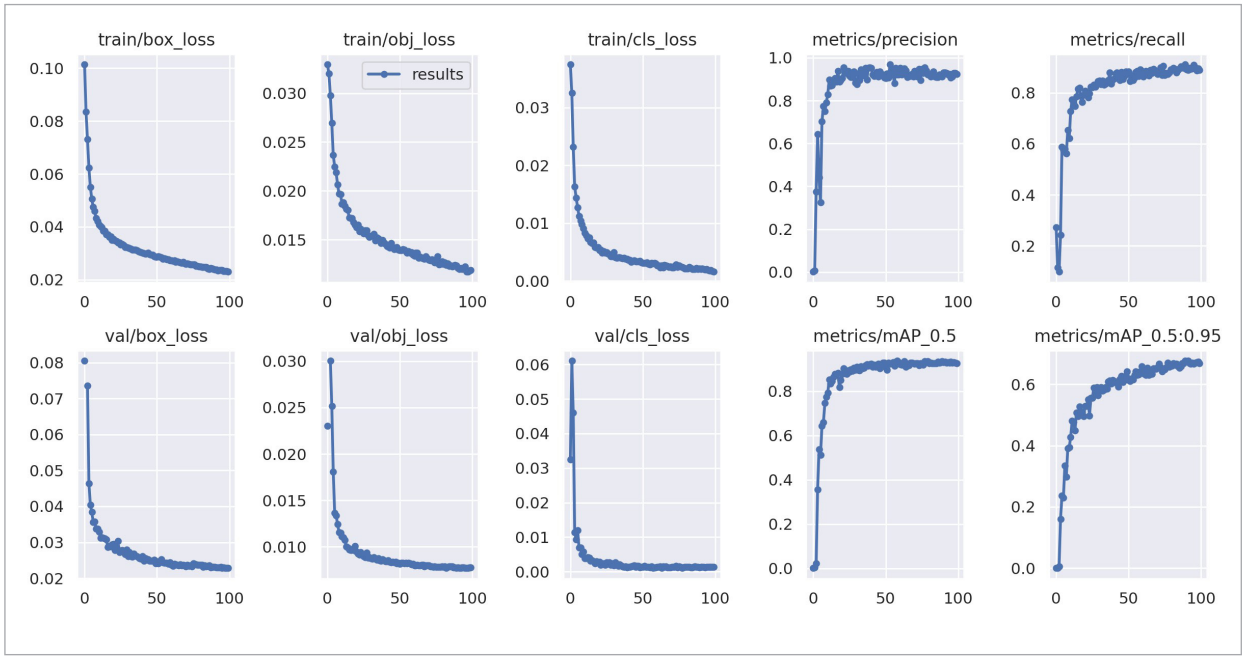


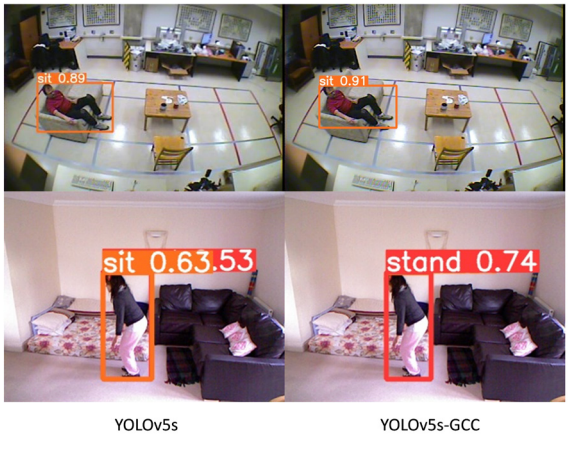
Figure 10
Comparison of stand test results before and after improvement

most the same stand recognition ability as YOLOv5s, and both maintain high recognition accuracy.

As can be seen from Figure 11, in different scenarios, the improved model YOLOv5s-GCC can greatly improve the recognition ability of fall compared with YOLOv5s. In the first scenario, the detection confidence of fall by YOLOv5s is 0.79, and that of fall by YOLOv5s-GCC is 0.84. In the second scenario, the detection confidence of fall by YOLOv5s is 0.81, and that of fall by YOLOv5s-GCC is 0.91. The improved model has better detection effect on fall than the original model.

Figure 11
Comparison of fall test results before and after improvement

Figure 12
Comparison of sit test results before and after improvement



As can be seen from Figure 12, in different scenarios, the improved model YOLOv5s-GCC has improved the recognition ability of sit compared with YOLOv5s. In the first scenario, the detection confidence of sit by YOLOv5s is 0.89, and that of sit by YOLOv5s-GCC is 0.91. In the second scenario, since the person is in the stand state and about to fall, it should not be detected as sit, but the detection confidence of sit by YOLOv5s is 0.63, and that of stand is 0.53, while that of stand by YOLOv5s-GCC is only 0.74. The improved model is better than the original model in the detection effect of sit.

6.3. Comparative Experiments

Compared with traditional models, YOLO series has higher detection accuracy and speed. In order to verify the effectiveness of improving the performance of the model, we trained YOLOv3 and YOLOv3-tiny algorithms on the same dataset under the same training parameters, for comparison with YOLOv5s-GCC. YOLOv3-tiny is a lightweight version of the YOLOv3 model, designed for scenarios that require higher speed and smaller model size. The comparison results are shown in Table 6.

As can be seen from Table 6, YOLOv5s-GCC has the highest mAP@0.5, 1.3% higher than YOLOv3 and 2.2% higher than YOLOv3-tiny; the weight size is the smallest, only 10.1M, 91.8% lower than YOLOv3 and 42.3% lower than YOLOv3-tiny; the calculation amount is the smallest, only 11.2G, 92.8% lower than

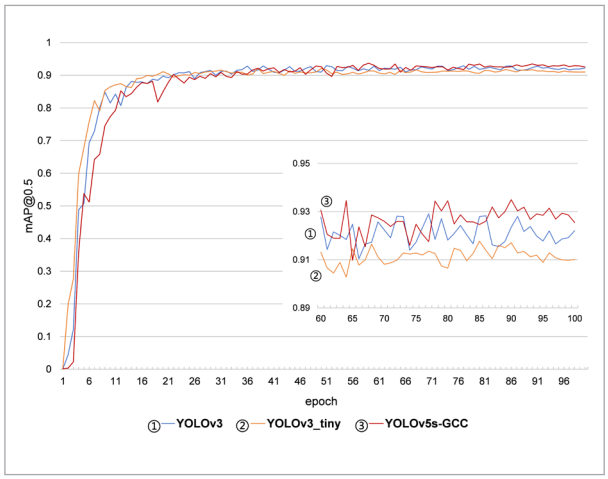
YOLOv3 and 13.2% lower than YOLOv3-tiny. It indicates that the improved model YOLOv5s-GCC has higher detection accuracy and faster speed for fall detection, and the weight file is smaller, which is conducive to deployment on other hardware platforms.

Table 6
Comparative experiments

Model	mAP@0.5	Weights(M)	FLOPs(G)
YOLOv3	0.922	123.6	154.6
YOLOv3-tiny	0.913	17.5	12.9
YOLOv5s-GCC	0.935	10.1	11.2

The comparison of mAP@0.5 curve during training is as follows:

Figure 13
mAP@0.5 change curve comparison



As can be seen from Figure 13, the mAP@0.5 of YOLOv5s-GCC is higher than that of the other two models in most of the last 40 rounds of training, reflecting the effectiveness of the improvement.

7. Conclusion

In order to solve the problem that the elderly fall indoors and cannot be found in time, this paper proposes a fall detection model for the elderly based on improved YOLOv5s. Compared with YOLOv5s, the improved YOLOv5s-GCC has 1.2% higher mAP@0.5, reaching 0.935; 29.1% lower FLOPs, which is reduced to 11.2G; 27.5% lower Params, which is reduced to 5.09M. While significantly reducing the amount of calculation and parameters of the model, it effectively improves the detection accuracy, and provides a new idea for computer vision to help indoor fall detection.

The follow-up work can be carried out around 3D indoor fall detection, such as building a spatial coordinate system to achieve more accurate fall detection through the change of human joint spatial coordinates. In the development and evaluation of fall detection models, it is necessary to ensure that the privacy of participants is protected, and any data collection and use must comply with ethical standards. We believe that privacy protection technologies can be explored, such as differential privacy mechanism and data desensitization technology, to reduce the infringement of personal privacy.

Appendix A

The download addresses of the four datasets used in this article are as follows:

UR Fall Detection Dataset: http:// fenix.univ.rzeszow.pl/~mkepski/ds/uf.html

Fall Detection Dataset (2017 IAPR MVA Conference): http://falldataset.com/

Multiple Cameras Fall Dataset: https:// www.iro.umontreal.ca/~labimage/Dataset/

COCO Dataset: https://cocodataset.org/#home

Data Sharing Agreement

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Ethical Permit

The authors declare that they have no conflict of interest. All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and national research committee. This article does not contain any studies with animals performed by any of the authors. Informed consent was obtained from all individual participants included in the study.

Funding

This article was supported by the National Innovation and Entrepreneurship Training Program for College Students in China (202310225215).

References

1. Bochkovskiy, A., Wang, C. Y., Liao, H. Y. M. YOLOv4: Optimal Speed and Accuracy of Object Detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE/CVF, 2020, 1544-1552. arXiv preprint arXiv:2004.10934
2. China National Bureau of Statistics. Bulletin of the Seventh National Population Census (No.5). Beijing, China, 2021.
3. Dai, J., Li, Y., He, K. M., Sun, J. R-FCN: Object Detection via Region-Based Fully Convolutional Networks. Proceedings of the 30th International Conference on Neural Information Processing Systems (NeurIPS), MIT Press, Barcelona, Spain, 2016, 379-387. arXiv preprint arXiv:1605.06409
4. Dang, J. W., Wei, Y., Liu, N. Survey Report on the Living Conditions of China's Urban and Rural Older Persons. Social Sciences Academic Press (China), 2018, 5-9.
5. Girshick, R. Fast R-CNN. Proceedings of the IEEE Conference on International Conference on Computer Vision (ICCV), IEEE, Santiago, Chile, 2015, 1440-1448. https://doi.org/10.1109/ICCV.2015.169
6. Girshick, R., Donahue, J., Darrell, T., Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Columbus, OH, USA, 2014, 580-587. https://doi.org/10.1109/CVPR.2014.81
7. Han, K., Wang, Y. H., Tian, Q., Guo, J. Y., Xu, C. J., Xu, C. GhostNet: More Features From Cheap Operations. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE/CVF, 2020, 1580-1589. https://doi.org/10.1109/CVPR42600.2020.00165
8. Howard, A., Sandler, M., Chen, B., Wang, W. J., Chen, L. C., Tan, M. X., Chu, G., Vasudevan, V., Zhu, Y. K., Pang, R. M., Adam, H., Le, Q. Searching for MobileNetV3. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), IEEE/CVF, Seoul, South Korea, 2019, 1314-1324. https://doi.org/10.1109/ICCV.2019.00140
9. Howard, A., Zhu, M. L., Chen, B., Kalenichenko, D., Wang, W. J., Weyand, T., Andreetto, M., Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, 2017. arXiv preprint arXiv:1704.04861
10. Hu, J., Shen, L., Sun, G. Squeeze-and-Excitation Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Salt Lake City, UT, USA, 2018, 7132-7141. https://doi.org/10.1109/CVPR.2018.00745
11. Krizhevsky, A., Sutskever, I., Hinton, G. E. Image-Net Classification with Deep Conv-olutional Neural Networks. Communications of the ACM, 2017, 60(6), 84-90. https://doi.org/10.1145/3065386
12. Li, Y., Ma, R., Zhang, R. T., Cheng, Y. F., Dong, C. W. A Tea Buds Counting Method Based on YOLOV5 and Kalman Filter Tracking Algorithm. Plant Phenomics, 2023, 5(1), 0030. https://doi.org/10.34133/plantphenomics.0030
13. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., Berg, A. C. SSD: Single Shot Multibox Detector. Proceedings of the European Conference on Computer Vision (ECCV), Springer, Amsterdam, The Netherlands, 2016, 21-37. https://doi.org/10.1007/978-3-319-46448-0_2
14. Ma, N. N., Zhang, X. Y., Zheng, H. T., Sun, J. ShuffleNetV2: Practical Guidelines for Efficient CNN Architecture Design. Proceedings of the European Conference on Computer Vision (ECCV), 2018, 116-131. https://doi.org/10.1007/ 978-3-030-01264-9_8
15. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, 2016, 779-788. https://doi.org/10.1109/CVPR.2016.91
16. Redmon, J., Farhadi, A. YOLO9000: Better, Faster, Stronger. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Honolulu, HI, USA, 2017, 7263-7271. https://doi.org/10.1109/CVPR.2017.690
17. Redmon, J., Farhadi, A. YOLOv3: An Incremental Improvement. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Salt Lake City, UT, USA, 2018, 7263-7271. arXiv preprint arXiv:1804.02767
18. Ren, S. Q., He, K. M., Girshick, R., Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6), 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031
19. Sandler, M., Howard, A., Zhu, M. L., Zhmoginov, A., Chen, L. C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Salt Lake City, UT, USA, 2018, 4510-4520. https://doi.org/10.1109/CVPR.2018.00474

20. Shen, G. X., Zhao, B. F., Chen, X. J., Liu, L. S., Wei, Y., Yin, T. R. Human Fall Detection Based on Re-Parameterization and Feature Enhancement. IEEE Access, 2023, 11, 133591-133606. https://doi.org/10.1109/ACCESS.2023.3335833

21. Sweta, P., Raju, U. S. N. MS-ML-SNYOLOv3: A Robust Lightweight Modification of Squeeze-Net Based YOLOv3 for Pedestrian Detection. Optik, 2022, 260(1), 1-12. https://doi.org/10.1016/j.ijleo.2022.169061

22. Tang, Y. H., Han, K., Guo, J. Y., Xu, C., Xu, C., Wang, Y. H. GhostNetV2: Enhance Cheap Operation with Long-Range Attention. Proceedings of the 35th International Conference on Neural Information Processing Systems (NeurIPS), MIT Press, New Orleans, LA, USA, 2022, 35, 9969-9982. arXiv preprint arXiv: 2211.12905

23. Wang, J., Chen, K., Xu, R., Liu, Z. W., Chen, C. L., Lin, D. H. Carafe: Content-Aware Reassembly of Features. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), IEEE/CVF, Seoul, South Korea, 2019, 3007-3015. https://doi.org/10.1109/ICCV.2019.00310

24. Wang, X., Jia, K. Human Fall Detection Algorithm Based on YOLOv3. Proceedings of the 2020 IEEE 5th International Conference on Image, Vision and Computing (ICIVC), IEEE, Beijing, China, 2020, 50-54. https://doi.org/10.1109/ICIVC50857.2020.9177447

25. Yang, F., Zhou, J., Chen, Y., Liao, J., Yang, M. X. MSF-YOLO: A Multi-Scale Features Fusion-Based Method for Small Object Detection. Multimedia Tools and Applications, 2024. https://doi.org/10.1007/s11042-023-17818-0

26. Zhang, X. Y., Zhou, X. Y., Lin, M. X., Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Salt Lake City, UT, USA, 2018, 6848-6856. https://doi.org/10.1109/CVPR.2018.00716

27. Zhao, D. A., Song, T., Gao, J., Li, D., Niu, Y. C. YOLO-Fall: A Novel Convolutional Neural Network Model for Fall Detection in Open Spaces. IEEE Access, 2024, 12, 26137-26149. https://doi.org/10.1109/ACCESS.2024.3362958

28. Zhao, Z. Z., Dong, Y. R., Cao, H., Cao, B. Research Status of Elderly Fall Detection Algorithms. Computer Engineering and Applications, 2022, 58(5), 50-65. https://doi.org/ 10.3778/j.issn.1002-8331.2109-0393