
Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2025 SCPE. Volume 26, Issues 3, pp. 1336–1347, DOI 10.12694/scpe.v26i3.4311

RESEARCH ON IDENTIFICATION AND DETECTION OF UNSAFE BEHAVIORS OF CONSTRUCTION WORKERS BASED ON DEEP LEARNING

MEIYU ZHANG*, HONGMING CHEN† AND XUEFENG HAN‡

Abstract. In order to improve the safety management level of construction sites, prevent and reduce the occurrence of building safety accidents, this article uses deep learning methods to study these unsafe behavior recognition and detection techniques. The most typical hazardous behavior is not wearing a safety helmet. However, on-site personnel often neglect to wear helmets due to various reasons. In this study, the target detection algorithm is applied to monitor helmet-wearing. The YOLOX algorithm is selected as the basic detection model and improved by combining the construction site environment and helmet detection characteristics, meeting the real-time monitoring needs of helmet-wearing. Comparison experiments before and after improvement were conducted on the self-constructed helmet dataset, verifying the performance of the improved YOLOX network model. The results show that the average accuracy of the enhanced network model on the helmet-wearing dataset increased to 89.12%, showing a better detection effect.

Key words: deep learning, YOLOX algorithm, sensory field, unsafe behaviour, target detection, building construction, behavior recognition

1. Introduction. Currently, the safety situation in building construction in China remains a serious concern, as the frequency of safety accidents continues to be high, resulting in elevated numbers of incidents and fatalities. [1]. The majority of these accidents are a result of unsafe behaviors exhibited by construction workers. These workers often engage in long hours of high-intensity physical labor, leading to fatigue and subsequently lazy and risky behaviors. Studies have indicated that traumatic brain injuries resulting from falling objects accounted for 24% of total construction worker accidents, with most of these fatal accidents attributed to a lack of safety helmet use [2]. Previous studies have explored the influencing factors, mechanisms, and pre-control methods of construction workers' unsafe behaviors, but have lacked effective strategies for directly controlling and correcting these behaviors [3]. Meanwhile, many scholars have conducted research on helmet-wearing state detection algorithms based on deep learning methods. In 2018, Fang et al [4] tried for the first time to apply the Faster R-CNN algorithm to helmet-wearing detection, and although the algorithm made some progress in improving detection accuracy, it could not meet the real-time requirements. In 2020, Liang et al [5] based on the YOLOv3 algorithm for helmet detection, due to its relatively single dataset, resulting in poor generalization of the model; in 2023, Qi Zezheng et al [6] used a hybrid pooling optimization spatial pyramid pooling (SPP) module (SPP) in the form of tandem pooling in YOLOv5s and embedded a coordinate attention mechanism in the slicing module, although the detection accuracy of the model is improved, the model is more complex and unfavorable for deployment. This paper aims to address the limitations of the aforementioned research by utilizing a deep learning platform to construct the YOLOX (You Only Look Once X) target detection network model. Additionally, the characteristics of the construction site environment are taken into consideration, and a structural reparameterization model is introduced to enhance the overall network's feature expression capability and computing speed. Furthermore, a sliding window transformation network is incorporated to improve the network's field of perception. The decoupled detector head is also further decoupled to enhance the model's feature extraction ability. The research focuses on developing a method for detecting helmet-wearing among construction personnel, to enable pre-warning, normal detection, and standardized management for construction safety. Two-Stages target detection algorithms based on candidate regions and One-Stage target detection algorithms based on regression are two types of mainstream target detection methods at this stage.

*Nanjing Tech University, China
†Nanjing Tech University, China
‡Nanjing Tech University, China

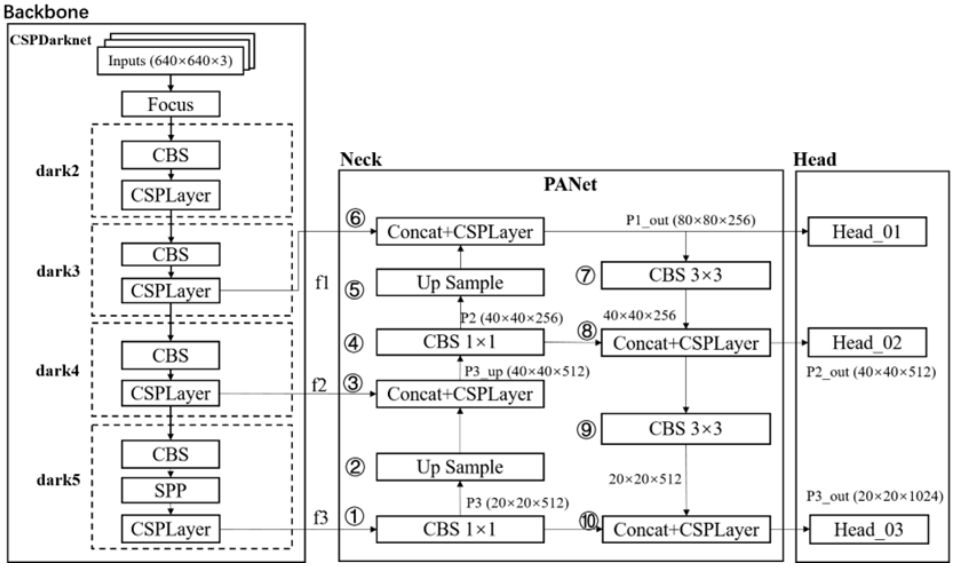


Fig. 2.1: Network structure of YOLOX.

Two-Stages algorithms mainly include R-FCN [7], Faster-RCNN [8], Mask-RCNN [9], and other detection algorithms. These algorithms need to generate candidate regions first, and then classify and localize the targets in the candidate regions; one-stage algorithms mainly include the YOLO (You Only Look Once) series, SSD [10], and other algorithms, which don't generate candidate frames but directly transform the localization problem of the candidate frames into a regression problem to be dealt with, and the whole detection process makes use of the end-to-end (end-to-end) detection of objects. to-end) direct regression of the object's category and location [11]. Compared with the two-stage algorithm, the single-stage algorithm has a faster processing speed and is suitable for real-time application scenarios [12].

2. YOLOX Network Structure. The network structure of the YOLOX algorithm model has three main parts, Backbone, Neck, and Head (as shown in Figure 2.1)[13].The Backbone network is the foundation of its entire architecture, effectively extracting features from input images through a combination of convolutional and pooling layers. These features are crucial for subsequent object detection as they provide basic information and contextual relationships of the image. The Neck section integrates feature information from different dimensions together, and the different focuses of the three outputs are more conducive to detecting targets at different scales. The Head network of YOLOX is the core of the entire object detection model, responsible for generating bounding boxes, category probabilities, and object confidence scores. These prediction results are the final judgment of the model on the target position and category in the input image.

3. Network Modelling Improvement. At this stage, the mainstream target detection methods can be divided into two types: two-stage (Two-Stages) target detection algorithms based on the candidate region and one-stage (One-Stage) target detection algorithms based on regression. From the results of a large number of studies, although the accuracy of the One-Stage target detection algorithm is slightly lower than that of the Two-Stage algorithm under the same circumstances, the detection speed has increased, which can better meet the timeliness of construction site inspection. The task of target detection involves identifying and localizing object information within an image. In the realm of image processing, the focus has shifted towards utilizing deep learning methods over traditional image processing techniques for target detection. When detecting workers wearing safety helmets in a complex construction environment, it is essential to consider the intricacies of the building construction site.

The analysis of site environment characteristics and safety helmet detection is outlined below:

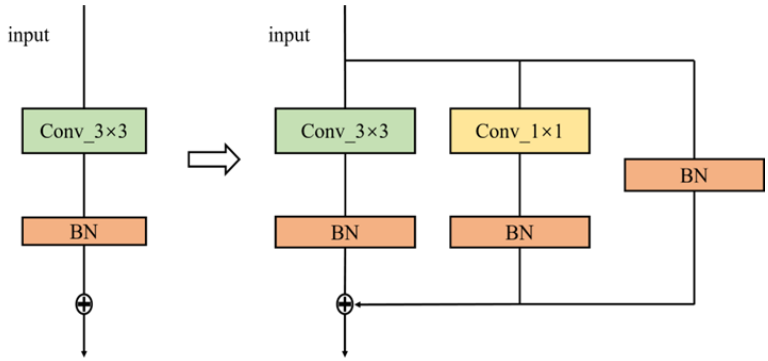


Fig. 4.1: Multi-branch structure of RepVGG.

1. The building construction site has unstable light and cluttered background, and the workers' location distance and occlusion are all uncertain, and these factors will affect the effectiveness of target detection.
2. The helmet target is small, so the detection accuracy of the deep learning target detection algorithm is required to be high. The current deep learning algorithm still has some difficulties in dealing with small targets, high-density targets, and other scenes.

Based on the analysis above, this paper modified the Backbone, Neck, and Head structure of YOLOX to enhance the model's feature extraction capability for detecting workers wearing safety helmets in building construction environments. This modification aims to improve the detection effect and reliability of the model.

4. Introduction of structural reparametric modeling. RepVGG (Re-parameterization VGG) [14], i.e., using the idea of structural re-parameterisation, uses a multi-branch structure to increase the number of parameters that can be computed during network training to improve performance and converts it to a single-path structure during network inference thereby increasing the speed of computation and reducing the memory. RepVGG uses the original VGG network structure as the backbone and makes structural improvements.

Figure 4.1 shows the transformation of the RepVGG-based VGG network into a multi-branch parallel structure during the training phase. It can be seen that the original single-path structure on the left mainly contains Conv_3×3 and BN layers, and the Conv_1×1 residual branch and identity residual branch are introduced on the right. The simple residual structure is transformed into a complex residual structure with the addition of multiple branches, and the network is transformed from a single flow path into multiple flow paths. Training such a network is equivalent to training multiple networks, and thus the parameters that can be computed by the model are greatly increased. This not only enhances the representation ability in the deep network of the model but also solves the problem of vanishing gradient in the deep network, making the network easier to converge. The multi-branch structure increases the number of parameters to be computed and acquires more feature representations, but multiple branches mean that all branches are computed before the next step of fusion, which leads to the inability to make full use of the computational power of the hardware, increasing the amount of computation and reducing the speed. Therefore, although the multi-branch model brings high performance of the deep network, it becomes slower and occupies too much memory, so it is not suitable for applications in industrial scenarios. To solve these problems, RepVGG converts the multi-branch into a single path model in the inference stage and applies the idea of structural reparameterization to perform Op fusion and Op substitution, and Figure 4.2 shows the conversion process of the single path model.

The main process of step 1 in Fig. 4.2 is to fuse the convolutional layers in the residual blocks of each branch with the BN layer and the equivalent replacement convolutional layers of the identity layer. The red box in the figure indicates the fusion of Conv_3×3 with the BN layer, and the yellow box indicates the fusion of Conv_1×1 with the BN layer, the merging of layers can effectively improve the performance, and the fusion process is as follows:

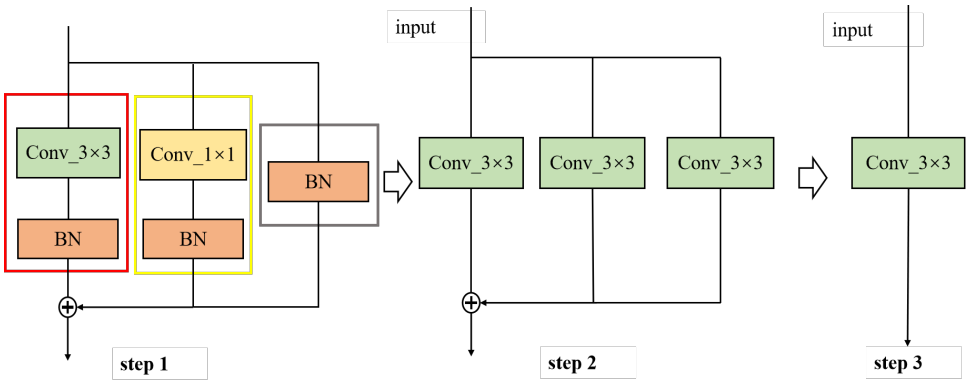


Fig. 4.2: Structure reparameterisation process.

Convolutional Layer Formulation:

$$Conv\left(\ x\ \right) = w * x + b \tag{4.1}$$

BN layer formula:

$$BN\left(\ x\ \right) = \gamma * \frac{x - \mu}{\sqrt{(\sigma)^2 + \epsilon}} + \beta \tag{4.2}$$

Among them, x is an element in the input feature map, w is the convolutional layer parameter, b is the bias term parameter, μ is the sliding mean of the BN layer, σ^2 is the sliding variance of the BN layer, γ and β are the scale factor and offset factor obtained from training and learning, and ϵ represents a minimal constant to avoid the denominator being zero. Since the BN layer is usually located after the convolutional layer, the output of the convolutional layer is used as the input parameter for the BN layer. Substituting (4.1) into (4.2) yields:

$$BN\left(\ Conv\left(\ x\ \right)\ \right) = \gamma * \frac{w * x + b - \mu}{\sqrt{(\sigma)^2 + \epsilon}} + \beta \tag{4.3}$$

Further simplification leads to:

$$BN\left(\ Conv\left(\ x\ \right)\ \right) = \frac{\gamma * w}{\sqrt{(\sigma)^2 + \epsilon}} * x + \left[\frac{\gamma * (b - \mu)}{\sqrt{(\sigma)^2 + \epsilon}} + \beta\right] \tag{4.4}$$

Re-order:

$$\begin{cases} \hat{w} = \frac{\gamma * w}{\sqrt{(\sigma)^2 + \epsilon}} \\ \hat{b} = \frac{\gamma * (b - \mu)}{\sqrt{(\sigma)^2 + \epsilon}} + \beta \end{cases} \tag{4.5}$$

Finally available:

$$BN\left(\ Conv\left(\ x\ \right)\ \right) = \hat{w} * x + \hat{b} \tag{4.6}$$

After fusion, it is transformed into a convolutional layer, where $\hat{w}$ represents the weight parameters of the fused convolutional layer, and $\hat{b}$ represents the bias term parameters of the fused convolutional layer. The

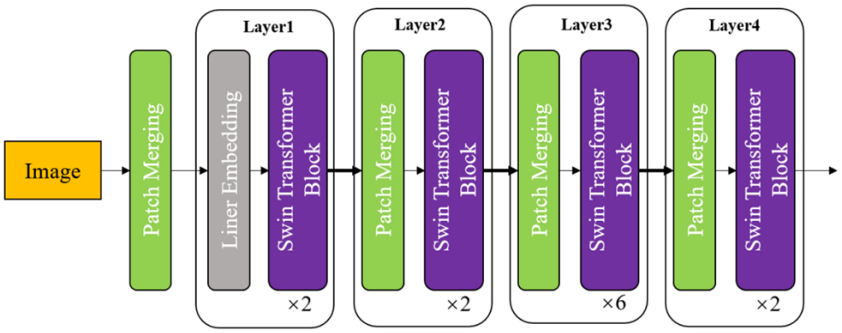


Fig. 4.3: Sliding window transform network structure.

entire process of fusing convolutional layers with BN layers does not increase computational complexity, but only modifies the convolution kernel to speed up the operation.

As shown in the yellow and grey boxes in Fig. 3, there are Conv_1×1 branches and identity branches in addition to the Conv_3×3 branch after fusion. Step 2 is to convert all the convolutions of different convolution kernels of other branches into Conv_3×3. The conversion of Conv_1×1 is mainly to use a matrix transformation to move the values in its convolution kernel to the center of Conv_3×3. Step 3 is to superimpose the convolutional weights and bias parameters of the three branches by using the same convolutional additive principle, and finally, the three branches are fused into a brand new Conv_3×3 single-path network structure.

RepVGG is more capable of enhancing the network's ability to express image features than ordinary convolutional layers for cases such as building construction environment occlusion and small targets at long distances. Worker helmet-wearing monitoring requires RepVGG to improve the detection accuracy while ensuring the detection speed to better adapt to the real-time monitoring needs. In this chapter, RepVGG is used to replace the 3×3 convolution in the Backbone, Neck, and Head parts to achieve the improvement of the overall performance of the network.

4.1. Introduction of the Swin Transformer Structure. Swin Transformer (Shifted Window Transformer) [15], a sliding window transform network, is a deep learning model designed for image recognition that improves on the Transformer and currently achieves state-of-the-art performance in computer target detection and image segmentation tasks. The Swin The general structure of the Transformer is shown in Figure 4.3.

To be closer to the original feature information, this paper chooses to introduce the Swin Transformer structure to replace the three CSPLayer layers similar to the Head in the PANet enhancement network. Compared with the CSPLayer structure, the Swin Transformer layer structure design can easily adjust the depth of the network, expand the sensory field of the network, extract features at different levels in the image, reduce the complexity of the entire PANet reinforcement network, and ultimately improve the target detection efficiency and accuracy, which is also beneficial for industrial real-time object detection or large-scale object detection tasks. This is also beneficial for industrial real-time object detection or large-scale object detection tasks. The improved PANet network with the introduction of Swin Transformer structure is shown in Figure 4.4.

4.2. Further decoupling of the detection header. YOLOX proposes a Decoupled Head for classification and regression tasks respectively, and the two branches are trained independently to achieve greater parallelization and faster convergence. Decoupled Head and Coupled Head have been compared and validated on the COCO dataset, and the accuracy has been significantly improved (see Figure 4.5).

The network can learn deeper features of the target object in the image, further improving the generalization ability of the network. In the helmet-wearing detection task, to further strengthen the network's ability to extract features and improve the network's generalization ability and robustness, this paper decouples the regression branch again and again to balance the positioning accuracy and classification accuracy. Figure 4.6 shows the comparison of the detection head before and after the improvement.

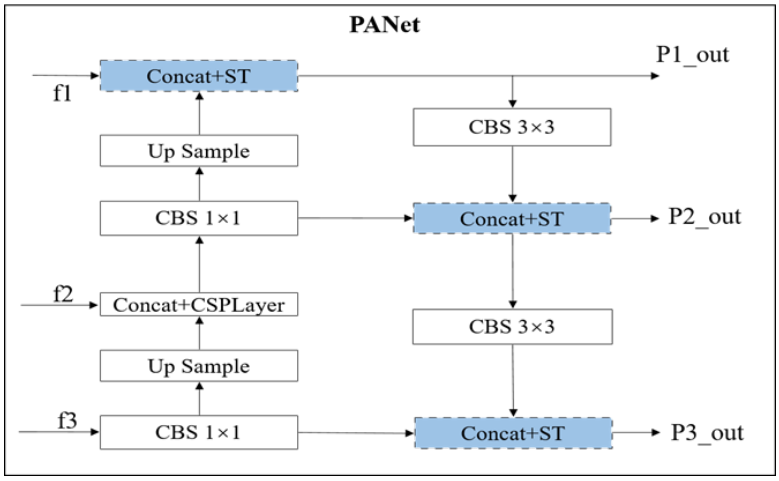


Fig. 4.4: Improved PANet network.

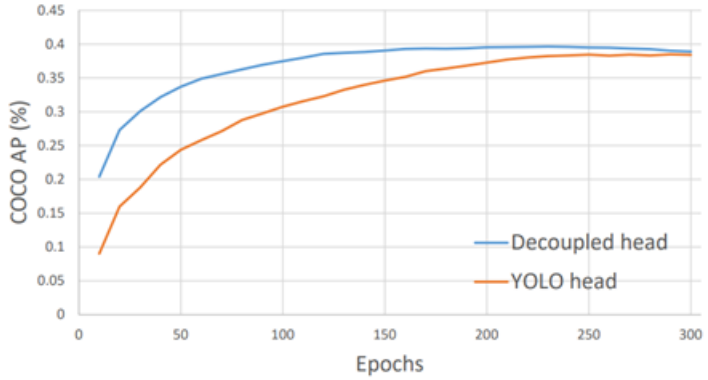


Fig. 4.5: Comparison of detection accuracyDecoupled Head.

4.3. Improved YOLOX network structure. In this paper, we improve on the YOLOX network structure by using the RepVGG structure parameterized model to replace the 3×3 convolutional layers of the three main parts of the Backbone, Neck, and Head, in which the activation function still uses SiLU (Sigmoid Linear Unit); in the PANet network, the Swin Transformer structure is introduced to replace the three CSPLayer layers; for the regression branch of the Decoupled Head detection head, the branch is further decoupled to independently perform the classification task, localization task, and confidence task. The overall network structure of the improved YOLOX is shown in Figure 4.7.

5. Construction of the dataset.

5.1. Data collection and labelling. To construct a dataset of workers wearing helmets in construction environments, images need to be collected, labeled, and numbered so that they meet the detection requirements of the model. One part of the images come from searching the keywords ”construction site helmet wearing” and ”construction workers” on the web, as well as images intercepted from construction videos of construction sites; the other part of the images are obtained by filtering and integrating open-source VOC and other public datasets on the web. publicly available datasets. Finally, these two parts of image data are merged and the detected objects in the images are labeled using the appropriate software. The annotation information of each image will generate a corresponding XML file so that it can meet the requirements of subsequent helmet model

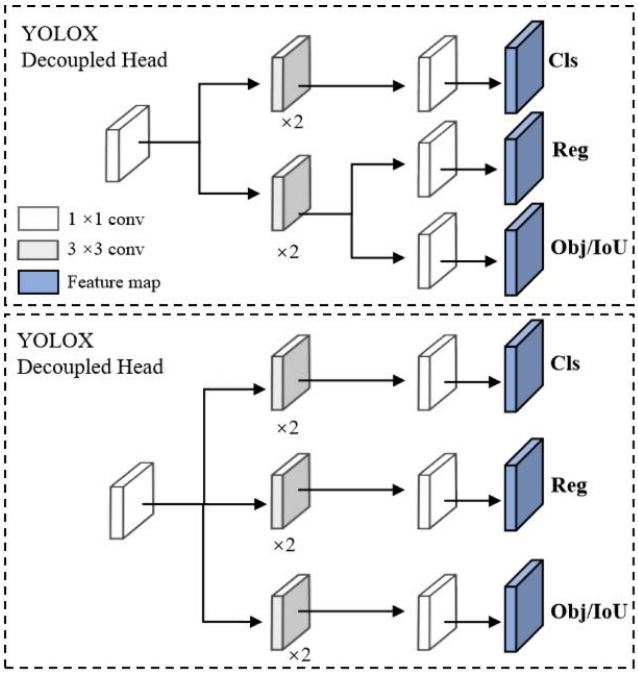


Fig. 4.6: Decoupled Head re-decoupled comparison.

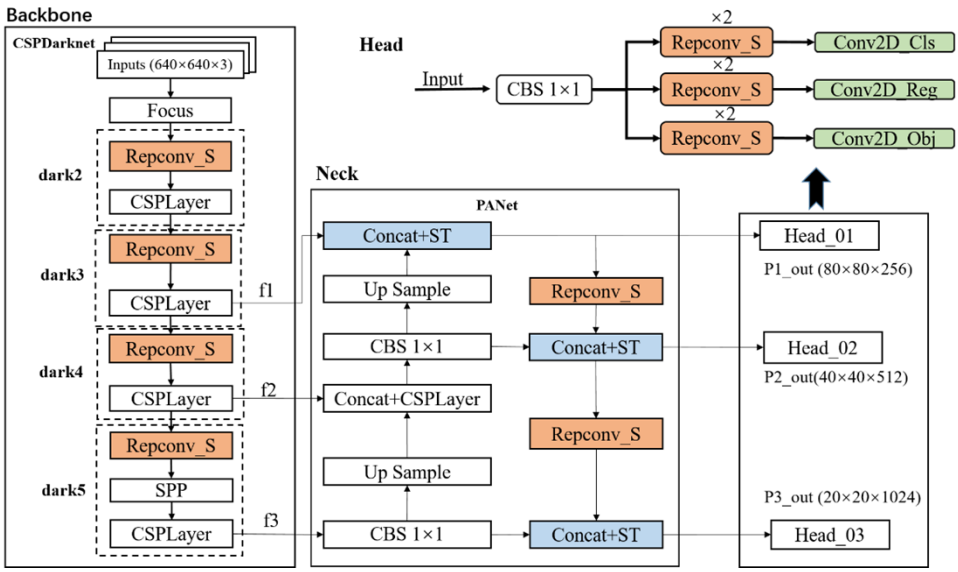


Fig. 4.7: Structure of the improved YOLOX.

training.

In this process, a total of 5973 images are collected, and the dataset is named ”safe_hat”. In the dataset, the original images are stored in the ”JPEGImages” folder, and their corresponding XML files are stored in the ”Annotations” folder. Some of the images in the dataset are shown in Figure 5.1.

In deep learning, the training of the model requires the division of the dataset images, which is generally

Fig. 5.1: Example of a partial image of the dataset.

divided into training set, validation set, and test set. In this paper, 5973 images in the dataset are randomly assigned to the training set and validation set according to the ratio of 8:2, and the test set is not set here because the validation set is not involved in the training, and at the same time can be used as a test set.

5.2. Data Enhancement. In YOLOX, both Mixup and Mosaic data enhancement methods are performed when the dataset is read. The Mixup method i.e., the obfuscation enhancement technique, uses the mixup function to linearly interpolate two different samples in the training dataset to mix them to generate new samples. Specifically for two samples, linear interpolation can be performed on their images, bounding box sizes, and categories respectively. Mosaic method i.e. mosaic enhancement technique, which stitches together four different images to form a new large image, and then randomly crops and scales this large image. Both methods can increase the diversity of the training data in the process of augmenting the data, thus improving the generalization ability and robustness of the model and reducing the risk of overfitting. In addition to these two methods, the model will use other methods for image data enhancement by each part of the structure during the training process.

5.3. Evaluation indicators. In object recognition for target detection, each detection frame can be regarded as a binary classification problem, where positive samples indicate that the detection frame correctly detected the target object and negative samples indicate that the detection frame did not correctly detect the target object. According to the classification results of the detection frame and the actual situation, it can be classified into the following four types:

True Positive (TP): i.e., the number of true cases, (generally set to 0.5) of the detection frame;

False Positive (FP): i.e., the number of detection frames of the false positive example (containing other redundant detection frames of the same true frame);

False Negative (FN): false negative, the number of undetected real boxes;

True Negative (TN): that is, the true negative case, the actual negative samples detected as negative samples.

(1) IOU i.e. Intersection Ratio. It is the ratio of the overlap area of the predicted and labeled boxes to their merged area. The larger the ratio, the more accurate the localization of the target. The image schematic is shown in Figure 5.2.

(2) Precision: i.e. the precision rate. It is the probability that the prediction is a positive sample and the actual sample is also positive, the formula is shown in equation (5.1).

$$Precision = \frac{TP}{TP + FP} \tag{5.1}$$

(3) Recall. It is the probability of being predicted as a positive sample in a sample that is positive, generally the higher the recall, the lower the accuracy, the formula is shown in equation (5.2).

$$Recall = \frac{TN}{TP + FN} \tag{5.2}$$

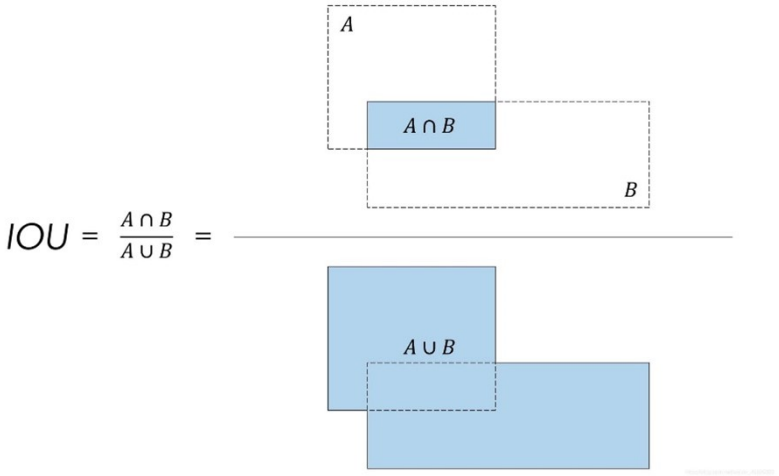


Fig. 5.2: Example of intersection-parallel ratio.

(4) AP: i.e. average precision. It is the area of the P-R curve with and as coordinates of the horizontal and vertical axes.

(5) mAP: i.e., mean average precision. It is the average value of all categories. It is a comprehensive measure to evaluate the comprehensive performance of the whole testing process by considering the two indicators of precision and recall.

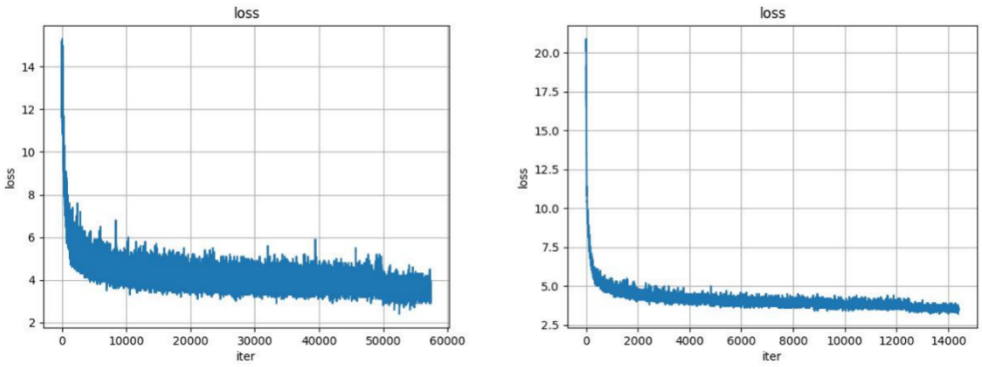
6. Calculation and Analysis.

6.1. Computer Operating Environment. This work is computationally intensive and the algorithmic procedures are mainly accelerated on GPU. The system environment is Ubuntu16.04, the GPU is NVIDIA Tesla V100, 32GB of RAM, PyTorch is used to build the deep learning framework, Python3.6 is used as the programming language, and some of the other acceleration tool libraries are Cuda10.5, Cudnn and so on.

6.2. Calculation Process and Analysis of Results. At the beginning of the training of this model, to make the model better adapt to the dataset, we choose to set the initial learning rate (learning rate) to 0.001, and the value of batch-size (batch-size) to 10, i.e., 10 images samples are selected for each training. The model uses the Adam optimizer, the learning rate decay coefficient is set to 0.9, the weight-decay coefficient (weight-decay) is set to 0.0005, and the confidence threshold is set to 0.5. During the experimental process, iteratively, the learning rate is gradually adjusted to 0.05, and the maximum number of training rounds (max-epoch) is 120, which is reached when the training is automatically stopped. The whole process is validated after the completion of each round of training, dynamically adjusting the model parameters and optimizing the model to avoid overfitting the model on the training set.

During the training process, the value of the loss function is used to present the model as good or bad, and the smaller the loss value, the better the model training. As shown in Figure 6.1, (a) graph represents the convergence process of the loss function during the training process of the original YOLOX model, and (b) graph represents the convergence process of the loss function during the training process of the improved YOLOX model. The horizontal and vertical coordinate values indicate the number of training iterations and the value of the loss function at that number of iterations, respectively.

From Figure 6.1, it can be seen that the improved YOLOX network model performs better with faster convergence and lower loss function values during training. To further evaluate the detection performance of the model, the next step is to analyze the detection effect of the model through the analysis of the P-R curve, in which the performance can be evaluated by the area (AP) enclosed by the curve and the coordinate axis, and the larger the area, the better the performance. Figure 6.2 exhibits the P-R curves obtained from the validation of the YOLOX model on the self-constructed helmet dataset before and after the improvement. As can be seen



(a) (b)

Fig. 6.1: Convergence plot of loss function before and after improvement.

Table 6.1: Comparison of helmet detection results of the model before and after improvement

Algorithmic model	categories	Pre(%)	Recall(%)	mAP(%)	FPS
previous approach:	hat	94.67	87.36	85.56	10
	person	89.47	81.95		
our approach:	hat	94.78	87.77	89.12	11
	person	95.12	87.13		

from the figures, (a) and (b) show the P-R curves of the pre-improved YOLOX model for the categories "hat" and "person", and (c) and (d) show the P-R curves of the improved YOLOX model for the categories "hat" and "person", and (e) and (f) show the P-R curves of the improved YOLOX model for the categories "hat" and "person". (c) and (d) show the P-R curves of the improved YOLOX model for the categories "hat" and "person". The P-R curves of the improved model for both helmet wearers and non-helmet wearers are more biased towards the upper right corner of the coordinate axis and enclose a larger area. From the graphs, it seems that the improved model has improved the detection accuracy for both "hat" and "person" categories. Although the improvement in detection accuracy is small for the case of wearing a helmet, the improvement in detection accuracy for the case of not wearing a helmet is 7.21%, which indicates that the detection accuracy of the improved model has been significantly improved.

The P-R curve shows the goodness of the category detection results, and the final measure of the comprehensive performance of the model, i.e., the goodness of the multiple categories, is still based on the value of mAP. Table 6.1 shows the comparison of the results of the YOLOX algorithm model before and after the improvement after experimentation on the self-constructed helmet dataset.

The results showed that the improved YOLOX model achieved better detection results on the helmet dataset compared to the original YOLOX model. The improved model has a mAP value of 89.12%, which is a 3.56% improvement over the original YOLOX model. In terms of real-time performance, the number of images detected per second by the improved network also increased. This indicates that the improved model can more accurately and quickly detect whether a worker is wearing a helmet or not, and is more suitable for target monitoring tasks in real construction scenarios.

7. Conclusion. In this study, the YOLOX algorithm is used as the basic detection model to achieve automatic identification of construction workers not wearing helmets on supervised construction sites, and the performance of the algorithm is improved and optimized. The improved YOLOX network model improved

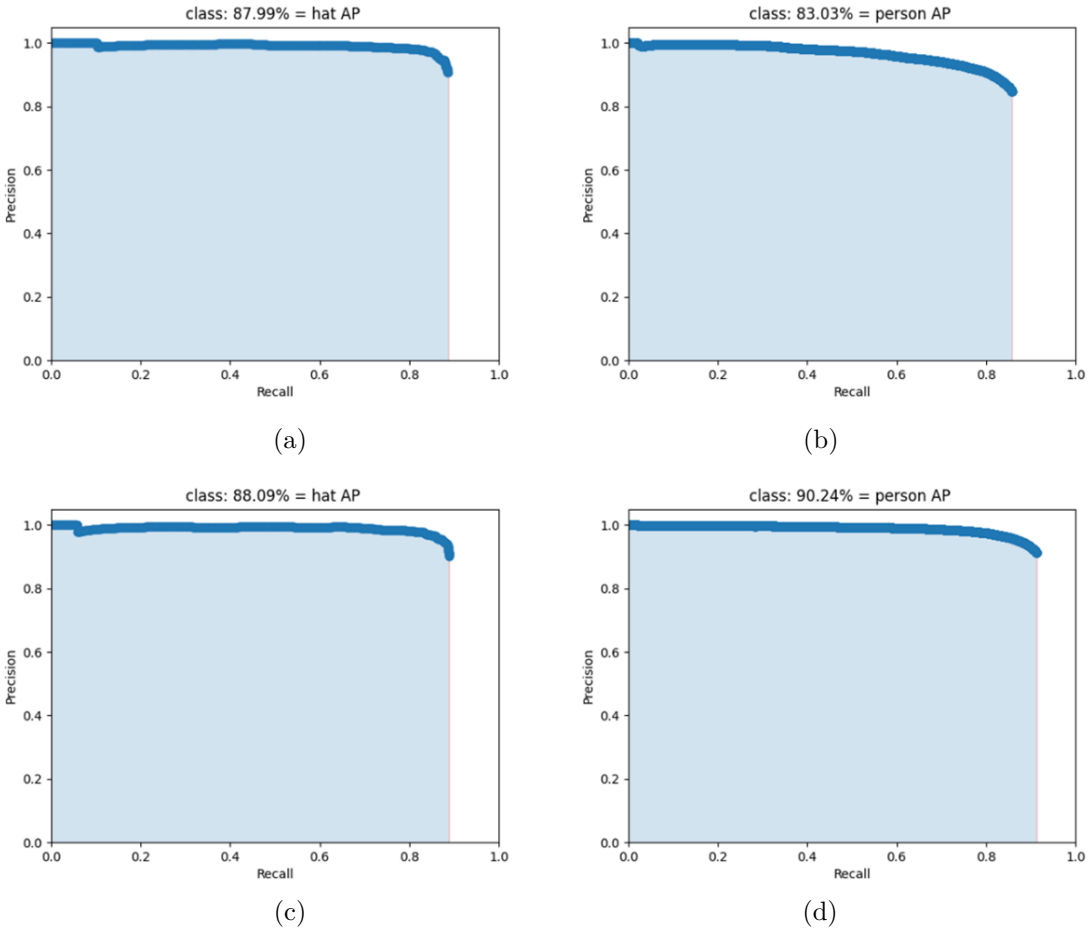


Fig. 6.2: P-R curves for each category before and after improvement.

the detection accuracy of both non-wearing and helmet-wearing personnel over the pre-improved model, with a total mAP improvement of 3.56%. This study demonstrates that the enhanced YOLOX network model exhibits superior performance and reliability in detecting helmet-wearing at construction sites, showcasing significant practical value and potential for widespread application.

REFERENCES

[1] MOHAMMADFAM I,GHASEMI F,KALATPOUR O, ET, *al.Constructing a bayesian network model for improving safety behavior of employees at workplaces[J]*,Applied Ergonomics,2017,58:35- 47.

[2] GOLOVANOV R, VOROTNEV D, KALINA D. , *Combining Hand Detection and Gesture Recognition Algorithms for Minimizing Computational Cost[A]*, 2020 22th International Conference on Digital Signal Processing and its Applications (DSPA)[C]. Moscow, Russia: IEEE, 2020: 1–4.

[3] LIU Y, JIANG W, , *ARTIFICIAL I.Detection of wearing safety helmet for workers based on YOLOv4[J]*, International Conference on Computer Engineering,2021:83-87.

[4] FANG Q, LI H, LUO X, ET, *al. Detecting non-hardhat-use by a deep learning method from far-field surveillance videos[J]*, Automation in construction, 2018, 85: 1-9.

[5] LI Y G,WEI H,HAN Z,ET, LU-*al.Deep learning-based safety helmet detection in engineering management based on convolutional neural networks[J]*, Advances in Civil Engineering,2020(6):1-10.

[6] WANG W,LI Y T,ZOU T,ET, *al.A novel image classification approach via dense-mobilenet models[J]*, Mobile Information Systems,2020:1-8.

[7] Bochkovskiy A, Wang C Y, Liao H Y M., *YOLOv4: optimal speed and accuracy of object detection [EB/OL]*, (2020-4-23) [2023-5- 29].
[8] REDMON J, FARHADI A., *YOLOV3: An Incremental Improvement[EB/OL]*, [2022-03-23]. https://arxiv.org/pdf/1804.02767.pdf.
[9] He K, Gkioxari G, Dollár P, et, *al. Mask r-cnn[C]//Proceedings of the IEEE international conference on computer vision*, 2017: 2961-2969.
[10] Liu W, Anguelov D, Erhan D, et, *al. Ssd: Single shot multibox detector[C]*,//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 21-37.
[11] GONG M,WANG D,ZHAO X,et, *al.A review of nonmaximum suppression algorithms for deep learning target detection[C]*, //Seventh Symposium on Novel Photoelectronic Detection Technology and Application,2021.
[12] WANG C Y,LIAO H Y,WU Y H,et, *al.CSPNet:A new backbone that can enhance learning capability of CNN[C]*,Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops,New York:IEEE,2020:390-391.
[13] Ge Z, Liu S, Wang F, et, *al. Yolox: Exceeding yolo series in 2021[J]*, arXiv preprint arXiv:2107.08430, 2021.
[14] Ding X, Zhang X, Ma N, et al., *Repvgg: Making vgg-style convnets great again[C]*, //Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 13733-13742.
[15] Liu Z, Lin Y, Cao Y, et, *al. Swin transformer: Hierarchical vision transformer using shifted windows[C]*, //Proceedings of the IEEE/CVF international conference on computer vision. 2021: 10012-10022.

Edited by: Bradha Madhavan
Special issue on: High-performance Computing Algorithms for Material Sciences
Received: Jul 18, 2024
Accepted: Aug 21, 2024