IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 24 September 2023, accepted 2 October 2023, date of publication 10 October 2023, date of current version 18 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3323588

RESEARCH ARTICLE

YOLOv7-WFD: A Novel Convolutional Neural Network Model for Helmet Detection in High-Risk Workplaces

JIANJUN CHEN, JUNNING ZHU, ZHUANG LI, AND XIBEI YANG
School of Computer Science, Jiangsu University of Science and Technology, Zhenjiang 212003, China

Corresponding author: Junning Zhu (zhujunning000824@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 62076111, in part by the Natural Science Foundation of Jiangsu Higher Education under Grant 17KJB520007, in part by the Key Research and Development Program of Zhenjiang-Social Development under Grant SH2018005, and in part by the Industry-School Cooperative Education Program of the Ministry of Education under Grant 202101363034.

ABSTRACT In the construction industry, it is common occurrence for head injuries caused by workers not wearing a helmet. However, the current models for detecting safety helmet either have insufficient detection accuracy or insufficient generalization ability. For this reason, an improved convolutional neural network model, called YOLOv7-WFD, is proposed for the detection of workers without helmets in this paper. Firstly, a new module called DBS in this paper is proposed to strengthen the ability of model to extract target features. This module consists of a Deformable Convolutional, a Batch Normalization layer and a SiLU activation function. Secondly, the Content-Aware ReAssembly of Features (CARAFE) module is introduced to perceive effective features, which improves the model's ability to reconstruct details and structural information during image up-sampling. Thirdly, Wise-IoU, which is a loss function with dynamic focusing mechanism, is adopted as the loss function to calculate localization loss, which enhances the generalization capability of model and accuracy of detection. Wise-IoU also can evaluate the ''outlier'' of the anchor box quality, and attenuate the negative impact of low-quality samples in the dataset and enhance the generalization ability of the model. Finally, the experiment shows that the improved YOLOv7-WFD achieves a mAP of 92.6% and a FPS of 79.3 when tested on SHEL5K dataset.

INDEX TERMS Safety helmet detection, YOLOv7, deformable convolution, CARAFE, Wise-IoU.

I. INTRODUCTION

In recent years, object detection has become a popular research topic in computer vision field, especially in application scenarios such as images [1], [2], [3] and videos [4]. Due to its ability to simultaneously classify and localize multiple object categories [5], this technology is significant in addressing engineering safety issues [6], [7].

Within the engineering field, the construction industry is thought of a high-risk industry because of a high fatality rate among workers, according to statistics, the average death rate from head injuries is more than 20%. In order to mitigate death rate, workers are universally mandated to wear safety protective equipment, such as safety helmets. However, due to inadequate on-site supervision and low safety awareness among workers, many workers don't wear helmets on the construction site. Hence, an effective monitoring method is urgently needed to monitor whether workers are wearing safety helmet [8].

In the past, it mainly relied on manual management to supervise the wearing of safety helmets on construction sites. However, due to the large flow of people and the wide scope of construction sites, the efficiency of supervision has been low [9]. With the advancement of technology, video surveillance has gradually become the main means of helmet detection [10]. However, traditional video surveillance

The associate editor coordinating the review of this manuscript and approving it for publication was Deepak Mishra.

relies on human judgment for final decision-making, not fully automated. Hence, an effective automatic monitoring system is urgently needed to monitor whether workers are wearing safety helmet [8]. At present, most of algorithms have been proposed to realize the automation of object detection [14].

Traditional object detection algorithms adopt region selection strategies based on sliding windows [11]. However, the scale and aspect ratio of the sliding window is difficult to set, and the sliding window takes much time to traverse the entire image. In addition, these traditional algorithms are not robust enough for extracting features of multiple targets [15].

Currently, with the development of computer hardware, especially GPU and CPU, the advantages of models based on deep learning method are gradually becoming prominent [12]. Because, firstly, models based on traditional machine learning method require manual design of feature extraction methods, while models based on deep learning method can automatically extract effective features, reducing human effort. Secondly, models based on deep learning method can learn the correlation between features, which enhances models' generalization capability. Thirdly, models based on deep learning method have faster training speed and higher detection accuracy [13]. The flowchart of the models based on deep learning method is shown in Fig. 1.

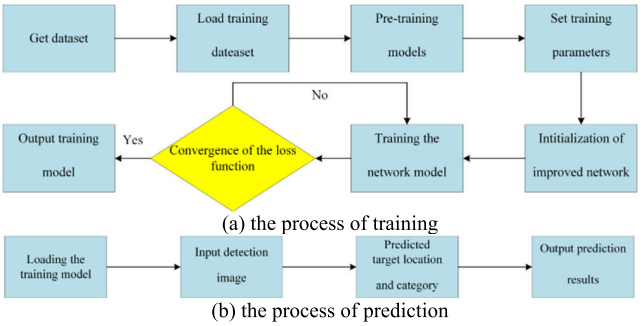

FIGURE 1. Flowchart of the models based on deep learning method.

Among numerous models based on deep learning method, the YOLO has significant advantages. Firstly, YOLO has fast detection speed, currently reaching up to 270 FPS. Secondly, YOLO can simultaneously detect the entire image and avoid false positives caused by background errors. Thirdly, YOLO can learn highly generalized features, making it suitable for transfer learning. Therefore, YOLO is widely used in object detection, especially for helmet detection. For example, M. Dasgupta et al. propose a two-stage motorcycle helmet detection model which combines YOLOv3 with CNN. YOLOv3 is used to identify multiple riders on motorcycle and CNN is proposed for motorcycle rider's helmet detection. This model is superior to other CNN-based models in terms of performance [29]. However, this two-stage architecture introduces additional complexity into the model pipeline, potentially increasing the computational overhead and deployment challenges. L. Huang et al. design a modified YOLOv3 model for helmet detection. First, the model extracts feature from the predicted anchor boxes. Then, extracted features are multiplied the corresponding weight coefficients, and outputs the confidence of each region. Finally, an empirical threshold is used to determine whether the workers comply with the helmet-wearing standard [16]. But introducing a step where features are extracted from predicted anchor boxes and multiplied by weight coefficients can increase the overall complexity of the model. This might impact both training and inference times. L. Shin et al. propose a two-stage algorithm for motorcycle helmet detection. It mainly contributes to adopt two algorithms to classify whether helmets are worn, one is based on handcrafted features, and another is based on convolutional neural networks (CNN) [17]. Handcrafted features can provide more interpretability, allowing designers to understand which specific features contribute to the classification decision. F. Wu et al. design a YOLO-Densebackbone model, which employs DenseNet with fewer parameters and stronger performance to replace the original backbone for feature extraction. Finally, the detection accuracy has been significantly improved [18]. DenseNet, known for its strong feature extraction capabilities, suggests that the model might benefit from improved accuracy due to better feature extraction. S. Tan et al. propose an improved YOLOv5 model by introducing DloU-NMS instead of NMS, making the model more accurate in suppressing predicted bounding boxes [19]. DloU-NMS is designed to improve upon traditional NMS by considering local uniformity and optimizing the non-maximum suppression process. This method could lead to more accurate bounding box suppression and improved object detection. W. Jia et al. design an improved YOLOv5 model for discovering whether motorcycle riders are wearing helmets. The improvements involve incorporating triplet attention and employing soft-NMS [30]. Triplet attention can enhance feature learning by considering relationships between anchor boxes, leading to more discriminative features for helmet detection.

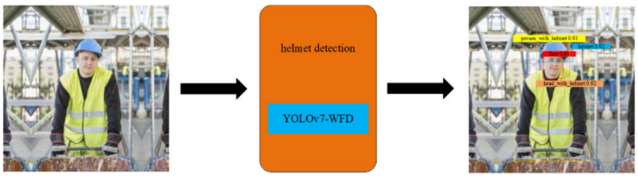

FIGURE 2. The structure of the safety helmet detector. The input image is passed through the safety helmet detector, which outputs the position information and confidence scores of the detected objects.

For safety helmets detection, there exist the following problem to be solved. Due to safety helmets have various shapes and sizes, and they are often partially occluded, making only a portion of them visible, false detection and missed detection are inevitable. To solve this problem, a safety helmet detection model YOLOv7-WFD is proposed, which leverages the benefits of deformable convolutional layers, CARAFE up-sampling operator and Wise-IoU loss function to enhance the learning ability and generalization ability of the model, and the structure of the safety helmet detector is shown in Fig. 2.

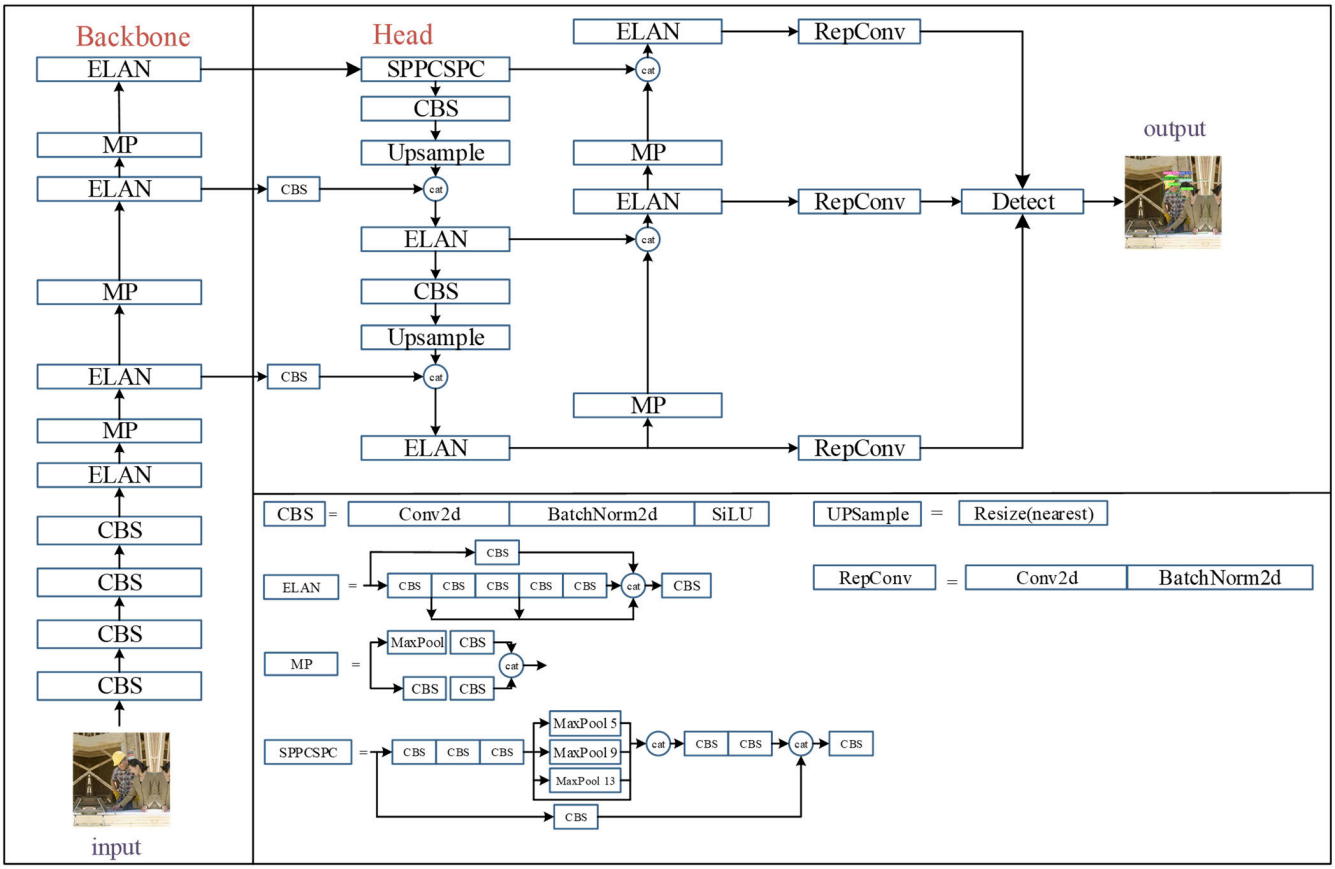

FIGURE 3. The structure of YOLOv7.

The contributions of this paper are summarized as follows:

1) For feature extraction, a new module called DBS is proposed, which consists of a Deformable Convolutional layer (DCN), a Batch Normalization layer and a SiLU activation function.
2) For feature fusion, the CARAFE upsampling operator is introduced to provide a larger field of view for the model.
3) In order to mitigate the adverse influence of low-quality samples and improve the robustness of model, Wise-IoU is introduced to calculate the localization loss.
4) YOLOv7-WFD is compared with different variants of YOLOv7 for helmet detection on the open-source dataset SHEL5K, demonstrating the advantages of YOLOv7-WFD.

The subsequent section of this paper is organized as follows. The structure and details of the improved YOLOv7-WFD are given in section II. Section III gives the experimental results and analysis. The conclusion and prospect are given in Section IV.

II. METHOLOGY

A. YOLOv7 AND YOLOv7-WFD

YOLOv7 is an improved model based on YOLOv5 proposed by the Alexey Bochkovskiy team. It is a one-stage network model known for its excellent detection accuracy and speed. YOLOv7 demonstrates outstanding performance in common object detection tasks such as the COCO dataset, making it an ideal choice as the object detection model [20]. Fig. 3 shows the frame diagram of the YOLOv7.

YOLOv7 consists of two parts: the Backbone and the Head. The workflow of YOLOv7 is descripted in the following. Firstly, the image undergoes preprocessing steps such as enhancement and resizing. Subsequently, the image is fed into the Backbone for feature extraction, resulting in downscaled feature maps. Then, each of these feature maps is performed upsampling operation in the Head, generating three feature maps of different sizes, and fusing them to form a new feature map. Finally, the fused feature map is passed to the Detection module, which processes the feature information and outputs the final detection results.

The Backbone and Head of YOLOv7 mainly consist of CBS modules, ELAN modules, MP modules, RepConv modules and SPPCSPC modules. The modules in the Backbone are used to extract features, and the modules in the Head are used to fuse the extracted features. The CBS module includes a Convolutional layer, a Batch Normalization layer and a SiLU activation function. The ELAN module, which controls the gradient path to promote efficient learning and convergence, and enhances the learning ability of the

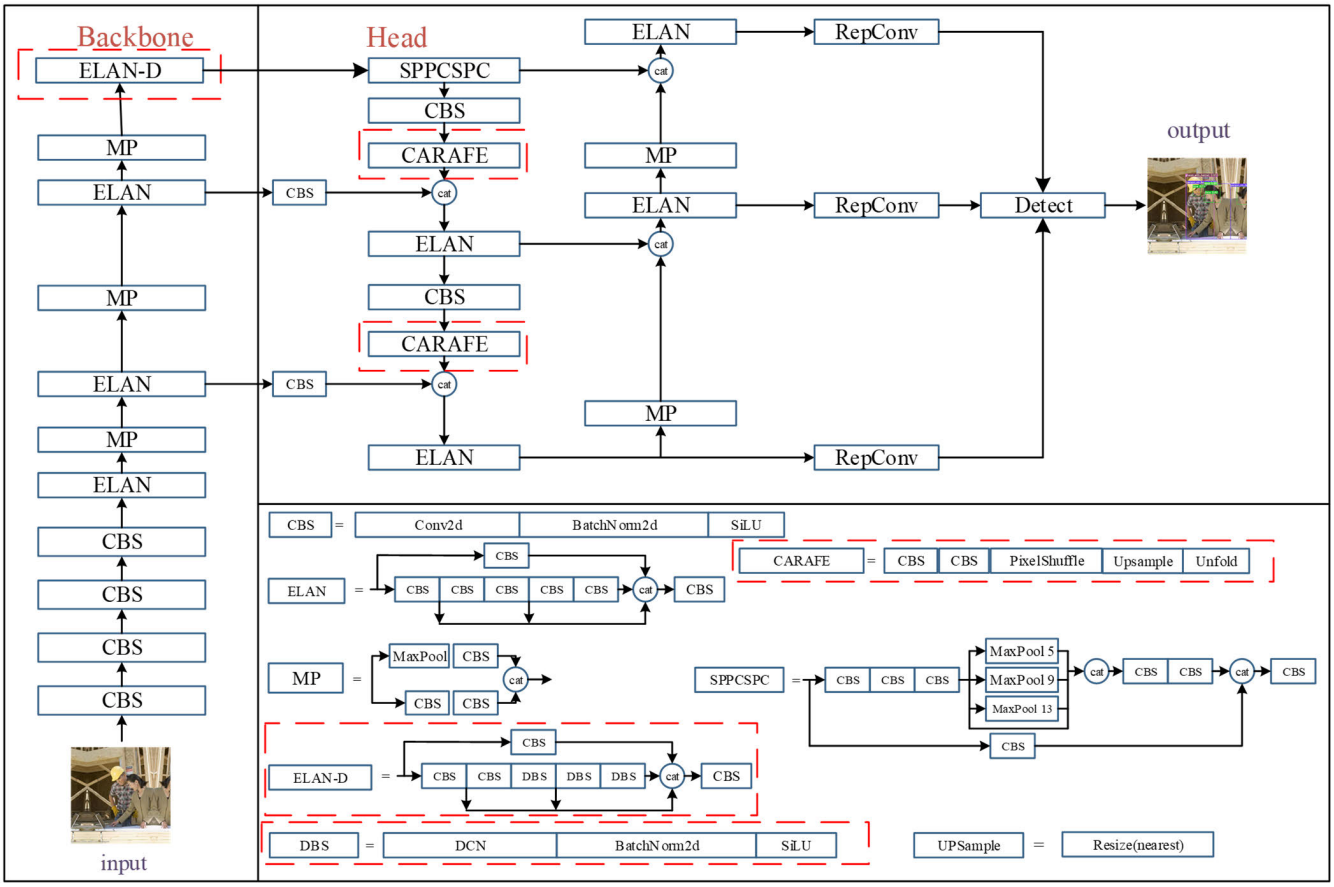

FIGURE 4. The structure of YOLOv7-WFD.

network by combining expansion, shuffling, and merging radix methods without destroying the original gradient path, is composed of multiple CBS modules. The MP module combines max-pooling layers and CBS modules. RepConv module includes a Convolutional layer, a Batch Normalization layer and a CBS module. SPPCSPC module includes multiple CBS modules and three max-pooling layers with different scales [21].

YOLOv7-WFD is an improved model based on YOLOv7. In the final ELAN module of the Backbone, some CBS modules are replaced with DBS modules, which introduces deformable convolutional layers. In the Head, the CARAFE upsampling operator is applied, and the Wise-IoU loss function is used to compute the localization loss. The frame diagram of the YOLOv7-WFD is shown in Fig. 4.

B. DBS FOR FEATURE EXTRACTION

DBS is an improved module based on CBS, and consists of a Deformable Convolutional (DCN), a Batch Normalization layer and a SiLU activation function. This module has advantages as follows: 1) The combination of deformable convolution, batch normalization layer, and SiLU can lead to faster convergence during training and improved generalization performance. The gentle non-linearity of SiLU can avoid some of the vanishing gradient issues. 2) By incorporating the DBS module into the architecture of YOLOv7, the model can potentially achieve better object detection results, especially when dealing with complex scenes or instances that exhibit significant variations in appearance. 3) The adaptive nature of DCN and the regularization effect of batch normalization layer can contribute to reduced overfitting, allowing the model to generalize better to unseen data. 4) The DBS module's components can be integrated into various neural network architectures, providing flexibility to experiment and improve model performance.

Due to the presence of rotation and deformation in safety helmets and faces during the detection process, traditional convolutional layers with fixed receptive fields may not effectively capture these details of spatial transformations. Thus, the deformable convolutional layers are introduced to enhancing the model's capability of feature extraction [33].

DCN operator can achieve higher detection accuracy at the expense of a slight decrease in detection speed. Compared to traditional convolutions, DCN has advantages as follows: 1) DCN introduces an offset at the sampling positions, which makes the structure of the convolutional kernel non-fixed, instead, it is dynamically adjusted based on the features of the objects in the images. This flexible mapping between kernel and features allows for a broader coverage of appearance features in the detected targets, thereby more valuable information can be captured. 2) DCN utlizes the depthwise separable convolution technique to detach

the convolution weights into depth-wise part and point-wise part. The depth part is responsible for the original location-aware modulation scalar. The point direction part is responsible for the shared projection weight between sampling points. Compared to conventional convolutions, DCN has fewer parameters and lower computational cost, resulting in faster computation speed for the model. 3) DCN introduces multi-group mechanism, each group performing different offset sampling, sample vector projection, and factor modulation. This method enhances the expressive ability of the DCN operator. 4) DCN utilizes the softmax function to normalize the modulation scalar, which enhances the stability of the model. 5) DCN adopts sparse sampling method, which can enhance the feature extraction capability of model when detecting humans and objects. Thus, DCN has greater adaptability, and addresses the problem of traditional convolutions' inability to acquire long-range dependencies, making the model more suitable for diverse object detection [20].

Given an input $x \in \mathbb{R}^{C\times H\times W}$ and current pixel p_0. DCN can be formulated as follows:

$$y(p_0) = \sum_{g=1}^{G}\sum_{k=1}^{K} w_g m_{gk} x_g(p_0 + p_k U 1 + \Delta p_{gk}) \qquad (1)$$

where, G represents the total number of groups. K represents the total number of sampling points, and k enumerates the sampling point. w_g denotes the projection weights of the g-th group's sampling point. m_{gk} represents the location-irrelevant projection weights of the g-th group and the k-th grid sampling location, which is normalized by sigmoid function. $x_g \in \mathbb{R}^{C'\times H\times W}$ represents the sliced input feature map. p_k represents that the k-th location of the pre-defined grid sampling is regular convolutions, and p_{gk} represents the offset corresponding to the g-th group and the k-th grid sampling location. $w_g \in \mathbb{R}^{C\times C'}$ and $C' = C/G$.

The structures of traditional convolution kernel and deformable convolution kernels in 3×3 standard as shown in Fig. 5, (a) represents traditional convolution, whereas (b), (c), and (d) represent different states of dynamic sparse kernels in deformable convolution under different offset values.

Batch normalization is a technique that normalizes the output of a layer within a mini-batch of data. It helps in mitigating the internal covariate shift, which can lead to more stable and faster training. In the DBS module, batch normalization is applied after the deformable convolution to ensure that the input distribution remains stable during the training process.

SiLU (Sigmoid Linear Unit), is an activation function that smoothly combines the properties of the sigmoid and linear functions. It is defined as $\mathrm{SiLU}(x) = x * \mathrm{sigmoid}(x)$, and its smoothness allows for better gradient flow during training compared to traditional activation functions like ReLU. SiLU has been observed to improve convergence and generalization in deep neural networks.

The processing of DBS is described as follows: Firstly, the input image is convolved by DCN. DCN applies convolutional operations with learnable offsets, allowing the network to adjust the sampling grid dynamically to capture deformable patterns within the image. The learnable offsets help in adaptively aligning the convolutional sampling points with relevant features, enabling the network to capture intricate and deformable patterns effectively. Secondly, the features output from DCN are normalized by Batch Normalization. Batch Normalization normalizes the activations through subtracting the batch mean and dividing by the batch standard deviation, improving the stability and efficiency of the network during training. Learnable scale and shift parameters in Batch Normalization allow the network to adapt and fine-tune the normalized activations. Finally, the normalized features are input to the SiLU. Sigmoid function is applied to scale the input by: $\mathrm{SiLU}(x) = x * \mathrm{sigmoid}(x)$, which introduces non-linearity to the features. SiLU has a smooth gradient, promoting smoother gradient flow during backpropagation, which help in efficient training of the network. After the processing of DCN, Batch Normalization, and SiLU activation, the output features represent the transformed and enhanced features of the input image within the DBS module. These features can then be further used for subsequent layers in the neural network.

In summary, the DBS module, comprising DCN, Batch Normalization layer, and SiLU activation, aims to enhance the YOLO model's capability to handle object detection tasks by adapting to object variations and improving training dynamics.

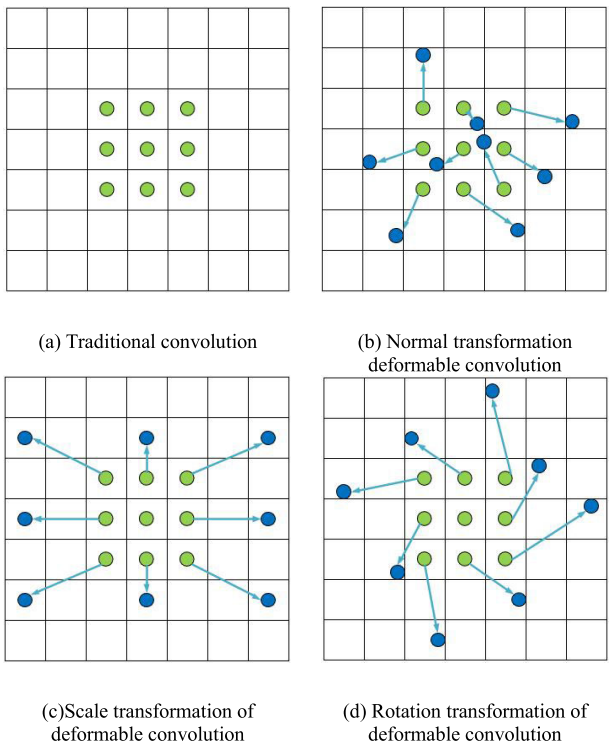

FIGURE 5. Traditional convolution and different representations of deformable convolution.

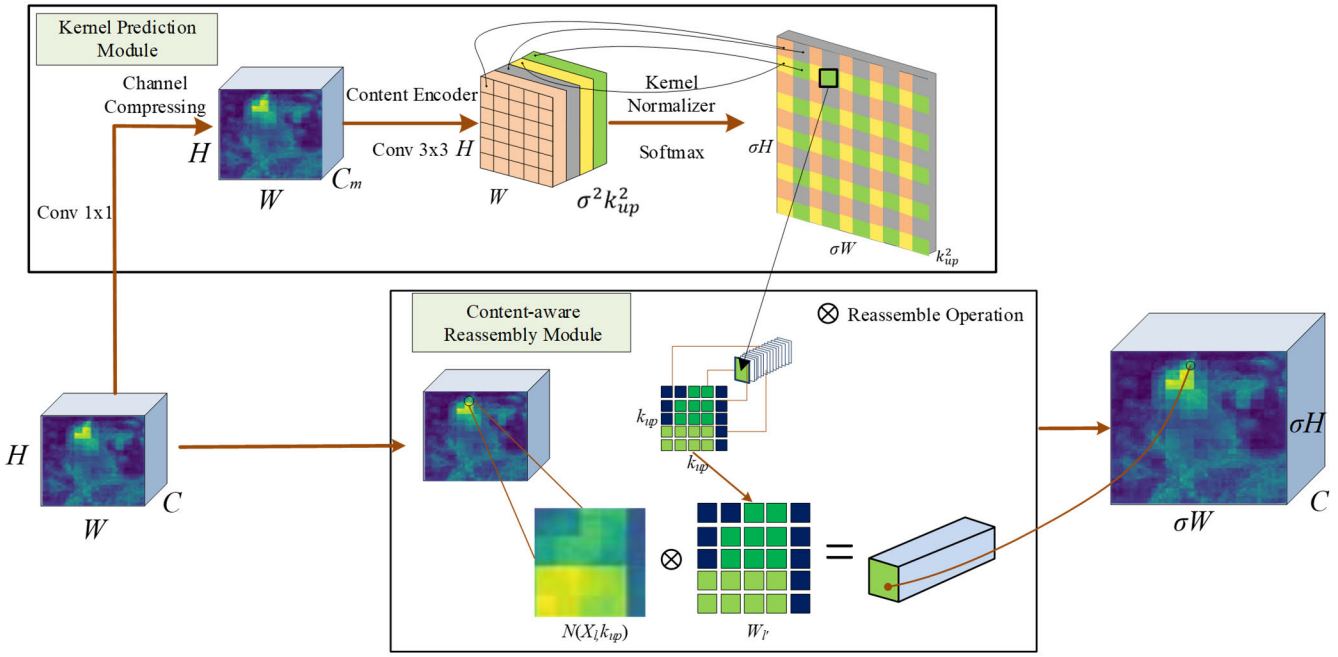

FIGURE 6. The overall framework of CARAFE ($\sigma = 2$, k_{up} is the reassembly kernel size, $N(X_l, k)$ is the $k \times k$ sub-region of X centered at the location l, i.e., the neighbor of X_l).

C. CARAFE FOR FEATURE FUSION

Feature upsampling is a key operator in object detection tasks. In YOLOv7-WFD, CARAFE operator is used for feature upsampling. The integration of the CARAFE module into the YOLOv7 model amplifies its proficiency in object detection tasks through a refined upscaling process. First of all, the CARAFE module takes a downsampled feature map as input, which encapsulates high-level semantic information extracted from the image. Secondly, CARAFE employs the subpixel convolution operation for expanding the spatial dimensions of the input feature map to the desired resolution. However, rather than executing a conventional convolution, CARAFE undertakes a content-aware reassembly approach. During this phase, the module judiciously reassembles feature values within each pixel's receptive field by employing learned weights for a weighted combination of neighboring feature values. This adaptive amalgamation harnesses contextual cues from proximate pixels, yielding an output that aligns with the image's content and structure. At last, the reassembled feature values undergo aggregation to their corresponding positions in the higher-resolution output, assuring the retention of fine-grained details and semantic fidelity during the upscaling process. The CARAFE upsampling operator has several characteristics as follows: 1) CARAFE has large receptive field. Conventional methods typically use nearest-neighbor interpolation and bilinear interpolation, which limit the receptive field of the model to 1×1 or 2×2. However, CARAFE considers the entire feature map during the upsampling process, which expands the receptive field of the model, thus better preserving image details and edge information while reducing the occurrence of jagged edges and blurring effects during upsampling. 2) CARAFE supports instance-specific content-aware processing by dynamically generating adaptive kernels that are suitable for different target. But nearest-neighbor interpolation and bilinear interpolation rely on fixed rules for upsampling and cannot adjust the size of kernel according to the content. 3) CARAFE introduces little computational overhead and has good adaptability to existing network models [22]. CARAFE can be seamlessly integrated into any position of deep neural networks. Compared to nearest-neighbor interpolation or bilinear interpolation, CARAFE is easier to be combined with other network layers such as convolutional layers or pooling layers. The overall framework of the CARAFE module is shown in Fig. 6.

During the calculation of CARAFE, if a feature map X with dimension $C \times H \times W$ and upsampling factor σ is provided, CARAFE will generate a new feature map X' of size $C \times \sigma H \times \sigma W$ [26]. For any position $l = (i, j)$ in the input X, there is a corresponding position $l' = (i', j')$ in the output X', Here, $i = [i'/\sigma]$ and $j = [j'/\sigma]$. Specifically, the CARAFE upsampling operator can be further subdivided into two modules: kernel prediction module and content-aware reassembly module [22].

1) KERNEL PREDICTION MODULE

CARAFE generates adaptive reassembly kernels through prediction based on the content of the target location. The role of reassembly kernels is to recombine and adjust the features

to obtain more accurate and richer upsampling results. This prediction process ensures the effective capture of features by the CARAFE operator, and the size of reassembly kernels is $k_{up} \times k_{up}$.

Equation (2) is the recombination kernel generation expression. The kernel prediction module ψ generates a recombination kernel based on the perception of the content and the neighborhood, and predicts a location-based kernel $w_{l'}$ for each location l'.

$$w_{l'} = \psi(N(x_l, k_{encoder})) \tag{2}$$

where, x_l is the coordinate of a certain pixel on X, $k_{encoder} = k_{up} - 2$ is size of convolution kernel.

The kernel prediction module can be subdivided into three sub-modules: channel compressor, content encoder, and kernel normalizer. These sub-modules are explained in detail as follows:

a: CHANNEL COMPRESSOR

By employing a 1×1 convolutional layer to compress the input feature channels from C to C_m, the number of parameters and computational cost of the model are reduced, resulting in improved computation speed. Additionally, this allows for a larger kernel size in the subsequent *contentencoder*.

b: CONTENT ENCODER

A convolution layer of kernel size $k_{encoder}$ is applied to generate reassembly kernels base on the content of input features. The parameter of the encoder is $k_{encoder} \times k_{encoder} \times C_{up}$, $C_{up} = \sigma^2 k_{up}^2$.

c: KERNEL NORMALIZER

The softmax function is applied to each reassembly kernel for normalization, ensuring the weights and adaptiveness of the kernels.

2) CONTENT-AWARE REASSEMBLY MODULE

In the content-aware reassembly module, a weighted sum operator ϕ is applied to perform feature reassembly on $N(x_l, k_{up})$ centered at $l = (i, j)$, the reassembly is shown in equation (3), where $r = [k_{up}/2]$.

$$\begin{aligned} \mathcal{X}'_{l'} &= \phi\left(N\left(\mathcal{X}_l, k_{up}\right), W_{l'}\right) \\ &= \sum_{n=-r}^{r} \sum_{m=-r}^{r} W_{l'(n,m)} \cdot \mathcal{X}_{(i+n,j+m)} \end{aligned} \tag{3}$$

CARAFE adopts a fixed set of hyper-parameters in experiments, where C_m is 64 for the channel compressor and $k_{encoder} = 3$, $k_{up} = 5$ for the content encoder.

D. WISE-IoU

When training the network model, different loss functions will have varying impacts on the training results of the model [25]. The loss function of YOLOv7, as shown in Equation (4), consists of three components: the confidence loss L_{conf}, the classification loss L_{cls}, and the localization loss L_{loc} [24]. Both L_{conf} and L_{cls} are calculated by BCE with logits loss function, and L_{loc} is calculated by CIoU. Equation (5) is the BCE with logits loss function calculation formula, and Equation (6) is the CIoU calculation formula.

$$Loss = L_{conf} + L_{cls} + L_{loc} \tag{4}$$

$$Loss_{BCE} = -\frac{1}{N} \sum_{i=1}^{N} [y_i log(p_i) + (1 - \mathbf{y}_i) log(1 - p_i)] \tag{5}$$

$$Loss_{CIoU} = 1 - I_{iou} + \frac{p^2_{(b,b^{gt})}}{d^2} + \alpha v \tag{6}$$

In Equation (5), y_i represents the the binary label of the sample, p_i represents the probability of the sample, and d is the diagonal distance of the smallest enclosed region that contains both the predicted bounding box and the ground truth bounding box. In Equation (6), b represents the centroid of the predicted bounding box, b_{gt} represents the centroid of the ground truth bounding box, p represents the euclidean metric between b and b_{gt}. The following formulas are defined to calculating α, v, I_{iou}, and the meaning of parameters are represented in Fig. 7.

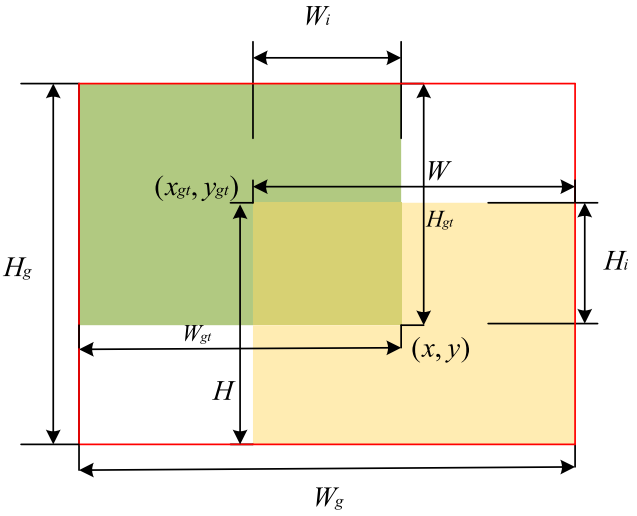

FIGURE 7. Schematic diagram of Calculation parameters.

$$\alpha = \frac{v}{(1 - IoU) + v} \tag{7}$$

$$v = \frac{4}{\pi^2}(arctan\frac{W_{gt}}{H_{gt}} - arctan\frac{W}{H})^2 \tag{8}$$

$$I_{iou} = 1 - \left| \frac{A \cap B}{A \cup B} \right| \tag{9}$$

where A represents the area of the ground truth bounding box, B represents the area of the predicted bounding box. In Fig. 7, the green region represents the ground truth bounding box, and the yellow region represents the predicted bounding box.

H_{gt} and W_{gt} are the height and width of the ground truth bounding box. H and W are the height and width of the predicted bounding box. H_g and W_g are the height and width of the smallest rectangle that simultaneously encloses both the predicted and ground truth bounding boxes. (x_{gt}, y_{gt}) and (x, y) are the centroids of the ground truth bounding box and the predicted bounding box, respectively. H_i and W_i are the height and width of the overlop between ground truth bounding box and predicted bounding box.

(a)model without Wise-IoU (b)model with Wise-IoU

FIGURE 8. The performance of model with or without Wise-IoU.

Although CIoU possesses well stability, it is not suitable for safety helmet detection task that requires well generalization capability. For safety helmet detection tasks, the data may exhibit class imbalance or sample imbalance. In such cases, the CIoU loss function may not effectively address these problems. In addition, the overlap between predicted boxes and ground truth boxes varies for different training samples, and low-quality samples will generate harmful gradients. In order to enhance the generalization capability of model and mitigate the negative impact of low-quality samples on the training results, the model adopts Wise-IoU. As shown in Fig. 8, there are two workers in different lighting and the face of one worker is blocked. In addition to different detection accuracies, the model without Wise-IoU failed to detect the occluded face. Wise-IoU combines a dynamic non-monotonic focusing mechanism that utilizes ''outlier'' to evaluate the quality of anchor boxes. A bigger outlier degree indicates lower quality for an anchor box, and this anchor box will be assigned a smaller gradient gain to focus the bounding box regression on anchor boxes of higher quality. The formula of Wise-IoU is as follows:

$$Loss_{Wise-IoU} = r \cdot exp(\frac{\rho^2_{(b,b^{gt})}}{(d^2)^*}) \cdot (1 - IoU) \quad (10)$$

$$r = \frac{\beta}{\delta \alpha^{\beta-\delta}}, \beta = \frac{(1 - IoU)^*}{1 - \bar{I}oU} \in [0, +\infty) \quad (11)$$

where $*$ indicates that W_g and H_g are separated from the computed graph to avoid creating gradients that affect convergence, r represents the gradient gain. β represents the degree of the outliers, α and δ are hyperparameters, with α set to 1.9 and δ set to 3 [32]. Since IoU is dynamic, the quality demarcation standard of anchor boxes is also dynamic, which allows Wise-IoU to make the gradient gain allocation strategy that is most in line with the current situation at every moment.

III. EXPERIMENTAL RESULTS AND ANALYSIS

A. DATASET AND EXPERIMENTAL ENVIRONMENT

The dataset used in this experiment is SHEL5K, a publicly available dataset from the Kaggle website [27], [28]. This dataset is an enhanced version of the SHD dataset [27] and consists of 5000 images with a resolution of 416×416 pixels and 75570 complex background labels.

This dataset is specifically designed for safety helmet detection and includes images from various environments, angles, lighting conditions, and scenarios. For example, it encompasses both indoor and outdoor scenes, scenarios with dense and sparse crowds, partially or completely occluded safety helmets, and helmets under diverse lighting conditions. However, the dataset exhibits imbalance in terms of sample quantities of different classes, which could potentially influence training and evaluation of model. The dataset includes six categories: helmet, head, head_with_helmet, person_with_helmet, people_no_helmet, and face. Fig. 9 illustrates the distribution of each class in the dataset in terms of percentages. Fig. 10 shows some samples in the dataset [26].

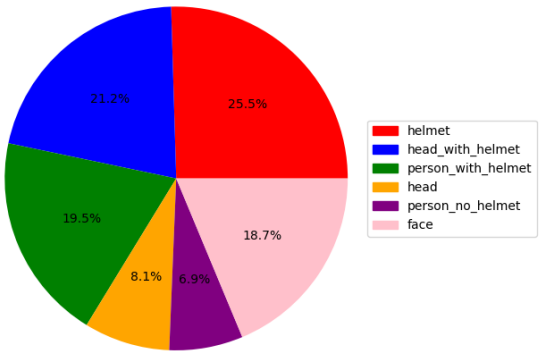

FIGURE 9. Scale of different categories in SHEL5K.

In this experiment, the dataset is divided into three parts: training (4000 samples), validation (500 samples), and testing (500samples). All the experiments are conducted in Windows 10 system environment, in which the Pycharm software used in this experiment is equipped with the following environment: pytorch1.7, python3.7, CUDA11.1; hardware environment and related model parameters are shown in Table 1 and Table 2, respectively.

TABLE 1. Experimental environment configuration.

Items	Description
CPU	i9-12900K
MEMORY	32 GB
STORAGE	Samsung 980 PRO 1TB
GPU	RTX 3090 Ti

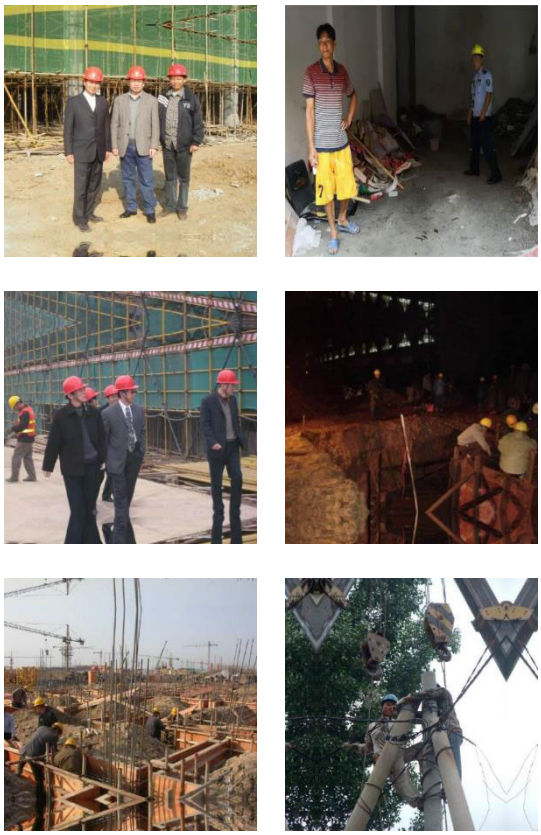

FIGURE 10. Image samples in SHEL5K.

TABLE 2. Experimental model parameters.

Training Parameters	Values
Learning Rate	0.01
Batch Size	16
Epochs	100
Image Size	640*640
Momentum	0.937
Weight Decay	0.0005

B. EVALUATION METRICS

Precision (P), Recall (R), Average Precision (AP) and mean Average Precision (mAP) are used to measure the detection tasks.

Precision and Recall provide measures of accuracy and coverage in object detection. Precision refers to the proportion of samples predicted as positive by the model that are actually positive. Recall refers to the proportion of true positive samples correctly detected by the model [31]. The Precision and Recall are calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

where T/F represents true/false of prediction results. P/N stands for positive/negative of prediction results.

AP is a metric commonly used in object detection tasks, which combines precision and recall to evaluate the model's performance at different thresholds. mAP, on the other hand, represents the average AP across all classes and is used to comprehensively assess the model's performance on multiple categories. Steps for calculating AP and mAP are as follows:

Step1: For each class, calculate the Precision-Recall curve based on the model's predictions and the ground truth labels.
Step2: Compute the area under the curve, which represents the Average Precision (AP).
Step3: Average the AP values across all classes to obtain the mAP.
AP and mAP are calculated as follows:

$$AP = \int_0^1 P(R)dR \tag{14}$$

$$mAP = \frac{1}{n}\sum AP \tag{15}$$

where n represents the number of classes.

C. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, the effectiveness of the proposed YOLOv7-WFD is demonstrated through experiments. The Precision-Recall curve of YOLOv7-WFD is shown in Fig. 11. The fluctuation of the P-R curve indicates the training performance of the model. Fig. 11 demonstrates that the model exhibits excellent training performance.

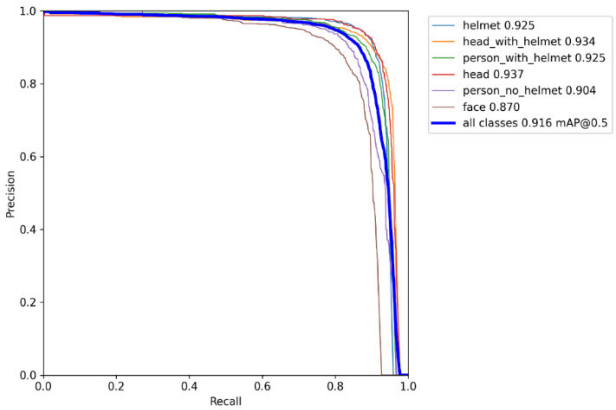

FIGURE 11. Precision-Recall curve of YOLOv7-WFD.

Since the performance of YOLOv7 has been demonstrated to be superior to other models in the YOLO family as well as classical models such as Faster R-CNN in the paper [31], this paper do not repeat the comparative experiments in this study. YOLOv7-WFD was compared with YOLOv7, YOLOv7-X, YOLOv7-W6, YOLOv7-E6, YOLOv7-D6, YOLOv7-E6E on the dataset SHEL5K. The mAP of each model is respectively shown in Table 3, in which C1-C6 represent different categories, where C1 represents "helmet", C2 represents "head_with_helmet", C3 represents "person_with_helmet",

TABLE 3. mAP and FPS of each model on each category.

Model	mAP(%)	C1	C2	C3	C4	C5	C6	FPS
YOLOv7	91.2	92.5	93.2	92.9	92.3	90.4	85.7	132
YOLOv7-X	91.4	92.5	93.1	92.9	92.2	90.9	86.6	100
YOLOv7-W6	89.0	90.2	92.0	92.1	91.3	90.1	78.2	115
YOLOv7-E6	88.9	89.7	91.6	91.8	90.8	89.8	79.4	85
YOLOv7-D6	87.0	88.1	90.7	89.8	90.0	87.1	76.2	70
YOLOv7-E6E	89.4	89.8	92.1	92.3	91.5	90.5	80.0	60
YOLOv7-WFD	**92.6**	**93.7**	**94.7**	**93.6**	**93.3**	**92.3**	**87.9**	**79**

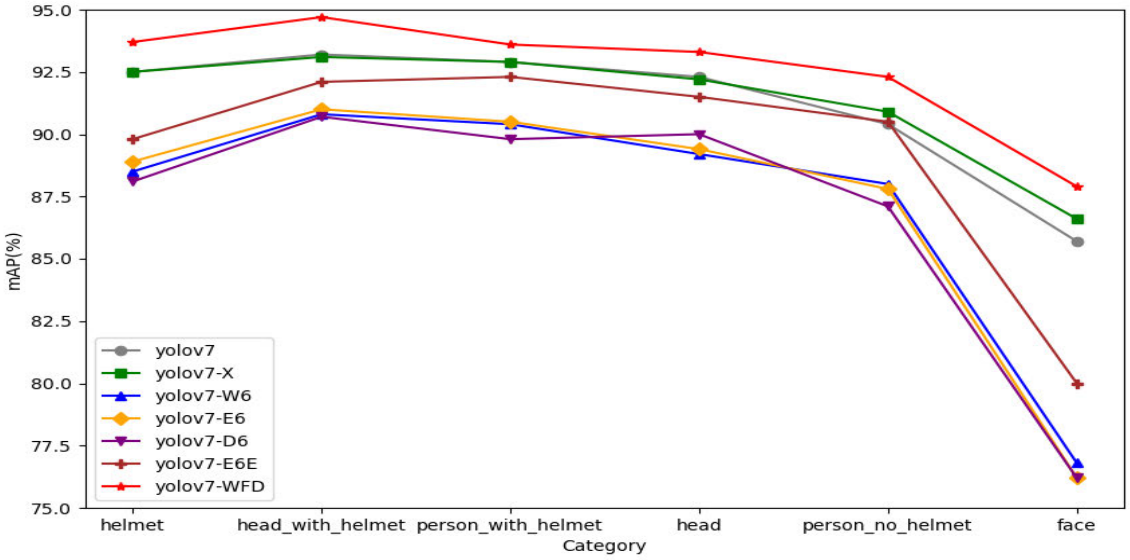

FIGURE 12. mAP of each model on each category.

TABLE 4. Pression, recall and F1 score of each model.

Model	Pression(%)	Recall(%)	F1(%)
YOLOv7	89.5	85.8	87.6
YOLOv7-X	91.4	84.4	87.8
YOLOv7-W6	89.5	82.8	86.0
YOLOv7-E6	90.3	82.0	86.0
YOLOv7-D6	89.8	79.9	84.6
YOLOv7-E6E	91.3	82.0	86.4
YOLOv7-WFD	**91.7**	**87.0**	**89.0**

TABLE 5. mAP of YOLOv7-WFD at different learning rates.

Learning rate	mAP(%)
0.001	83.1
0.005	90.1
0.01	92.6
0.015	91.1
0.02	91.0

C4 represents "head", C5 represents "person_no_helmet", and C6 represents "face". FPS represents the number of images the model can process per second. The visual results of the data from Table 3 are shown in Fig. 12. The Pression, Recall and F1 score of each model is shown in Table 4.

Combining Table 3 and Fig. 12, it can be found that the mAP of the model proposed in this paper has been improved to varying degrees in each category. Especially, the mAP of the C6 (face) has been improved significantly. By introducing CARAFE and deformable convolution, YOLOv7-WFD gains a stronger ability to capture complex details. Furthermore, it can be observed that the FPS of YOLOv7-WFD has decreased. There are several reasons for this: Deformable convolution and CARAFE introduce additional operations and parameters, leading to increased computational overhead, which can result in longer processing times per frame; the hardware acceleration is not fully utilized, which also leads to the reduction of FPS. The introduction of the DBS module is likely to increase the computational cost. Depending on the specific operations within the DBS module, such as convolutions and activations, the module's computational demand may vary. The added computations could affect both training

TABLE 6. The result of ablation study.

DBS	CARAFE	Wise-IoU	mAP@0.5(%)	mAP@0.5:0.05:0.95(%)	GFLOPs	FPS	Params (Millions)
			91.2	56.6	105.2	100.0	37.21
		√	92.3(+1.1)	59.3(+2.7)	105.2	104.1	37.22
	√	√	92.6(+0.3)	59.7(+0.4)	103.3	90.9	37.26
√	√	√	92.6(+0)	60.2(+0.5)	102.1	79.3	35.73

(a) The original image (b) YOLOv7 (c) YOLOv7-X (d) YOLOv7-W6

(e) YOLOv7-E6 (f) YOLOv7-D6 (g) YOLOv7-E6E (h) YOLOv7-WFD

FIGURE 13. Target detection results obtained using different methods.

and inference times. The CARAFE module's purpose is to improve the model's feature perception during image up-sampling. While it enhances detail reconstruction, it can also increase computational requirements due to additional convolutional operations and calculations associated with the CARAFE mechanism.

In addition, the adoption of the Wise-IoU loss function introduces a change in the loss calculation process during training. Depending on its complexity compared to traditional loss functions, it might have a minor impact on training time due to the additional calculations required during each training iteration. Also, the inference process involves forwarding an image through the model to make predictions. The introduced DBS and CARAFE modules could increase inference time due to the added computation in feature extraction and up-sampling stages.

To validate the effectiveness of YOLOv7-WDF, ablation experiments is conducted. The mAP of YOLOv7-WFD at different learning rates are shown in Table 5, and the results of the ablation experiments are shown in Table 6.

In Table 6, 0.5 and 0.5:0.05:0.95 represent IoU thresholds. GFLOPs refers to the number of billions of floating-point operations per second, which measures the complexity of the model. Params represents the number of parameters in the model. It can be observed that the introduction of DCN in the model resulted in a decrease in GFLOPs and Params. There are several reasons for this: the deformable convolution requires additional calculation for learning offsets and weights, leading to a decrease in GFLOPs. Additionally, Deformable convolutions may have replaced some parameters of the original convolutional layers, such as substituting certain position-sensitive convolutions or specific convolutional operations, leading to a decrease in Params. From Table 4, it can be found that the proposed model achieves improved accuracy compared to the original YOLOv7, while still meeting the real-time detection requirements. Therefore, the ablation results prove the effectiveness of the improved model in this study.

Fig. 13 shows an example of target detection results obtained using different methods. From Fig. 13, it can be observed that the model proposed in this paper has better capability of detection for each category and better capability of localization.

IV. CONCLUSION AND PROSPECT

In this paper, an improved model YOLOv7-WFD, based on YOLOv7, is proposed to enhance model's capability of object detection. The improvements include:

1) A new module called DBS in this paper is proposed. In the DBS, traditional convolutional layer is replaced with deformable convolutional layer. This approach strengthens the ability of model to extract target features.
2) The introduction of the CARAFE upsampling operator enables the model to better reconstruct details and structural information during the image upsampling process.
3) By adopting Wise-IoU instead of CIoU, the adverse influence of low-quality samples in the dataset on the model is mitigated, and the generalization capability is improved.

In addition, the experimental results show the rationality and effectiveness of YOLOv7-WFD.

In the following directions, we will make efforts to improve the model to better meet the requirements of the safety helmet detection project:

1) Construct a more suitable dedicated dataset that better reflects real-world scenarios and features.
2) Make appropriate parameter adjustments to improve detection speed while ensuring accuracy.
3) Design a complete system to apply YOLOv7-WFD to actual construction sites environments.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their precious suggestions.

REFERENCES

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.
[2] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[4] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang, and W. Ouyang, "T-CNN: Tubelets with convolutional neural networks for object detection from videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2896–2907, Oct. 2018.
[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2009.
[6] R. Waranusast, N. Bundon, V. Timtong, C. Tangnoi, and P. Pattanathaburt, "Machine vision techniques for motorcycle safety helmet detection," in *Proc. 28th Int. Conf. Image Vis. Comput. New Zealand (IVCNZ)*, Nov. 2013, pp. 35–40.
[7] R. R. V. E. Silva, K. R. T. Aires, and R. D. M. S. Veras, "Helmet detection on motorcyclists using image descriptors and classifiers," in *Proc. 27th SIBGRAPI Conf. Graph., Patterns Images*, Aug. 2014, pp. 141–148.
[8] J. Han, Y. Liu, Z. Li, Y. Liu, and B. Zhan, "Safety helmet detection based on YOLOv5 driven by super-resolution reconstruction," *Sensors*, vol. 23, no. 4, p. 1822, Feb. 2023.
[9] H. Zhao, J. Liu, H. Chen, J. Chen, Y. Li, J. Xu, and W. Deng, "Intelligent diagnosis using continuous wavelet transform and Gauss convolutional deep belief network," *IEEE Trans. Rel.*, vol. 72, no. 2, pp. 1–11, Jun. 2022.
[10] C. Huang, X. Zhou, X. Ran, Y. Liu, W. Deng, and W. Deng, "Co-evolutionary competitive swarm optimizer with three-phase for large-scale complex optimization problem," *Inf. Sci.*, vol. 619, pp. 2–18, Jan. 2023.
[11] T. Surasak, I. Takahiro, C.-H. Cheng, C.-E. Wang, and P.-Y. Sheng, "Histogram of oriented gradients for human detection in video," in *Proc. 5th Int. Conf. Bus. Ind. Res. (ICBIR)*, May 2018, pp. 172–176, doi: 10.1109/ICBIR.2018.8391187.
[12] X. Luo, J. Chen, T. Song, and G. Wang, "Semi-supervised medical image segmentation through dual-task consistency," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, May 2021, pp. 8801–8809.
[13] X. Luo, G. Wang, W. Liao, J. Chen, T. Song, Y. Chen, S. Zhang, D. N. Metaxas, and S. Zhang, "Semi-supervised medical image segmentation via uncertainty rectified pyramid consistency," *Med. Image Anal.*, vol. 80, Aug. 2022, Art. no. 102517.
[14] J. Li, H. Qin, J. Wang, and J. Li, "OpenStreetMap-based autonomous navigation for the four wheel-legged robot via 3D-lidar and CCD camera," *IEEE Trans. Ind. Electron.*, vol. 69, no. 3, pp. 2708–2717, Mar. 2022, doi: 10.1109/TIE.2021.3070508.
[15] J. Lu, J. Chen, T. Xu, J. Song, and X. Yang, "Element detection and segmentation of mathematical function graphs based on improved mask R-CNN," *Math. Biosci. Eng.*, vol. 20, no. 7, pp. 12772–12801, 2023.
[16] L. Huang, Q. Fu, M. He, D. Jiang, and Z. Hao, "Detection algorithm of safety helmet wearing based on deep learning," *Concurrency Comput., Pract. Exper.*, vol. 33, no. 13, p. e6234, Feb. 2021.
[17] L. Shine and C. V. Jiji, "Automated detection of helmet on motorcyclists from traffic surveillance videos: A comparative analysis using hand-crafted features and CNN," *Multimedia Tools Appl.*, vol. 79, nos. 19–20, pp. 14179–14199, Feb. 2020.
[18] F. Wu, G. Jin, M. Gao, H. E. Zhiwei, and Y. Yang, "Helmet detection based on improved YOLOv3 deep model," in *Proc. IEEE 16th Int. Conf. Netw., Sens. Control (ICNSC)*, May 2019, pp. 363–368.
[19] S. Tan, G. Lu, Z. Jiang, and L. Huang, "Improved YOLOv5 network model and application in safety helmet detection," in *Proc. IEEE Int. Conf. Intell. Saf. Robot. (ISR)*, Mar. 2021, pp. 330–333.
[20] Z. Yang, H. Feng, Y. Ruan, and X. Weng, "Tea tree pest detection algorithm based on improved YOLOv7-tiny," *Agriculture*, vol. 13, no. 5, p. 1031, May 2023.
[21] C.-Y. Wang, A. Bochkovskiy, and H.-Y.-M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7464–7475.
[22] J. Wang, K. Chen, R. Xu, Z. Liu, C. C. Loy, and D. Lin, "CARAFE: Content-aware reassembly of features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3007–3016.
[23] J. Ying, H. Li, H. Yang, and L. Zheng, "GPK-YOLOv5s: Content-aware reassembly of features and self attention for high altitude parabolic detection and tracking," in *Proc. MATEC Web Conf.*, vol. 363, Aug. 2022, p. 01012.
[24] B. Li, Y. Chen, H. Xu, and F. Zhong, "Fast vehicle detection algorithm based on lightweight YOLO7-tiny," 2023, *arXiv:2304.06002*.
[25] X. Chen, M. Yuan, Q. Yang, H. Yao, and H. Wang, "Underwater-YCC: Underwater target detection optimization algorithm based on YOLOv7," *J. Mar. Sci. Eng.*, vol. 11, no. 5, p. 995, May 2023.
[26] Y. Li, Q. Fan, H. Huang, Z. Han, and Q. Gu, "A modified YOLOv8 detection network for UAV aerial image recognition," *Drones*, vol. 7, no. 5, p. 304, May 2023.
[27] Larxel. *Safety Helmet Detection*. Accessed: May 2023. [Online]. Available: https://www.kaggle.com/datas-ets/andrewmvd/hard-hat-detection
[28] H. Liang and S. Seo, "Automatic detection of construction Workers' helmet wear based on lightweight deep learning," *Appl. Sci.*, vol. 12, no. 20, p. 10369, Oct. 2022.
[29] M. Dasgupta, O. Bandyopadhyay, and S. Chatterji, "Automated helmet detection for multiple motorcycle riders using CNN," in *Proc. IEEE Conf. Inf. Commun. Technol.*, Dec. 2019, pp. 1–4.
[30] W. Jia, S. Xu, Z. Liang, Y. Zhao, H. Min, S. Li, and Y. Yu, "Real-time automatic helmet detection of motorcyclists in urban traffic using improved YOLOv5 detector," *IET Image Process.*, vol. 15, no. 14, pp. 3623–3637, Jun. 2021.
[31] X. Chen and Q. Xie, "Safety helmet-wearing detection system for manufacturing workshop based on improved YOLOv7," *J. Sensors*, vol. 2023, pp. 1–14, May 2023.
[32] Z. Tong, Y. Chen, Z. Xu, and R. Yu, "Wise-IoU: Bounding box regression loss with dynamic focusing mechanism," 2023, *arXiv:2301.10051*.
[33] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li, X. Wang, and Y. Qiao, "InternImage: Exploring large-scale vision foundation models with deformable convolutions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 14408–14419.

JIANJUN CHEN received the B.S. degree in applied mathematics from Shanxi Datong University, Datong, China, in 2007, the M.S. degree in applied mathematics from Dalian Jiaotong University, Dalian, China, in 2010, and the Ph.D. degree in intelligent systems design engineering from Toyama Prefectural University, Toyama, Japan, in 2015. He is currently engaged in research on soft computing and its application to assistive systems for people with disabilities.

ZHUANG LI received the B.S. degree in information and computing science from Henan Polytechnic University, Jiaozuo, China, in 2021. He is currently pursuing the M.E. degree with the School of Computer Science, Jiangsu University of Science and Technology, Zhenjiang, China. He is also engaged in research on computer vision and its application.

JUNNING ZHU received the B.E. degree in computer science and technology from the Jiangsu University of Science and Technology, Zhenjiang, China, in 2022, where he is currently pursuing the M.E. degree with the School of Computer Science. He is also engaged in research on computer vision and its application.

XIBEI YANG received the master's degree from the Jiangsu University of Science and Technology, Zhenjiang, in 2006, and the Ph.D. degree in computer science and engineering from the Nanjing University of Science and Technology, Nanjing, China, in 2010. His current research interests include granular computing and rough set.