scientific reports

OPEN

A YOLOv8 algorithm for safety helmet wearing detection in complex environment

Chunning Song & Yinzhong Li

Helmets are the most common protective equipment on construction sites and can effectively reduce head injuries caused by falling objects. In actual helmet detection on construction sites, traditional target detection faces challenges such as complex environments and unclear target identification. To address this issue, we developed the URD-YOLOv8 helmet detection algorithm. The algorithm is based on an improvement of YOLOv8 and aims to enhance the performance of helmet detection. First, the upsampling module is integrated into the neck network of the model, which makes the model upsampling ability improved and make the image more detailed, thus preventing information loss. Second, a novel convolution module is proposed to help the network focus more on important feature information and improve the effectiveness of the model in feature extraction. Finally, propose a new structure of information aggregation, It better fuses information about target characteristics and context at different scales, allowing the information to flow between channels, thus improving the algorithm's performance. Experiments show that the precision, recall, mAP@0.5 and mAP@0.5:0.95 of the improved helmet wear detection algorithm are higher than the original algorithm by 1.07%, 0.58%, 1.18% and 0.95, respectively.

Keywords YOLOv8, Safety helmet, Targeted detection, Upsampling, Information aggregation

The rapid development of modern society has led to the widespread construction of infrastructure. Safety management has become a critical area of concern in the advancement of the construction, heavy industry, and other sectors. In both construction sites and the industrial sector, the majority of accidents occur in workplaces lacking adequate protective equipment[1]. Among these accidents, they are often caused by the staff's low awareness of safety protection and failure to wear helmets. Therefore, the helmet is an important protective tool for the staff. Currently, helmet supervision is mainly manual. However, due to the extensive area and high volume of personnel on construction sites, manual supervision is inefficient, error-prone, and requires significant manpower and financial resources. Therefore, applying computer vision for the automatic helmet detection is beneficial. Monitoring helmet usage with detection algorithms has significant practical value. This approach not only saves manpower and financial resources but also reduces the risk of safety accidents and enhances the personal safety of workers.

With the development of deep learning, computer vision has received wide attention in fields such as agricultural production[2], machine manufacturing[3], medical analysis[4], transportation[5] and unmanned aerial vehicles (UAV)[6]. In recent years, many studies have focused on helmet detection algorithms. In general, target detection techniques can be broadly classified into two main categories: those based on traditional machine learning methods and those based on deep learning strategies. Light variations or occlusions of target objects can negatively impact the generalization ability of traditional algorithms. Therefore, the performance of traditional algorithms in real construction sites is often suboptimal, although they have shown some improvement. Specifically, Park et al.[7] combined histograms of oriented gradients (HOG) with HSV colour histograms and used this method to detect the target object. Shrestha et al.[8] applied edge detection to the extracted target features to find the helmet's contour information.

Although traditional target detection algorithms are somewhat effective in detecting helmet wear, and play a role in safety management within the construction industry, there still face issues such as weak generalization[9]. A key breakthrough in deep learning-based object detection is the Region-Based Convolutional Neural Network (R-CNN), introduced by Ross Girshick et al.[10]. R-CNN marked a significant advancement in object detection by using a region proposal network (RPN) to extract potential object regions from images, which are subsequently classified and refined through a convolutional neural network (CNN). This architecture notably improved the

School of Electrical Engineering, Guangxi University, Nanning 530004, Guangxi, China. email: scn206@gxu.edu.cn

accuracy and speed of object detection, laying the foundation for more advanced models in computer vision. Subsequently the two-stage method Faster R-CNN proposed by Ren et al.[11], which further improved the R-CNN approach. Then Cai et al.[12] proposed Cascade R-CNN algorithm and so on. In their study[13], Qin et al. proposed an improved version of the faster R-CNN algorithm aimed at enhancing target detection performance. And by adopting the DarkNet53 model architecture, the gradient vanishing problem caused by the increase of network depth is effectively mitigated. Li et al.[14] combined the improved FAST R-CNN and adaptive Canny algorithm to increase the recognition accuracy of ECT. The Inception-v4 architecture is added to the FAST R-CNN structure and used as a convolutional layer for better defect detection.

Although the two-stage target detection algorithm has been improved relative to the traditional target detection algorithm, the problems such as long training time and slow detection speed still exist. The YOLO (You Only Look Once) family of algorithms[15–20], first introduced by Redmon et al. in their work[15] published in 2015, started the YOLO algorithm, It addresses the issues of the two-stage target detection algorithm. The YOLO series has reached YOLOv12, yet YOLOv8 remains widely used in industrial deployments due to its optimal balance between accuracy and speed. For the above mentioned target detection algorithms, the YOLO algorithm has significant advantages. Firstly, YOLO's detection speed is fast. Secondly, YOLO can detect the whole image at the same time to avoid false alarms caused by background errors. Third, YOLO can learn highly generalised features and is suitable for transfer learning. However, the YOLO algorithm still faces challenges in detecting multi-scale small targets such as safety helmets. To address this issue, Zhang et al.[21] addressed the issue of small target detection with multi-scale defects and proposed a network for this purpose. First, a feature enhancement method is introduced to improve the feature information. Second, a novel PAN structure is designed, and finally, a feature conversion module is developed utilizing an attention mechanism.

For the helmet detection task, the following issues must be addressed. Since helmets vary in shape and size across different construction sites, the algorithm requires a high degree of generalization. Additionally, helmets are often partially obscured on construction sites. Consequently, this presents a significant challenge to the algorithm's detection capabilities. To address these challenges, a helmet detection model based on an improved YOLOv8 is designed in this study. This model enhances the generalization ability and detection performance, addressing problems such as the difficulty of targeting algorithms to detect small helmets in complex backgrounds. The main contributions of this paper are as follows:

1. The DySample upsampling operator is introduced and integrated into the model. This module effectively enhances the algorithm's detection accuracy and optimizing its upsampling efficacy.
2. The C2f-RFA module is proposed as a replacement for certain convolutional modules in the model. This module allows the network to better focus on critical feature information by overcoming the limitations of parameter sharing. Thereby significantly enhancing its feature extraction capability.
3. The CSP-EDPAN module is proposed and integrated into the algorithm, alleviating gradient vanishing while effectively capturing multi-dimensional feature information, establishing the dependency relationship between them, and enhancing network performance.

Related work

YOLOv8 algorithm

YOLO algorithms demonstrate high detection accuracy in target detection tasks and outperform other algorithms in detection speed, making the YOLO family widely applicable across various fields. YOLOv8 has advantages such as a lightweight structure, which makes the model faster to train and better at detecting inference. YOLOv8 is available in five versions: n, s, m, l, and x. The YOLOv8 algorithm consists of the following components: data input, backbone network, neck network, and head. Since datasets typically contain images with varying aspect ratios, the YOLO algorithm preprocesses the input images and adaptively scales them to a specified size, improving data reading speed and model training efficiency.

The backbone network primarily extract features from the input image. It consists of three main modules: standard convolution with SiLU activation (CBS), C2f, and Spatial Pyramid Pooling Fast (SPPF). The CBS module serves as the convolutional block of the algorithm and consists of three sub-modules, i.e., Conv2d, SiLU activation function and BatchNorm2d. The C2f module integrates the ELAN component[20] from YOLOv7 with the C3 module, aiming to enable YOLOv8 to effectively enrich gradient information while maintaining a lightweight model. The SPPF module use of multiple connected small pool cores in the SPP module, replacing the use of a single large core. The role of this module can convert feature maps of different sizes into fixed-size feature maps and fuse the information of these features.

The neck network design incorporates two architecturees, the Feature Pyramid Network[22] and the Path Aggregation Network[23], aiming to improve the efficiency of multi-scale fusion and feature delivery. The FPN networks merge low-resolution features with high-resolution features to generate semantically rich feature maps. PAN accelerates the flow of feature information between networks, fusing and disseminating multi-scale feature information more efficiently.

In the YOLOv8 algorithm, adopts an efficient head structure, configured with three detection layers to detect feature maps at different scales, this design fully exploits the multi-scale features to enhance the model's ability to detect targets of different sizes. By employing this multi-scale strategy, YOLOv8 significantly enhance the detection accuracy of the model while maintaining efficient computation.

Upper sampler

Feature upsampling is a crucial step in recovering feature resolution for the model. Since the backbone network typically outputs multi-scale features, upsampling low-resolution features to higher resolution is necessary. An effective upsampler can significantly enhance model predictions. The mainstream upsamplers commonly used in algorithms are Nearest Neighbour or Bilinear Interpolation, which interpolate low-resolution feature maps according to certain rules, which often ignores the semantic meaning in the feature maps. Therefore, this paper uses dynamic upsampling[24] to solve the above problem.

For the sample point generator, the upsampling scale factor is set to 0.25, and the feature map X with channel number C is passed through a linear layer to generate an offset of size 2 s^2× H × W, which is then transformed by pixel shuffling[25] to 2 × sH × sW. Its structure is shown in Fig. 1. The sample set S is the sum of the offset O and the original sampling grid G, i.e.,

$$O = 0.25linear\left(X\right) \tag{1}$$

$$S = G + O \tag{2}$$

Thus, an upsampled feature map X′ can be generated by taking the feature map to be upsampled and using Eq. 3:

$$X' = grid.sample\left(X, S\right) \tag{3}$$

RFAConv

Due to the presence of parameter sharing, standard convolution may not be able to adequately distinguish subtle differences in features at different locations, which in turn will lead to poorer performance of the model. To address these issues, the Receptive-Field Attention (RFA)[26] not only focuses on spatial feature extraction in the receptive field, but also considers the variability of each feature within the receptive field by introducing attention weights for the convolutional kernel to enable finer selection and processing of the input information. By multiplying each element in the receptive field slider with the corresponding attentional weights and then acting with the convolution kernel as the new convolution kernel parameter, this strategy effectively solves the traditional convolution parameter sharing problem. For this reason, Receptive field attention convolution (RFAConv) has been developed using RFA, which can significantly improve the network performance. The structure of the standard RFAConv module is shown in Fig. 2.

In the design of RFAConv, firstly, grouped convolution is adopted to capture the spatial features of the receptive field. For the input data X, it is shaped as $R^{C\times H\times W}$, where C, H and W represent the number of channels, height and width, respectively. After a specific unfolding operation, the dimension of X is converted to 9C × H × W. AvgPool is used to summarise the global feature information within each receptive field. Next, inter-feature interactions and information fusion are achieved by applying a convolution operation with a 1 × 1 convolution kernel. Finally, a softmax function is employed to highlight the relative importance of each feature among the receptive field features. To summarise, the operation of Receptive Field Attention Convolution (RFA) can be expressed in the following way:

$$\begin{aligned} F = Soft\max\left(g^{1\times1}\left(AvgPool\left(X\right)\right)\right) \\ \times \mathrm{ReLU}\left(Norm\left(g^{k\times k}\left(X\right)\right)\right) = A_{rf} \times F_{rf} \end{aligned} \tag{4}$$

In this formulation, Norm represents the process of normalisation the data, $g^{i\times i}$ refers to a grouped convolution operation, where i × i represents the specific size of the convolution kernel, and X refers to the input feature map. And F is the result obtained by multiplying the attention map A_{rf} with the transformed receptive field space feature F_{rf}.

Dual path aggregation module

The strategy of using 1 × 1 convolutional kernel can effectively reduce the computational burden and complexity of the algorithm, which enable cross-channel information fusion, and enhance the network's nonlinear representation. The deeply separable convolution is proposed in MobileNetV1 network. The module is implemented by decomposing the standard convolution operation into two parts: pointwise convolution and depthwise convolution. Zhang et al.[27] introduced a group convolution strategy aimed at reducing the

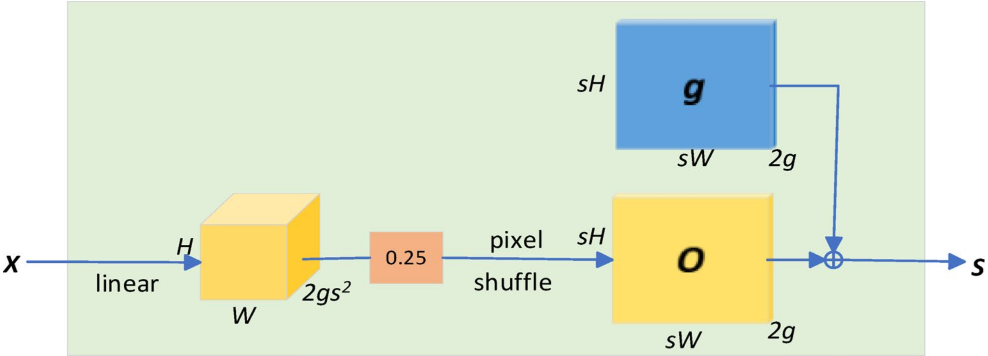


Fig. 1. Sampling point generator in DySample.

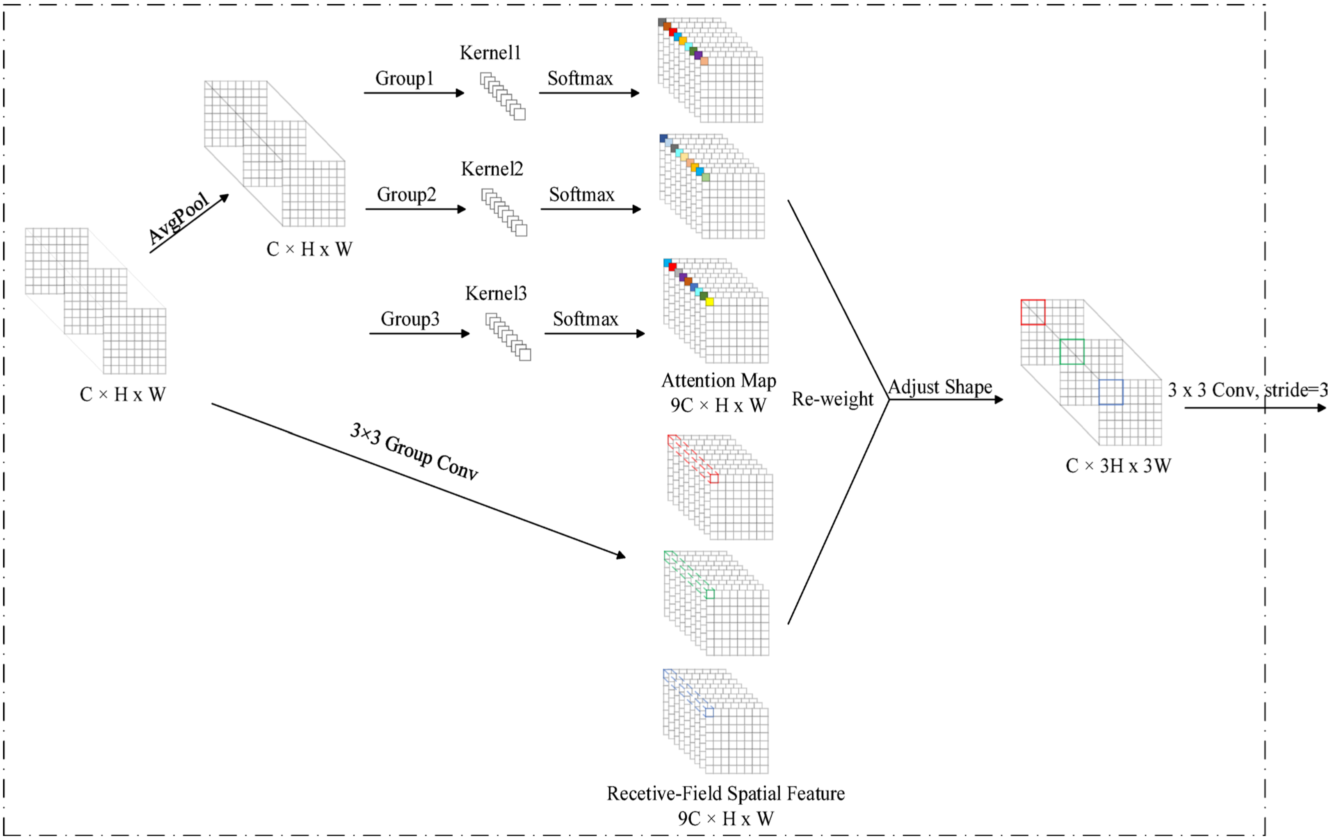


Fig. 2. The detailed structure of RFAConv.

computational overhead of the network. On the other hand, Singh et al.[28] further explores the balance between efficiency and performance by using a combination of convolution kernels of different sizes to form a heterogeneous convolution (HetConv).

Influenced by GroupConv and HetConv, Zhong et al.[29] combined 1 × 1 pointwise convolution with 3 × 3 group convolution to propose the module Dual Path Aggregation module, which can replace standard convolution. By integrating 1 × 1 pointwise convolution, this module aims to preserve the initial information of the input feature map, thus ensuring that deeper convolution filters continue to extract the original information from the feature map.

Figure 3 illustrates the specific structure of the module, with respect to the parameters: while G specifies the number of groups used in the group convolution operation, N denotes the number of channels of the output feature map, and M represents the number of channels of the input feature map. The orange convolution filter size is 1 × 1 and green convolution filter size is 3 × 3. The 1 × 1 convolution acts on all channels of the input feature map for computation, while the 3 × 3 convolution operation slides over the depth direction of the feature map (i.e., the channel dimension) to process the channel information.

Methods

DySample module

In this section, an improvement has been made to the YOLOv8 algorithm. The original standard upsampling module was replaced with the DySample module to optimise model performance. The static range factor is set to 0.25, and Group is experimentally set to 4. By applying the DySample module to the algorithm, the up-sampler can be guided to improve the quality of the boundaries by finding the correct semantic information for each up-sampling point more efficiently. This module effectively enhances the model's up-sampling quality. The details of its implementation in the YOLOv8 architecture are visualised in Fig. 4, where the improvements are highlighted in red.

C2f-RFA module

The RFAConv module is replaced with C2f-RFA module. Applying the C2f-RFA module to the backbone and neck networks, and replacing some convolutional operations with the RFAConv module, the network obtains a significant enhancement. This improvement not only strengthens the backbone's ability to extract image features but also enhances the network's focus on and recognition of key information and features. Figure 5a clearly shows the internal structural design of the RFABottleneck module, while Fig. 5b shows the construction details of the modified C2f-RFA module.

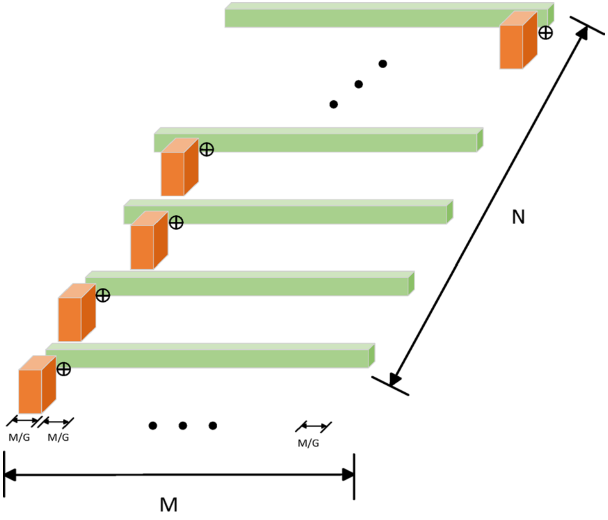


Fig. 3. Dual Path Aggregation module structure diagram.

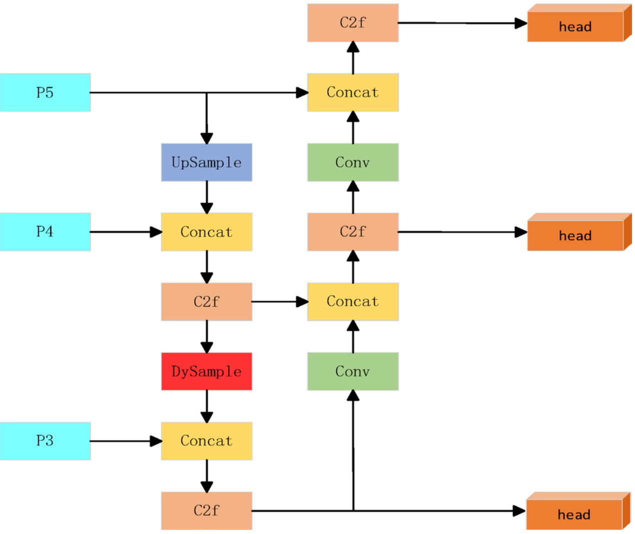


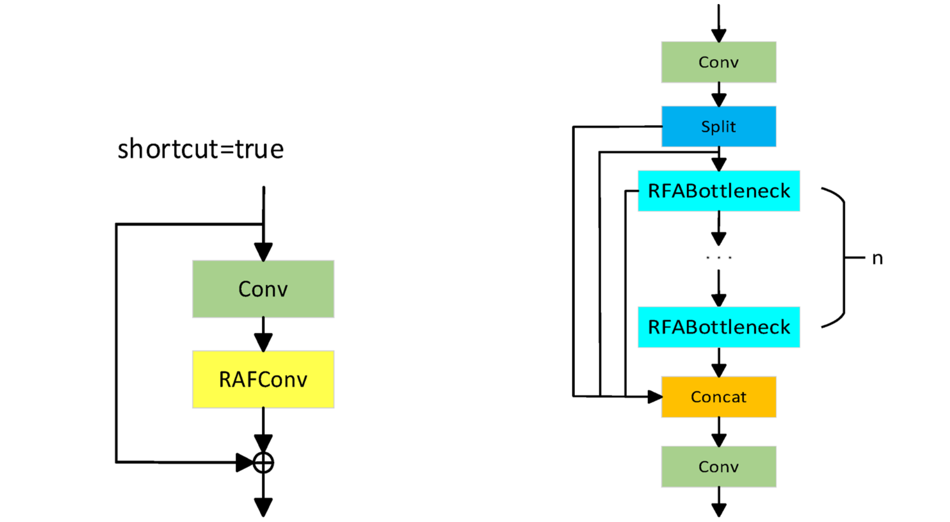
Fig. 4. The specific location of the DySample insertion.

CSP-EDPAN module

The CSP Efficient Dual Path Aggregation Network module (CSP-EDPAN) uses 1 × 1 convolution and 3 × 3 convolution as the basis of the modular structure, incorporating ideas from ResNet and grouped convolution to build an efficient network structure. The 1 × 1 convolution in the module not only preserves the integrity of the input information, but also promotes the effective fusion and interaction of local and global features, providing rich feature information for deeper convolutional layers. By applying the module to the neck network of the model, it not only increases the width and depth of the network and mitigates the gradient vanishing, but also effectively captures the feature information of different dimensions and establishes the dependency relationship between the dimensions, so as to provide high-quality feature information for the subsequent convolution operation. Figure 6 shows the structure diagram of the module.

Proposed algorithm

In this study, we innovatively design an improved algorithm based on YOLOv8, with the core innovations being: in the neck network part, the original up-sampling module is optimally replaced with the DySample module; at the same time, part of the C2f module is replaced by the more powerful C2f-RFA module; and the neck of the algorithm further integrates the CSP-EDPAN module to enhance the network performance. The complete structure of the algorithm is presented in detail in Fig. 7, where the improved modules are highlighted in red to visually indicate the algorithm's enhancements.



(a) ResNet-RFABottleneck module structure diagram. (b) C2f-RFA module structure diagram.

Fig. 5. Network structure diagram of the ResNet-RFABottleneck module and C2f-RFA module.

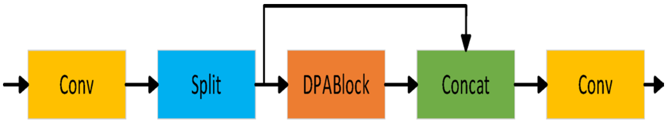


Fig. 6. CSP-EDPAN module structure diagram.

Experimentation

Description of the data set

The helmet detection dataset used for the experiments in this study is SHWD. The dataset, consisting of 7581 images, contains a wide range of examples of helmet and head detection in diverse scenes, lighting conditions, size scales, and different degrees of target occlusion, providing a rich sample resource for the study. The dataset was labelled using the LabelImg annotation software to label the objects in the images. The images included 9,044 labelled helmet targets and 111,515 labelled normal head targets. The dataset follows a 7:2:1 random split ratio. Specifically, it is divided into three major subsets: the training set contains 5306 images for model learning; the validation set has 1516 images for evaluating the model's performance during training; and the test set consists of 759 images for the final test of the model's generalisation ability.

Experimental environment

In this study, the same hyperparameters are used for training all the proposed models. Although fine-tuning the hyperparameters can improve the model's detection performance, the central focus of this paper is on designing and implementing a new target detection model, rather than tuning hyperparameters. The hyperparameter settings for the model in this paper are shown in Table 1. In this study, the PyTorch deep learning framework was used to conduct the relevant experiments. Training was performed on NVIDIA GeForce RTX 4080 GPU. The configuration details are outlined in Table 2.

The size of the input image during the training phase is directly related to the performance of the algorithm and its operational efficiency. Although high-resolution image inputs enhance recognition accuracy, they also inevitably slow down the training process. The YOLO family of algorithms aims to enhance the consistency of training data and optimise the model training process by uniformly scaling the input images to a preset size. The image mosaic technique, a data enhancement strategy, is incorporated into the YOLO algorithm system as an important hyperparametric means to improve model robustness and diversity. By using the Mosaic method, the diversity and complexity of the dataset can be effectively enhanced, which in turn improves the model's generalisation ability. The optimiser is a tool to guide the updating of network parameters, and the YOLO algorithm usually uses the stochastic gradient descent (SGD) method to update the gradient. To ensure that the model training process can effectively jump out of the region of local optimal values, 0.937 is used as the momentum value; meanwhile, to prevent the model from overfitting, the weight decay coefficient is set to 0.0005. The learning rate indicates the size of the optimisation algorithm's weight updates and affects the stability of training. Higher learning rates accelerate training, but also introduce oscillations. Therefore, the model weights are obtained after 300 training epochs, with the starting learning rate to 0.01 and the terminating learning rate to 0.01.

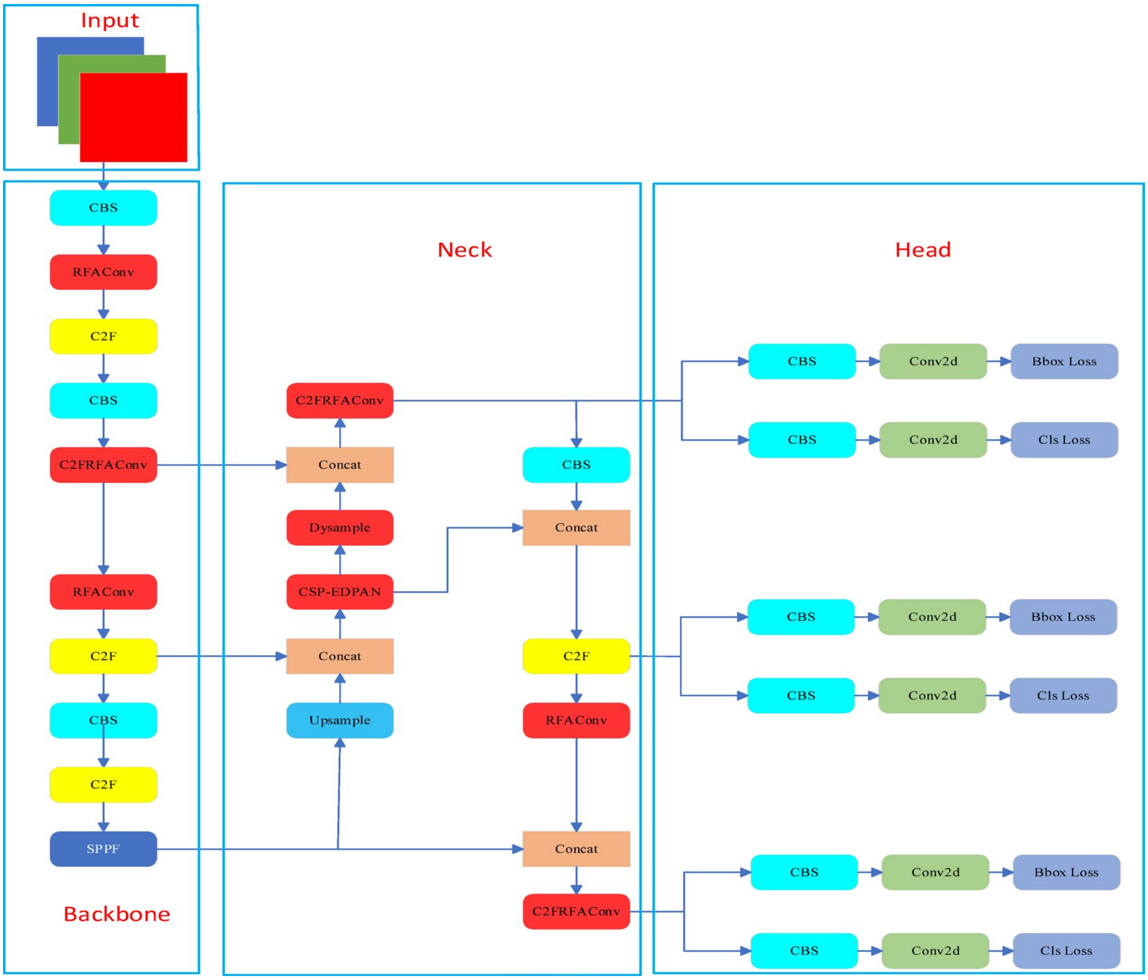


Fig. 7. The proposed algorithm architecture.

Parameter	Configuration
Image size	640 × 640
Initial learning rate	0.01
Final learning rate	0.01
Batch size	16
Optimizer	SGD
Epoch	300

Table 1. Hyperparameter settings of network training.

Configuration	Details
GPU	NVIDIA GeForce RTX4080
Operating systems	Windows 11
Memory	32 GB RAM

Table 2. Experimental configuration.

Analysis of experimental results

Experimental evaluation indicators

Model evaluation is an important task that measures the excellence of an algorithm. Evaluation metrics usually include precision, recall, mean accuracy, frames per second, model size and GFLOPs. The precision is a measure of the algorithm's ability to correctly predict positive class samples. Recall is defined as the ratio of correctly identified positive class samples to the total number of true positive samples. Mean Average Precision (mAP) is one of the performance criteria for evaluating algorithms. This metric is calculating by averaging the detection accuracy for each category after setting the IoU threshold. To accurately and comprehensively assess the detection capability of the algorithms, evaluation metrics are adopted in this study: mean average precision (mAP), precision (P), recall (R), frames per second (FPS), model size and GFLOPs. These metrics provide a comprehensive view of the performance and capabilities of the detection algorithm from different perspectives.

$$\Pr ecision = \frac{TP}{TP + FP} \tag{5}$$

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

TP, FP, and FN in Eq. are represented in binary confusion matrices. Where TP (True Positives) refers to the number of actual positive samples that correctly identified as positive; FP (False Positives) denotes the number of negative samples but were incorrectly labelled as positive; and FN (False Negatives) refers to the number of positive samples that were not correctly identified.

Average Precision (AP) is a comprehensive metric that measures model performance by integrating and averaging the model's precision at different confidence thresholds. The mean average precision (mAP), on the other hand, is the average of the integral area under the Precision-Recall curve (i.e., the AP value for each category) under all categories, thus reflecting the average detection precision of the model across categories.

$$AP = \int_{0}^{1} P\left(r\right) dr \tag{7}$$

$$mAP = \frac{\sum\limits_{i=1}^{k} AP_i}{k} \tag{8}$$

K is the number of categories in the dataset.

FPS is an important performance metric for how fast an algorithm processes video or image streams. FPS represents the number of frames processed per second, reflecting the real-time capability of the algorithm.

The size of the model is also extremely important, and smaller models usually have faster reasoning speeds and are more suitable to deploy on resource-constrained devices (such as edge devices, mobile devices).

Analysis of training result plots

Figure 8a shows the mAP curve of the proposed improved algorithm. As can be seen from the figure, the algorithm achieves an AP of 93.2% for detecting targets in the "people" category, and an AP of 89.8% for the detection of "hats". The mAP for all target categories is 91.5%. Figure 8b illustrates the confusion matrix of the algorithm, where the rows of the matrix represent the true category labels and the columns correspond to the predicted categories as a way of depicting in detail the correct and incorrect classification results. Observation of the confusion matrix shows that most of the target predictions are accurate, this shows that the algorithmic improvements proposed in this study exhibit superior performance in the helmet detection task.

Ablation experiments

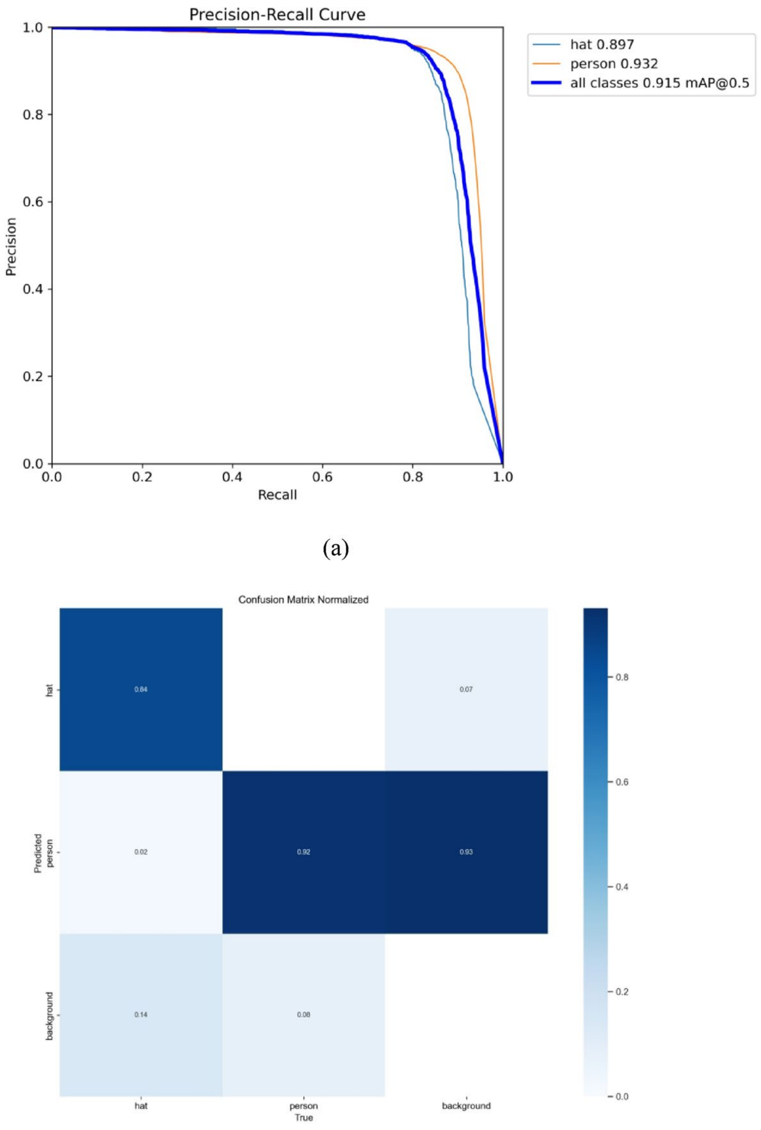
To confirm that the proposed improved module indeed enhances the performance of the YOLOv8 algorithm, we implemented detailed ablation experiments for verification. YOLOv8n was used as the baseline model, with the DySample module, C2f-RFA module, and CSP-EDPAN module were added sequentially.

The precision, recall and mAP@0.5, were used as metrics for the ablation study to validate the effectiveness of each module. The original YOLOv8n algorithm has 91.40%, 84.88% and 90.33% for precision, recall and mAP@0.5, respectively. When the DySample module was added, the algorithm evaluation metrics were 91.58%, 84.93% and 90.69%. Where precision rose 0.05% and recall rose 0.36%, mAP@0.5 increased by 0.32%. After integrating the C2f-RFA module, the algorithm achieves 92.72%, 84.44% and 91.18% for each individual performance metric. The accuracy was improved by 1.32%, mAP@0.5 by 0.85%. When the CSP-EDPAN module is added, the indicators are 92.08%, 84.45% and 90.56%. This includes a 0.68% increase in accuracy and a 0.23% increase in mAP@0.5. Finally, all the improved modules are added to the model and the individual metrics of the proposed algorithm are 92.47%, 85.46% and 91.51%. All of its evaluation indicators have improved. Table 3 shows the results of the ablation experiments.

Comparative experiments

In order to verify the performance of the algorithms in this study, we conducted a comparison experiment. Various algorithms including Faster R-CNN, SSD and RetinaNet were selected as references. The experiments show that the proposed improved algorithm has a mAP value of 91.51%, and that the proposed algorithm exhibits significant advantages in performance, and the specific comparative advantages are detailed in Table 4.

For the YOLO series of algorithms, performance metrics are compared using YOLOv3, YOLOv4, YOLOv5, YOLOv7, YOLOv8, YOLOv10 and YOLOv11. Table 5 specifically shows the results of the comparison experiments of this algorithm with other algorithms in the YOLO family. From the analysis in the table, it can be



(a)

(b)

Fig. 8. (**a**) Mean average precision graph. (**b**) Confusion Matrix.

DySample	C2f-RFA	CSP-EDPAN	Precision (%)	Recall (%)	mAP@0.5 (%)
			91.40	84.88	90.33
√			91.58	84.93	90.69
	√		92.72	84.44	91.18
		√	92.08	84.45	90.56
√	√	√	**92.47**	**85.46**	**91.51**

Table 3. Ablation experiment. The best performance parameters are in bold.

seen that the proposed algorithm in this paper surpasses most of the compared algorithms in terms of evaluation metrics.

Visualisation results

The data used in this paper is public and has been deposited on GitHub at https://github.com/njvisionpower/Safety-Helmet-Wearing-Dataset. The visualization images were derived from the above datasets. Figure 9 shows the visualisation of the effectiveness of the helmet detection task. The column on the left is the results of the benchmark model, and on the right is the results of the improved model. In the presence of target occlusion (a),

Method	AP (%) hat	AP (%) person	mAP@0.5 (%)
Faster R-CNN	88.91	52.76	70.83
SSD[30]	83.40	55.39	69.40
RetinaNet[31]	**93.48**	84.70	89.09
Prop	89.79	**93.24**	**91.51**

Table 4. Comparative experiments with other series of algorithms. The best performance parameters are in bold.

Method	P (%)	R (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Weight (MB)	FPS	GFLOPs
YOLOv3	89.47	72.48	79.23	47.66	16.6	115.1	12.9
YOLOv4	90.80	83.02	88.27	54.10	100	84.4	118.9
YOLOv5	91.09	84.40	90.50	53.38	13.7	**126**	15.8
YOLOv7	92.28	**86.44**	91.22	57.27	71.3	92.86	103.2
YOLOv8	91.40	84.88	90.33	56.95	5.95	117	8.1
YOLOv10[32]	90.60	85.86	91.47	57.73	5.48	112	6.5
YOLOv11[33]	91.10	85.19	**91.65**	57.46	**5.23**	97	**6.3**
Prop	**92.47**	85.46	91.51	**57.90**	5.99	109	8.3

Table 5. Comparison experiment. The best performance parameters are in bold.

The original model failed to detect the staff dressed in red clothes, while the improved model solved this missed detection problem; In dense scenarios (b), the original model has serious misdetection in the intensive detection scenario, The model incorrectly identified background elements and workers' torsos as helmet-wearing workers, but reduces false detection in the improved model; In factory environments with blurred backgrounds (c), the original model mistakenly detected two distant blurry objects as helmet-wearing workers when processing blurred images. The improved model solves this problem very well.

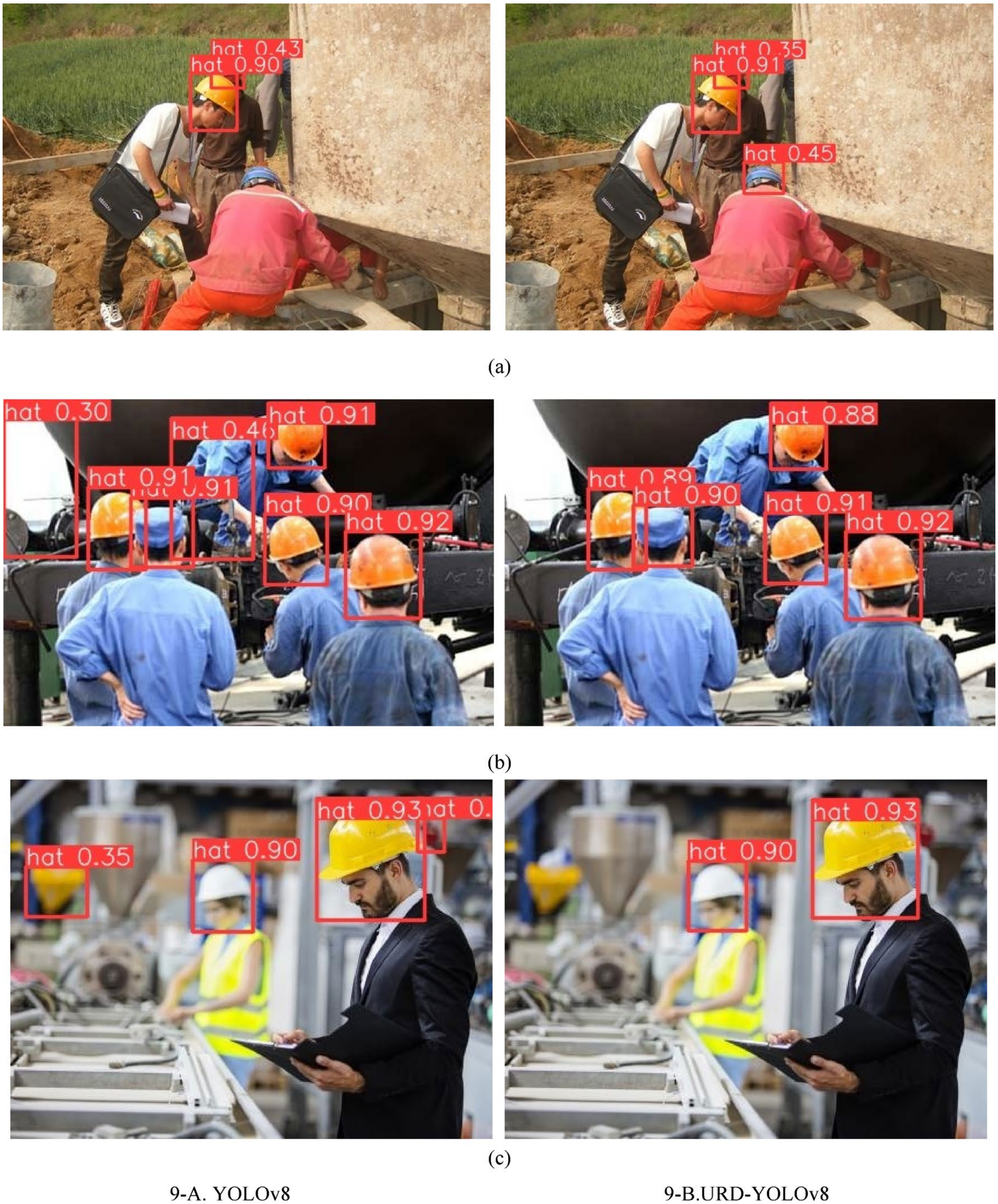
The visualisation results show that the algorithm studied in this paper can accurately identify the target features with good detection results in the complex and diverse detection scenarios.

Conclusion

For the task of helmet wear detection at construction sites, in this study, a helmet wearing detection algorithm based on the YOLOv8 framework is innovatively designed and implemented. Compared with other algorithms, the algorithm shows a significant advantage in performance.

The contributions of this paper are threefold. Firstly, the up-sampling of the model is optimised by introducing the DySample module in the neck structure of the model. The DySample module nicely improves the up-sampling quality of the network, allowing the network to obtain feature maps with distinct boundaries, which in turn improves the effectiveness of the subsequent sampling of the convolutional layers. Secondly, we upgraded the original C2f module in the network with the C2f-RFA module, which not only enhances the network's efficacy in feature extraction, but also optimises the network's ability to focus on the detected target, effectively mitigates the limitations of parameter sharing in traditional convolutional layers. Finally, the CSP-EDPAN module is proposed and applied to the neck of the algorithm. Its better fusion of multi-scale features of the target and background information allows the information to flow between channels, thus improving the performance of the algorithm. Experiments show that the algorithm proposed in this paper has significant advantages in helmet wearing detection, its precision, recall, mAP@0.5 and mAP@0.5:0.95 are 1.07%, 0.58%, 1.18% and 0.95 higher than the original algorithm respectively.

In our ongoing work, we plan to lightweight the model and deploy it to edge devices for real-time detection. Our goal is to apply the proposed model to the practical production and life, addressing the challenge of detecting safety helmets in complex environments.



(a)

(b)

(c)

9-A. YOLOv8

9-B.URD-YOLOv8

Fig. 9. Visualisation results.

Data availability

The data used in this paper is public and has been deposited on GitHub at https://github.com/njvisionpower/Safety-Helmet-Wearing-Dataset. The datasets used and analysed during the current study available from the corresponding author on reasonable request.

Received: 13 November 2024; Accepted: 24 June 2025
Published online: 07 July 2025

References

1. Lee, J. Y., Choi, W. S. & Choi, S. H. Verification and performance comparison of CNN-based algorithms for two-step helmet-wearing detection. *Expert. Syst. Appl.* **225**, 120096. https://doi.org/10.1016/j.eswa.2023.120096 (2023).

2. Tian, Y. et al. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Comput. Electron. Agric.* **157**, 417–426. https://doi.org/10.1016/j.compag.2019.01.012 (2019).
3. Li, W. et al. Deep learning based online metallic surface defect detection method for wire and arc additive manufacturing. *Robot. Comput-Integr. Manuf.* **80**, 102470. https://doi.org/10.1016/j.rcim.2022.102470 (2023).
4. Priyanka, S., Baranwal, N., Singh, K. N. & Singh, A. K. YOLO-based ROI selection for joint encryption and compression of medical images with reconstruction through super-resolution network. *Future. Gener. Comput. Syst.* **150**, 1–9. https://doi.org/10.1016/j.future.2023.08.018 (2024).
5. Kang, L., Lu, Z., Meng, L. & Gao, Z. YOLO-FA: Type-1 fuzzy attention based YOLO detector for vehicle detection. *Expert. Syst. Appl.* **237**, 121209. https://doi.org/10.1016/j.eswa.2023.121209 (2024).
6. Sadykova, D., Pernebayeva, D., Bagheri, M. & James, A. IN-YOLO: Real-time detection of outdoor high voltage insulators using UAV imaging. *IEEE Trans. Power. Deliv.* **35**(3), 1599–1601. https://doi.org/10.1109/TPWRD.2019.2944741 (2020).
7. Park, M. W. & Brilakis, I. Construction worker detection in video frames for initializing vision trackers. *Automat. Constr.* **28**, 15–25. https://doi.org/10.1016/j.autcon.2012.06.001 (2012).
8. Shrestha, K., Shrestha, P. P., Bajracharya, D. & Yfantis, E. A. Hard-hat detection for construction safety visualization. *Constr. Eng.* **2015**, 1–8. https://doi.org/10.1155/2015/721380 (2015).
9. Shen, J. et al. Detecting safety helmetwearing on construction sites with bounding-box regression and deep transfer learning. *Comput. Aided Civ. Infrastruct. Eng.* **36**(2), 180–196. https://doi.org/10.1111/mice.12579 (2021).
10. Girshick, R., Donahue, J., Darrell, T. & Malik, J. Rich feature hierarchies for accurate object det-ection and semantic segmentation (2013). http://arxiv.org/abs/1311.2524.
11. Ren, S., He, K., Girshick, R. & Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks (2015). http://arxiv.org/abs/1506.01497.
12. Cai, Z. & Vasconcelos, N. Cascade R-CNN: Delving into high quality object detection (2017). Preprint at http://arxiv.org/abs/1712.00726.
13. Qin, H. et al. An improved faster R-CNN method for landslide detection in remote sensing images. *J. Geovis. Spat. Anal.* https://doi.org/10.1007/s41651-023-00163-z (2024).
14. Li, J., Zhang, L. & Zheng, W. Improved faster R-CNN and adaptive Canny algorithm for defect detection using eddy current thermography. *AIP Adv.* https://doi.org/10.1063/5.0189084 (2024).
15. Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. You only look once: Unified, real-time object detection (2015). http://arxiv.org/abs/1506.02640.
16. Redmon, J. & Farhadi, A. YOLO9000: Better, Faster, Stronger (2016). Preprint at http://arxiv.org/abs/1612.08242.
17. Tian, Y. et al. Apple detection during different growth stages in orchards using the improved YO-LO-V3 model. *Comput. Electron. Agric.* **157**, 417–426. https://doi.org/10.1016/j.compag.2019.01.012 (2019).
18. Wang, C.-Y., Bochkovskiy, A. & Liao, H.-Y. M. Scaled-YOLOv4: Scaling cross stage partial network (2020). Preprint at http://arxiv.org/abs/2011.08036.
19. Zhang, Y. et al. Complete and accurate holly fruits counting using YOLOX object detection. *Comput. Electron. Agric.* https://doi.org/10.1016/j.compag.2022.107062 (2022).
20. Wang, C.-Y., Bochkovskiy, A. & Liao, H.-Y. M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors (2022). Preprint at http://arxiv.org/abs/2207.02696.
21. Zhang, Y., Zhang, H., Huang, Q., Han, Y. & Zhao, M. DsP-YOLO: An anchor-free network with DsPAN for small object detection of multiscale defects. *Expert Syst. Appl.* **241**, 122669. https://doi.org/10.1016/j.eswa.2023.122669 (2024).
22. Lin, T.-Y. et al. Feature pyramid networks for object detection (2016). Preprint at http://arxiv.org/abs/1612.03144.
23. Liu, S., Qi, L., Qin, H., Shi, J. & Jia, J. Path aggregation network for instance segmentation (2018). Preprint at http://arxiv.org/abs/1803.01534.
24. Liu, W., Lu, H., Fu, H. & Cao, Z. Learning to upsample by learning to sample (2023). Preprint at http://arxiv.org/abs/2308.15085.
25. Shi, W. et al. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network (2016). Preprint at http://arxiv.org/abs/1609.05158.
26. Zhang, X. et al. RFAConv: Innovating spatial attention and standard convolutional operation (2023). Preprint at http://arxiv.org/abs/2304.03198.
27. Zhang, X., Zhou, X., Lin, M. & Sun, J.: Shuffle-Net: An extremely efficient convolutional neural network for mobile devices (2017). Preprint at http://arxiv.org/abs/1707.01083.
28. Singh, P., Verma, V. K., Rai, P. & Namboodiri, V. P. Hetconv: Heterogeneous kernel-based convolutions for deep CNNs. In *Proceedings of the IE-EE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 4830–4839 (IEEE, 2019). https://doi.org/10.1109/CVPR.2019.00497.
29. Zhong, J., Chen, J. & Mian, A. DualConv: DualConvolutional kernels for lightweight deep neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **34**(11), 9528–9535. https://doi.org/10.1109/TNNLS.2022.3151138 (2023).
30. Liu, W. et al. SSD: Single shot multibox detector. In *2016 European Conference on Computer Vision (ECCV)* 21–37 (2016). https://doi.org/10.1007/978-3-319-46448-0_2.
31. Lin, T. Y. et al. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell. Arrow* **42**(2), 318–327. https://doi.org/10.1109/TPAMI.2018.2858826 (2018).
32. Wang, A. et al. YOLOv10: Real-time end-to-end object detection. Preprint at https://arxiv.org/abs/2405.14458.
33. Khanam, R. & Hussain, M. YOLOv11: An overview of the key architectural enhancements. Preprint at https://arxiv.org/abs/2410.17725.

Author contributions

N.S. Review and edited the manuscript, Z.L. conceptualized and wrote the main manuscript text and prepared all figures. All authors reviewed the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Ethical approval

We confirm that all methods have been implemented in accordance with relevant guidelines and regulations. Experimental protocols have been approved by the College of Electrical Engineering at Guangxi University. Informed consent has been obtained from the team and participating members.

Additional information

Correspondence and requests for materials should be addressed to C.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.