Enhanced Helmet Wearing Detection Using Improved YOLO Algorithm

Liuai Wu, Nannan Lu, Xiaotong Yao, and Yong Yang

***Abstract*—To address the accuracy limitations of existing safety helmet detection algorithms in complex environments, we propose an enhanced YOLOv8 algorithm, called YOLOv8-CSS. We introduce a Coordinate Attention (CA) mechanism in the backbone network to improve focus on safety helmet regions in complex backgrounds, suppress irrelevant feature interference, and enhance detection accuracy. We also incorporate the SEAM module to improve the detection and recognition of occluded objects, increasing robustness and accuracy. Additionally, we design a fine-neck structure to fuse features of different sizes from the backbone network, reducing model complexity while maintaining detection accuracy. Finally, we adopt the Wise-IoU loss function to optimize the training process, further enhancing detection accuracy. Experimental results show that YOLOv8-CSS significantly improves detection performance in general scenarios, complex backgrounds, and for distant small objects. YOLOv8-CSS improves precision, recall, mAP@0.5, and mAP@0.5:0.95 by 1.67%, 5.55%, 3.38%, and 5.87%, respectively, compared to YOLOv8n. Our algorithm also reduces model parameters by 21.25% and computational load by 15.89%. Comparisons with other mainstream object detection algorithms validate our approach's effectiveness and superiority.**

***Index Terms*—Object detection, Deep learning, Computer vision, YOLO, Construction safety**

I. Introduction

With the continuous improvement of China's infrastructure and the flourishing civil engineering industry, on-site safety has become fundamental for worker protection. In high-risk industries like construction, steel, and coal mining, safety helmets are crucial personal protective equipment that effectively prevent head injuries and reduce accident risk [1]. National regulations mandate that construction workers wear safety helmets on construction sites [2]. However, in practice, some workers do not wear safety helmets correctly or at all, leading to accidents. Currently, manual supervision or video surveillance is typically used on construction sites to check if workers are wearing safety helmets correctly. However, due to supervisors' limited attention spans, missed inspections frequently occur. Therefore, relying solely on supervisors to monitor helmet usage cannot meet safety demands on construction sites. Recently, with the advancement of computer vision technology, many scholars have extensively researched safety helmet detection. Traditional detection algorithms use the Support Vector Machine (SVM) algorithm [3] and Histogram of Oriented Gradient (HOG) features to detect human bodies first, then identify helmets based on their color [4]. However, the complexity of the actual operating environment and the small size of safety helmets reduce the robustness and generalization of traditional detection methods.

As deep learning technology advances, researchers have started applying deep convolutional networks to object detection. Deep learning-based object detection approaches are mainly divided into two-stage and one-stage methods. These network models, with their various architectures and strong learning capabilities, can efficiently extract feature information from images for object classification and localization. Two-stage object detection algorithms, such as R-CNN [5] and Faster R-CNN [6], segment images into multiple regions and generate candidate boxes, which are then classified or regressed upon. For example, Xu Shoukun et al. [7] increased the anchor number in Faster R-CNN, optimizing the model to address issues like large target size disparities and occlusions, thereby achieving high accuracy in safety helmet detection. However, these methods suffer from slow speed. The other category includes one-stage detection algorithms, such as the YOLO [8] series and SSD [9] series. Shi Hui et al. [10] improved the accuracy and generalization of safety helmet detection in YOLOv3 through methods such as network structure enhancement, model compression, optimization of non-maximum suppression algorithms, and multi-scale detection. However, in complex scenarios involving small targets, occlusions, and dense crowds, these algorithms do not perform satisfactorily. Yang Yongbo [11]. replaced the backbone network in YOLOv5s with MobileNetv3, used Diou-NMS instead of NMS, and added the CBAM attention mechanism, successfully reducing the model size and computational load but compromising accuracy. Overall, compared to traditional methods, the aforementioned deep learning-based safety helmet detection methods have shown improvements in detection speed and accuracy, yet there remains room for enhancement.

To address these issues, this paper proposes a safety helmet detection algorithm named YOLO-CSS, based on YOLOv8 with optimizations integrating attention mechanisms and multi-scale feature fusion. This algorithm achieves high detection accuracy while consuming fewer computational resources. This work makes the following primary contributions:

1) Reducing model complexity and improving detection accuracy by fusing features from various-sized maps in the backbone network using the SLIM-NECK structure.
2) Enhancing the focus on small targets and enhancing

Manuscript received March 29, 2024; revised August 5, 2024.

Liuai Wu is an associate professor at the School of Electronics and Information Engineering, Lanzhou Jiaotong University, Lanzhou, 730070, China.(e-mail: yuukichen0717@gmail.com)

Nannan Lu is a postgraduate student at the School of Electronics and Information Engineering, Lanzhou Jiaotong University, Lanzhou, 730070, China. (corresponding author: e-mail: lnn46013@gmail.com).

Xiaotong Yao is a professor at the School of Electronics and Information Engineering, Lanzhou Jiaotong University, Lanzhou, 730070, China.(e-mail: lnn7@qq.com)

Yong Yang is a postgraduate student at the School of Electronics and Information Engineering, Lanzhou Jiaotong University, Lanzhou, 730070, China.(e-mail: 4mystudyspace@gmail.com)

feature extraction and fusion capabilities by integrating the Coordinate Attention (CA) module into the feature fusion network (Neck).

3) Introducing the Multi-Head Attention Network Module (SEAM) from YOLO-FaceV2, enabling the model to handle occluded target scenes effectively.
4) The model's bounding box regression performance is successfully increased by replacing the original CIoU loss function with the Wise-IoU loss function.
5) Comparisons utilizing the SHWD dataset show that the proposed model outperforms the state-of-the-art models in detection performance, exhibiting superior robustness in real-world scenarios and different working environments.

II. Related Work

A. YOLOv8

The newest model in the YOLO series, YOLOv8, is useful for a number of applications, including object tracking, instance segmentation, and image classification. As the depth and width of the network increase, so do the model parameters and computational load for YOLOv8, which is split into YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. Depending on the application scenarios they are using, users can select the right network model. Having the fewest parameters and the least computing load among them, the YOLOv8n model retains a relatively quick detection speed and good accuracy. Figure 1. shows the YOLOv8 network structure.

The YOLOv8n detection network comprises four primary components: Input, Backbone, Neck and Head.

The Backbone of YOLOv8n mirrors that of YOLOv5, yet the C3 module is supplanted by the C2f module, grounded on the CSP concept. This configuration enables YOLOv8 to learn residual features, maintaining a lightweight model while acquiring richer gradient flow information. The popular SPPF [12] module, which employs three sequential 5x5 max-pooling layers followed by concatenation of each level, is still utilized at the conclusion of the backbone. This ensures accurate detection of objects at various scales while maintaining a lightweight model.

In the Neck, YOLOv8 continues to employ the PAN-FPN [13] feature fusion method, which strengthens the fusion and utilization of feature layer information at different scales. Compared to YOLOv5, there are two major improvements in the Head. First, the previous anchor-based detection method has been replaced with anchor-free [14] detection based on center detection algorithms. The anchor-free approach eliminates the need for manually designed anchor boxes, requiring only regression of the target center points and dimensions on feature maps of different scales. This reduces computational time and resources and avoids missed or duplicate detections caused by unreasonable anchor settings. Second, a mainstream decoupled-head [15] structure is utilized as the final detection head. The decoupled head can better handle semantic information at different scales and resolutions. By separating pixel-level predictions from feature extraction, it can better utilize semantic information between low-level and high-level features, improving network performance.

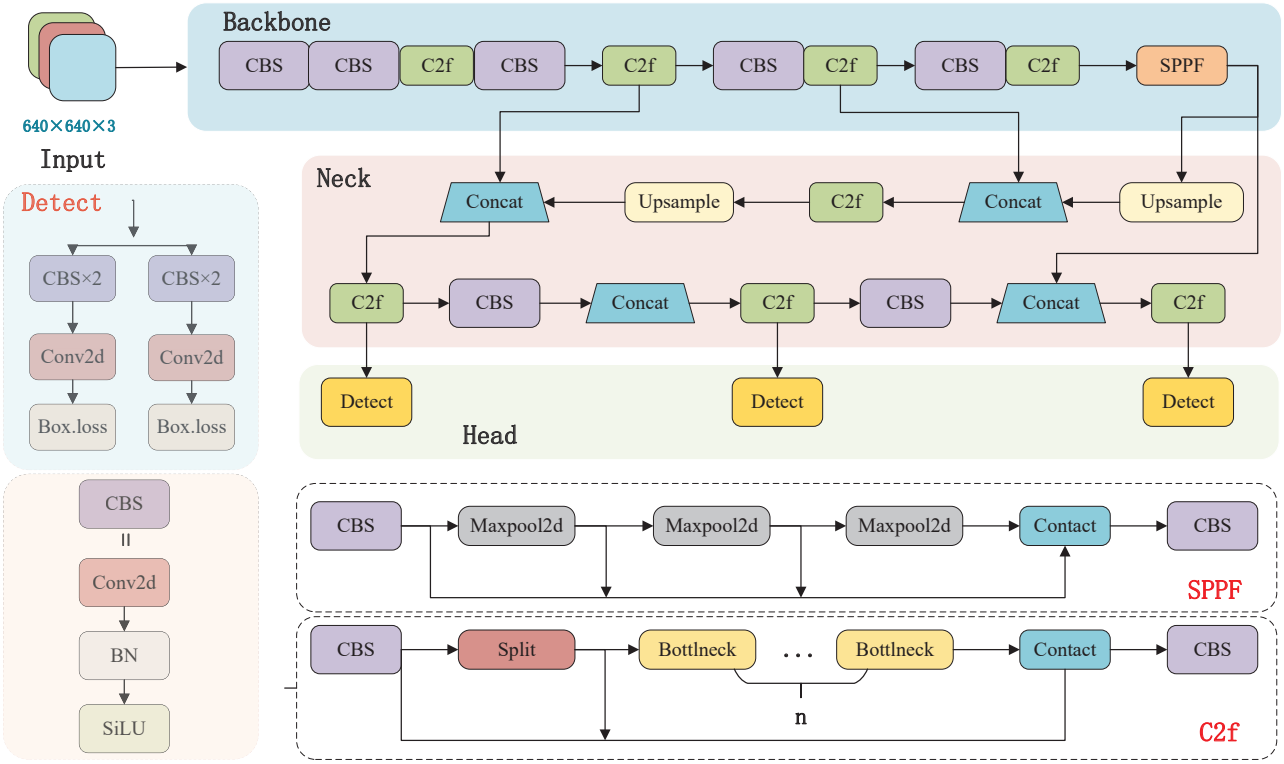


Fig. 1. YOLOv8 network architecture.

B. Coordinate Attention Mechanism

The attention mechanism is a powerful tool that is highly flexible in structure and aids networks in learning more discriminative feature representations. This mechanism can be easily integrated into the core structure of algorithms. One of the most prominent attention mechanisms is SE [16] (Squeeze-and-Excitation) attention, representing channel attention. This mechanism can explicitly traverse different dimensions to extract crucial inter-channel information. Another attention mechanism is CBAM [17] (Convolutional Block Attention Module), which represents spatial attention. CBAM leverages semantic dependencies between spatial and channel dimensions in feature maps, establishing cross-channel spatial information.

In complex environments like construction sites, target detection often faces challenges such as occlusion, background interference, and low image quality. These issues make it difficult for YOLOv8-based detection models to effectively extract target features, especially in tasks like detecting human behaviors. To address these problems, this paper introduces the Coordinate Attention (CA) mechanism [18] into the core structure of the YOLOv8 baseline model. This mechanism helps the network focus on extracting personnel features from images, highlighting important information in complex backgrounds. Especially in tasks like detecting safety helmets on workers, it can more accurately identify targets.

By integrating position data into channel attention, CA enables the network to concentrate on broader areas without incurring substantial computational expenses. Channel attention decomposes into a one-dimensional feature encoding process, aggregating features along two spatial directions separately, in contrast to traditional channel attention processes that solely focus on inter-channel information encoding while ignoring positional information. This method preserves exact location information in one spatial direction while capturing long-range interdependence in the other. The resulting feature maps are then separately encoded into position- and direction-sensitive attention maps, which are then complementarily applied to the input feature map to improve object representation. Figure 2 displays the structure of the CA.

The X Avg Pool and Y Avg Pool in the diagram correspond to the coordinate information embedding procedure. Using pooling kernels with predetermined dimensions, encoding is carried out for the input X along the horizontal and vertical coordinate directions. Equation 1 provides the output equation for the c channel at a height of h.

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i < W} x_c(h, i) \tag{1}$$

Equation 2 provides the output expression for the c channel, which has a width of w.

$$z_c^{\mathrm{w}}(\mathrm{w}) = \frac{1}{H} \sum_{0 \le j < H} x_c(j, \mathrm{w}) \tag{2}$$

Subsequently, the feature maps derived from the width and height orientations are joined together. The dimensions of these are then reduced to C/r, where C is the number of channels and r is the reduction ratio, by feeding them into

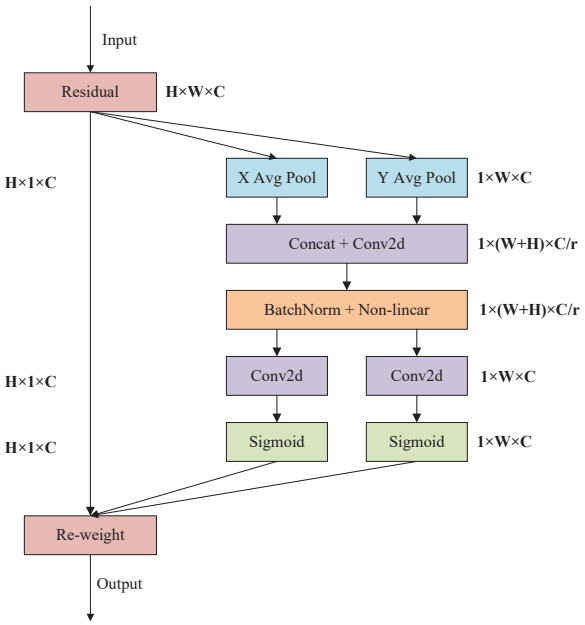


Fig. 2. CA module.

a common convolutional module that uses 1×1 convolutional kernels. Then, as indicated by Equation 3, the batch-normalized feature maps F_1 are run through the Sigmoid activation function to produce a feature map f.

$$f = \delta\left(F_1\left(\left[\mathrm{z}^h, z^{\mathrm{w}}\right]\right)\right) \tag{3}$$

Afterwards, 1×1 convolutional kernels are used to convolve the feature map f along its original directions of width and height, yielding feature maps with the same number of channels as the original. The attention weights for the height and breadth directions are derived by passing through a sigmoid activation function, as indicated by Equations 4 and 5.

$$\mathrm{g}^h = \sigma\left(F_h\left(f^h\right)\right) \tag{4}$$

$$\mathrm{g}^{\mathrm{w}} = \sigma\left(F_{\mathrm{w}}\left(f^{\mathrm{w}}\right)\right) \tag{5}$$

The height and width directions of the input feature map's attention weights are then determined by these computations. Finally, by performing a weighted multiplication on the initial feature map, as shown by Equation 6, the feature map with attention weights in both directions is obtained.

$$y_c(i, j) = x_c(i, j) \times \mathrm{g}_c^h(i) \times \mathrm{g}_c^{\mathrm{w}}(j) \tag{6}$$

To better demonstrate the improvement in network performance brought by the CA structure, this paper uses CAM (Class Activation Mapping) [19]for visual comparison of images. CAM visualizes the location of targets by generating heatmaps. After conducting GradCAM [20] tests on three types of networks (with CA added, with CBAM added, and the original YOLOv8 structure), the obtained heatmaps for target recognition are shown in Figure 3. It is clear that compared to the original YOLOv8 structure and the structure with CBAM added, the addition of the CA structure results in higher heatmap values for safety helmets, with more precise positions, demonstrating a more reliable effect.

(a) Original (b) YOLOv8n (c) YOLOv8n+CBAM (d) YOLOv8n+CA

Fig. 3. Heatmaps were compared following the incorporation of two distinct attention processes.

C. SLIM-NECK structure

Real-time object detection is challenging for large models. Although lightweight models with many depthwise separable convolution layers can improve speed, they often fail to meet accuracy requirements. Therefore, lightweight design has become an effective method for balancing speed and accuracy, reducing computational costs and allowing object detectors to find the optimal balance. To increase accuracy, the YOLOv8n algorithm makes use of C2f modules and ordinary convolutions. On the other hand, speed is decreased and model parameters are increased. The SLIM-NECK structure [21] integrates features from different-sized maps in the backbone network, improving detection accuracy and minimizing complexity while maintaining accuracy.

Depthwise separable convolution (DSC) [22] has greatly increased detection speed in lightweight models such as Xception [23], ShuffleNets [24], and MobileNets [25]. However, this enhancement often comes at the expense of accurate detection Li et al. [21] introduced GSConv to overcome the accuracy loss with depthwise separable convolution. This technique combines depthwise separable convolution data with information from standard convolution using Shuffle. GSConv uses ordinary convolution to downsample the input first, then depthwise convolution, as Figure 4 illustrates. The two convolutions' outputs are then concatenated and jumbled.

VoV-GSCSP is a module that is based on GSConv and the Cross-Stage Partial Network (CSP). In the neck network, GSConv and VoV-GSCSP modules form a cross-stage partial network similar to residual blocks. In this structure, feature maps from previous and subsequent layers are concatenated and then subjected to convolutional operations to avoid information loss and gradient vanishing. In the VoV-GSCSP network, GSConv replaces traditional convolution, and two GSConv modules are concatenated, while VoV-GSCSP replaces the C3 structure in the neck network. These components aim to reduce computational complexity while enhancing accuracy. Figure 5(a) illustrates the GS bottleneck structure, and Figure 5(b) showcases the VoV-GSCSP structure.

Two key improvements in the SLIM-NECK design are the single aggregation module VoV-GSCSP and the lightweight convolution algorithm GSConv. While maximizing the maintenance of inter-channel connections, GSConv lowers time complexity. The inference process of lightweight detection

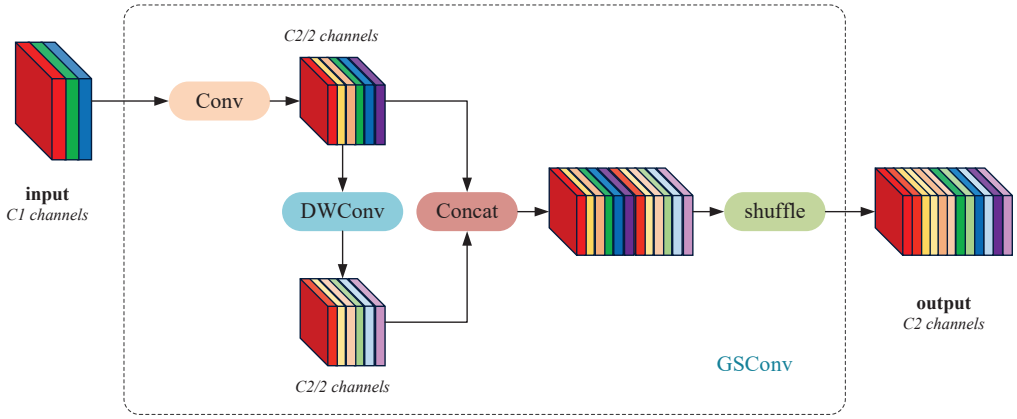


Fig. 4. Structure diagram for GSConv. Three layers make up the "Conv" block: the activation layer, the batch normalization layer, and the convolution layer. The Depthwise Separable Convolution (DSC) operation is represented by the blue-marked "DWConv" in this example.

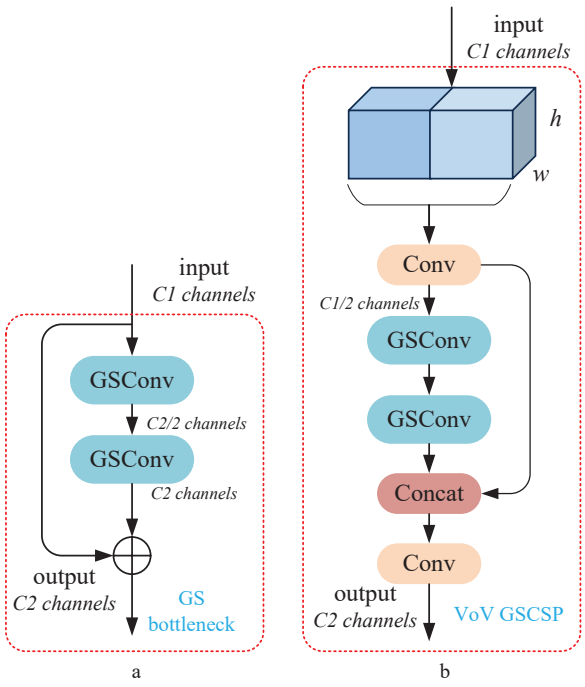


Fig. 5. (a) GS bottleneck structure (b) VoV-GSCSP structure.

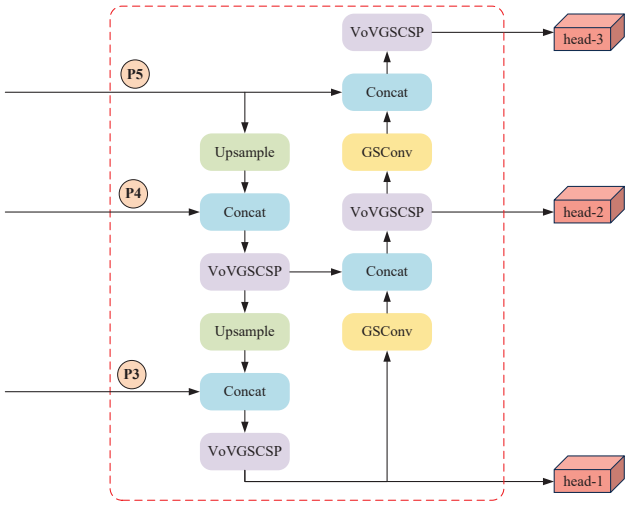


Fig. 6. SLIM-NECK embedded in YOLOv8n structure.

models is accelerated by using VoV-GSCSP in place of conventional CSP. The SLIM-NECK module is a feature fusion module that is intended for use in object detection tasks. By lowering network characteristics and computing load, it seeks to improve speed and efficiency. This module increases speed and efficiency by first adding low-dimensional feature mappings to the input, then extracting richer semantic information through convolutional processes. The SLIM-NECK structure of YOLOv8 is shown in Figure 6.

D. SEAM Attention Module

In the context of safety helmet detection, inter-class occlusion can lead to alignment errors, local aliasing, and missing features. To address these issues, we introduce a multi-head attention network, namely the SEAM [26] module (see Figure 7).

This module aims to achieve multi-scale helmet detection, emphasize the helmet area in the image, and de-emphasize the background area. The SEAM module begins with depthwise separable convolution with residual connections, which operates channel by channel. While this reduces the number of parameters and learns the importance of different channels, it neglects the inter-channel information relationships. To mitigate this, we combine the outputs of the depthwise convolutions using pointwise (1×1) convolutions. Subsequently, a two-layer fully connected network fuses the channel information, strengthening the connections between all channels. By learning the relationships between occluded and unobstructed helmets, this model compensates for the loss in occlusion scenarios. The logits produced by the fully connected layer are processed using an exponential function, expanding the value range from $[0,1]$ to $[1,e]$, which provides a monotonic mapping relationship and enhances tolerance to positional errors. Finally, the SEAM module's

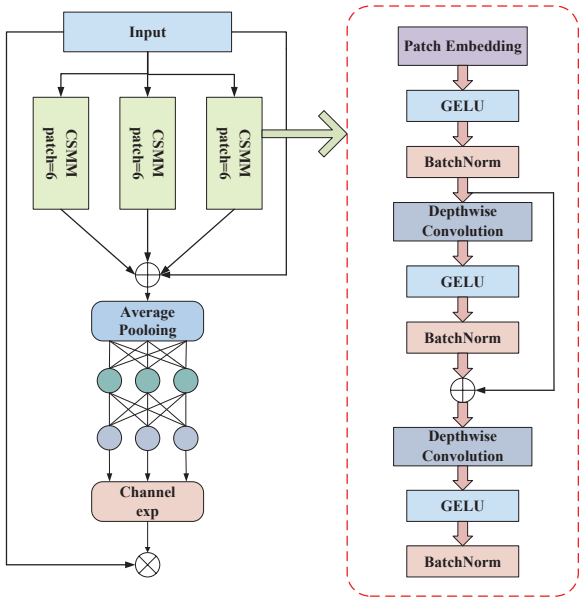


Fig. 7. Detect-SEAM network structure.

output is used as attention and is multiplied by the original features, enabling the model to handle helmet occlusion more effectively.

E. Wise-IoU

The original YOLOv8 model employs a regression loss comprising DFL Loss [27] + CIoU [28] Loss. The formula for the CIoU loss function is as follows:

$$L_{CIOU} = L_{IOU} + \frac{\sigma\left(b, b'\right)}{\left(\mathrm{w}^c\right)^2 + \left(\mathrm{h}^c\right)^2} + av \tag{7}$$

$$L_{IOU} = 1 - \frac{|b \cap b'|}{|b \cup b'|} \tag{8}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{\mathrm{w}'}{h'} - \arctan\frac{\mathrm{w}}{h}\right)^2 \tag{9}$$

$$a = \frac{v}{L_{IOU} + v} \tag{10}$$

Here, w, h, and b represent the width, height, and center coordinates of the predicted box, while w', h', and b' correspond to the ground truth box's coordinates. The Euclidean distance between the center locations of the predicted and ground truth boxes is denoted by σ. The variables h^c and w^c indicate the height and width of the minimal enclosing rectangle for both the ground truth and predicted boxes.

During training, class imbalance is not considered, and the CIoU loss function presents challenges in characterizing relative aspect ratios. Consequently, the Wise-IOU [29] loss function is introduced in this work. The loss function assesses anchor box quality with outliers using a dynamic, non-monotonic focus mechanism when the quality of the training data annotations is low. By reducing the penalty on geometric factors when there is a high overlap between the predicted and target boxes, the loss function can achieve better generalization with fewer training treatments. Therefore, Wise-IoU v3 employs a dynamic non-monotonic FM mechanism and two-layer attention mechanisms .

$$f_{BBRL} = \left(1 - \frac{W_t H_t}{S_u}\right) \exp\left\{\frac{(x_p - x_{gt})^2 + (y_p - y_{gt})^2}{\left(W_g^2 + H_g^2\right)^*}\right\} \cdot \gamma \tag{11}$$

$$\gamma = \frac{\beta}{\delta \alpha^{\beta - \delta}} \tag{12}$$

Anchor box quality is shown by the anticipated box's anomalous degree, denoted by β. A lower degree of anomaly indicates a better quality anchor box. We create non-monotonic focus numbers, which enable smaller gradient increases to be applied to projected boxes with large anomaly values. With this method, negative gradient effects on subpar training samples are successfully mitigated. The hyperparameters are α and β. Figure 8 illustrates the implications of additional parameters.

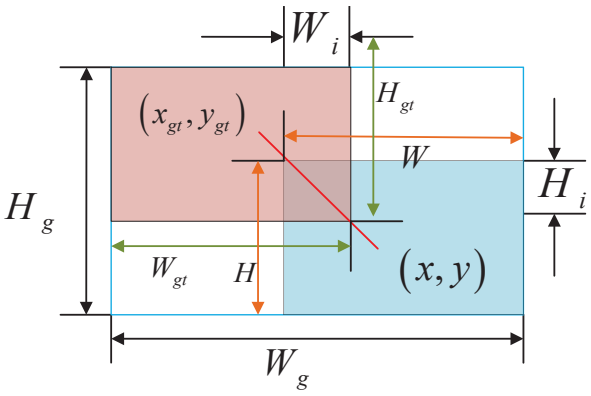


Fig. 8. Schematic diagram of calculation parameters.

The coordinates of the ground truth box are represented by x_{gt} and y_{gt}, while the coordinates of the predicted box are denoted by x_p and y_p. The width and height of the two boxes are indicated by the corresponding H and W values, respectively. The following can be deduced: $S_u = wh + w_{gt}h_{gt} + W_iH_i$.

F. Proposed Algorithm

In YOLOv8n, we introduced the CA mechanism to enhance feature interaction and expressive capabilities at different levels. Incorporating the SEAM module, a multi-head attention network, enables the model to handle occlusion scenarios effectively. Adopting GSConv and the SLIM-NECK design reduces the model's computational and parameter load, improving operational efficiency. Figure 9 illustrates the modified model's general structure.

III. Result And Discussion

A. Dataset

The dataset used in this study is the open-source SHWD (Safety Helmet Wearing Dataset, https://github.com/njvisionpower/Safety-Helmet-Wearing-Dataset).

The SHWD dataset consists of 7581 640x640 photos taken under various conditions. Specifically, 5306 photos were utilized for training, 1516 for validation, and 759 for testing, following a 7:2:1 distribution. The photographs were subsequently randomly partitioned into training, validation, and test subsets. Through the use of LabelImg, two distinct

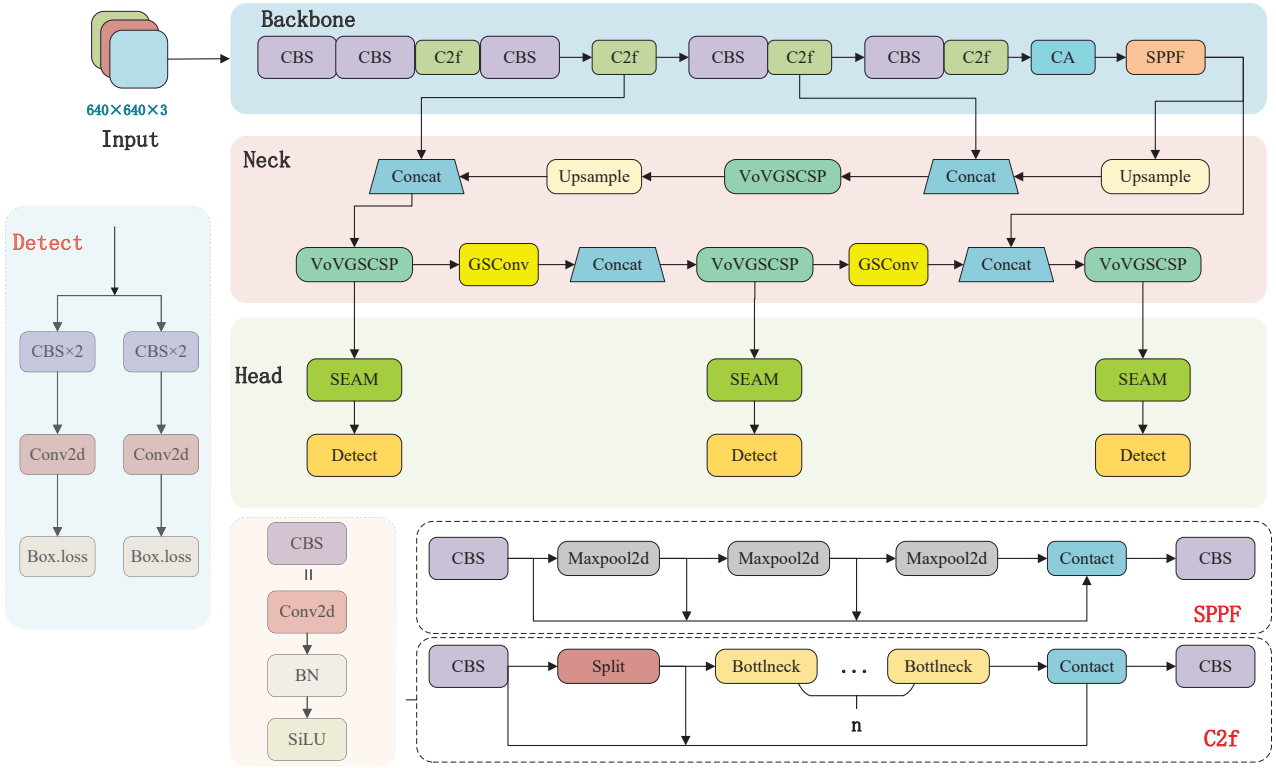


Fig. 9. YOLO-CSS network structure.

types of annotations were generated for the data: ”hat” for individuals wearing safety helmets and ”person” for those not wearing helmets. A representative sample of the SHWD dataset's instances is illustrated in Figure 10.

B. Experimental setup and evaluation metrics

The deep learning framework utilized in the experiment was PyTorch, and the operating system employed was Ubuntu 20.04. The default network model adopted was YOLOv8n. Table I presents the configuration details for the experimental environment as follows.

TABLE I
EXPERIMENTAL ENVIRONMENT CONFIGURATION

Parametres	Configuratio
System	Ubuntu 20.04
Deep Learning Framework Version	PyTorch 1.11.0
Python Version	Python 3.8
CPU	Intel(R) Xeon(R) Platinum 8255C
GPU	GeForce RTX 3090(24GB)
RAM	80 GB

Consistent hyperparameters were applied throughout all experiments. Table II lists the precise hyperparameters used during training.

In order to verify the enhanced YOLO-CSS's performance, this study employs multiple assessment criteria, such as Recall (R), Precision (P), Average Precision (AP), Mean Average Precision (mAP), Inference time(ms), and Model Size(MB). Equations 13 through 16 display the formulas for these metrics.

TABLE II
TRAINING HYPERPARAMETERS

Parameters	Value
Learning Rate	0.01
Image Size	640 × 640
Momentum	0.937
Optimizer	SGD
Batch Size	16
Epoch	150
Weight Decay	0.0005

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

$$P = \frac{TP}{TP + FP} \tag{14}$$

$$AP = \int_0^1 P\left(R\right) dR \tag{15}$$

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{16}$$

- True Positive (**TP**): A correctly detected object where the predicted bounding box matches the ground truth with sufficient overlap.
- False Positive (**FP**): An incorrectly detected object where the predicted bounding box does not match any ground truth or overlaps insufficiently.
- True Negative (**TN**): Correctly identified absence of an object, though not commonly used in object detection metrics.

Fig. 10. Image samples in SHWD dataset.

- False Negative (**FN**): A missed detection where the model fails to identify an object that is present in the ground truth.
- Average Precision (**AP**): A summary metric that calculates the area under the precision-recall curve for a specific class.
- mean Average Precision (**mAP**): The mean value of Average Precision across all classes in a dataset, providing an overall performance measure.

C. Model Training

To verify the effectiveness of YOLO-CSS, we trained the model using the same dataset and hyperparameters as YOLOv8n. We compared the loss and evaluation metrics curves of both algorithms, as shown in Figure 11 and Figure 12. The evaluation metrics include confidence loss, bounding box loss, class loss, precision, recall, mAP@0.5, and mAP@0.5:0.95, plotted against epochs.

Figure 11 shows that in both training and validation sets, the losses of YOLO-CSS are lower than YOLOv8n, and it converges faster.

Figure 12 shows that the mAP@0.5 of YOLOv8n stabilizes around 0.82 after approximately 20 epochs, reaching 0.915 after 150 epochs.

In contrast, the mAP@0.5 of YOLO-CSS reaches 0.908 at 22 epochs and achieves 0.948 eventually. Similarly, the mAP@0.5:0.95 of YOLOv8n reaches 0.571 after 150 epochs, while YOLO-CSS achieves 0.632. In terms of precision and recall, YOLO-CSS consistently outperforms YOLOv8n, demonstrating better performance.

Overall, the results indicate that YOLO-CSS outperforms YOLOv8n in various metrics, achieving higher detection accuracy and convergence speed.

D. Ablation Experiments Result And Discussion

We employed popular loss functions including GIoU [30], CIoU, DIoU [31], EIoU [32], and Wise-IoU in our studies to examine the effects of these functions on YOLOv8n.

Table III displays the training accuracies following the replacement of the YOLOv8n's original CIoU loss with various loss functions. Based on the results, Wise-IoU outperforms other loss functions in terms of precision, recall, mAP@0.5, and mAP@0.5:0.95. Bounding box regression factors, such as aspect ratios, overlap regions with predicted boxes, and the gap between expected and predicted boxes, are the major components of traditional object detection loss functions. The CIoU used in YOLOv8n exhibits ambiguity in describing relative aspect ratios and does not address sample balance during training, leading to slow convergence and large fluctuations in predicted frames. In contrast, Wise-IoU successfully reduces gradient vanishing, guaranteeing a more stable training procedure. Furthermore, Wise-IoU improves object detection accuracy by providing weighting factors that help it better capture the relative spatial relationships between anticipated and ground truth boxes.

We also conducted ablation studies employing the techniques utilized in this research: Coordinate Attention (CA) module, the SEAM module, the SLIM_NECK, and the Wise-IoU loss. Each of these methods was separately incorporated into Yolov8n for evaluation. The outcomes of the ablation studies are eummarized in Table IV. Based on these findings, we derived the following conclusions:

1) YOLO combined with CA can achieve a high mAP, but blindly adding attention modules increases the parameter count, resulting in model redundancy. The use of SLIM_NECK significantly reduces the model parameters, making it more lightweight. Meanwhile, directly incorporating the SEAM module does not effectively improve model accuracy, but the SEAM network more effectively addresses occlusion by weakening the background and highlighting the target area. Overall, our proposed model maintains high accuracy while significantly reducing the parameter count.
2) In order to explore the specific improvements each module brings to the detection of targets wearing and not wearing safety helmets in the dataset, we conducted tests on the SHWD dataset, using the original YOLOv8n model as the baseline, and then gradually added the CA, SEAM, and SLIM_NECK modules. The P-R curves of their respective experimental results are shown in the Figure 13. The larger the area enclosed

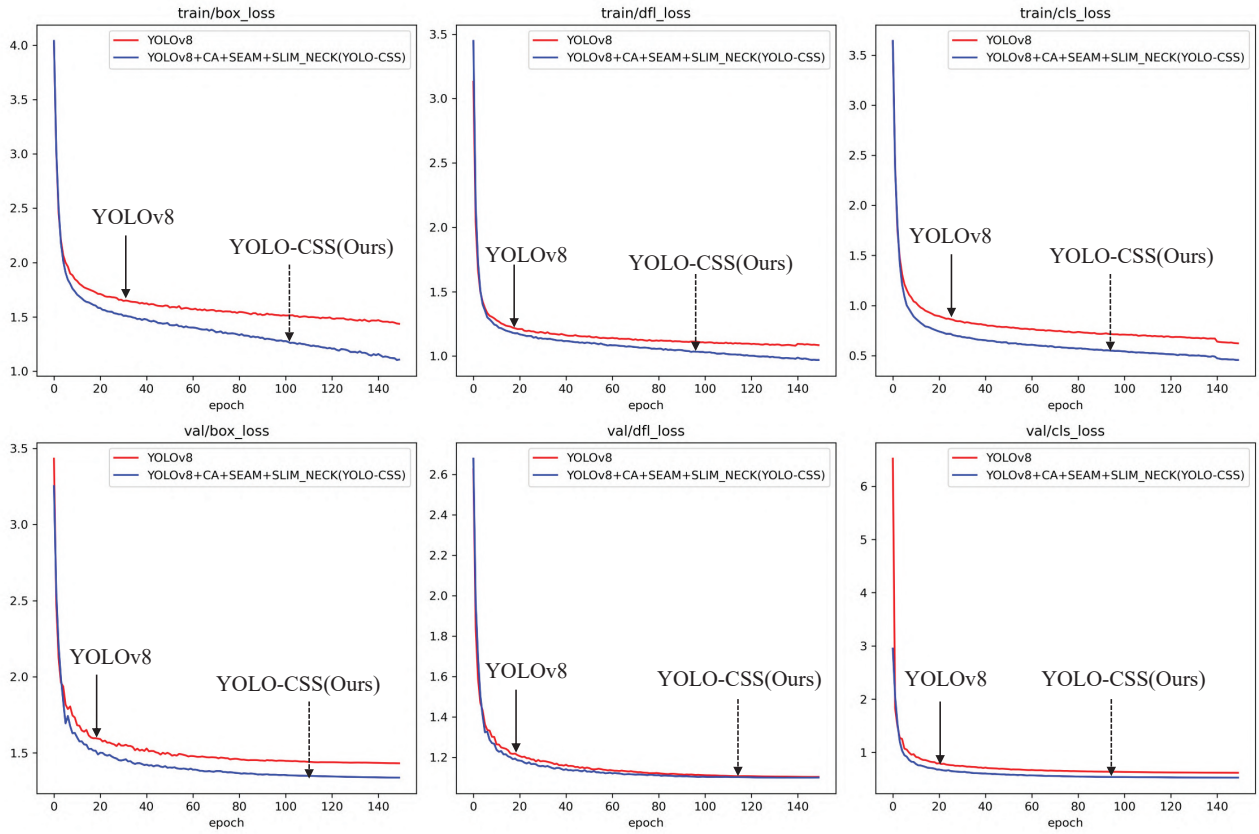


Fig. 11. Loss changes of each model.

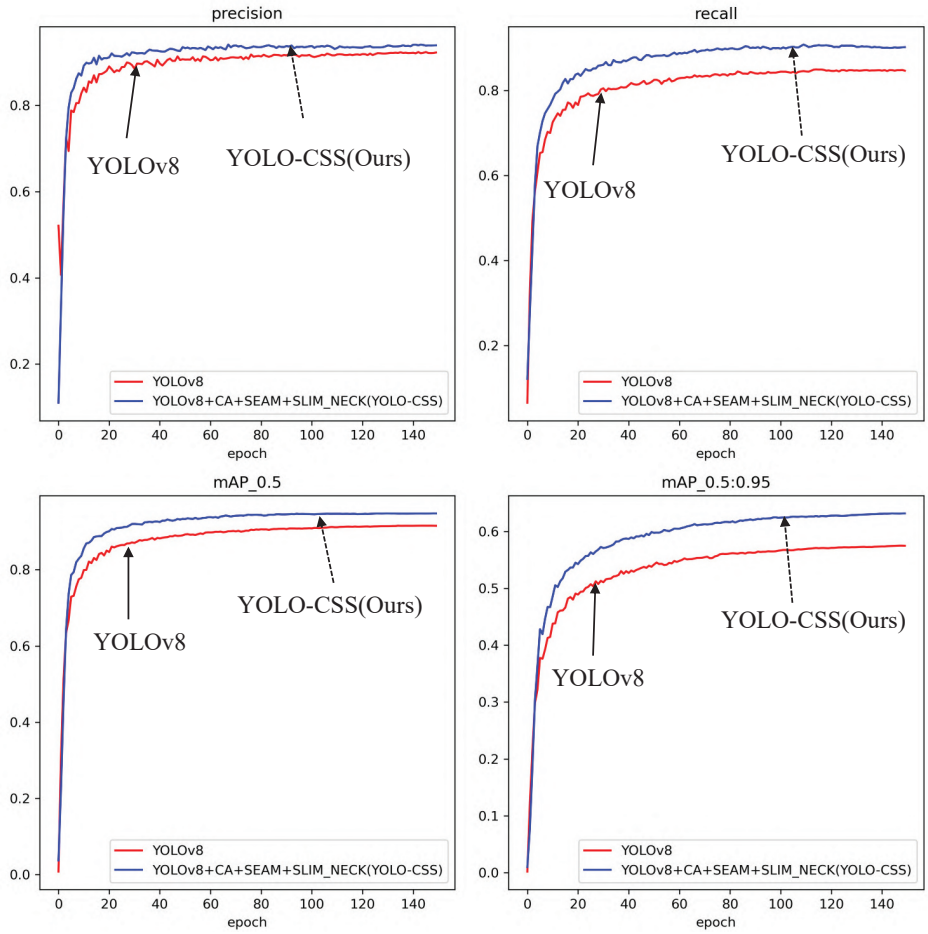


Fig. 12. Changes in the four indicators of each model.

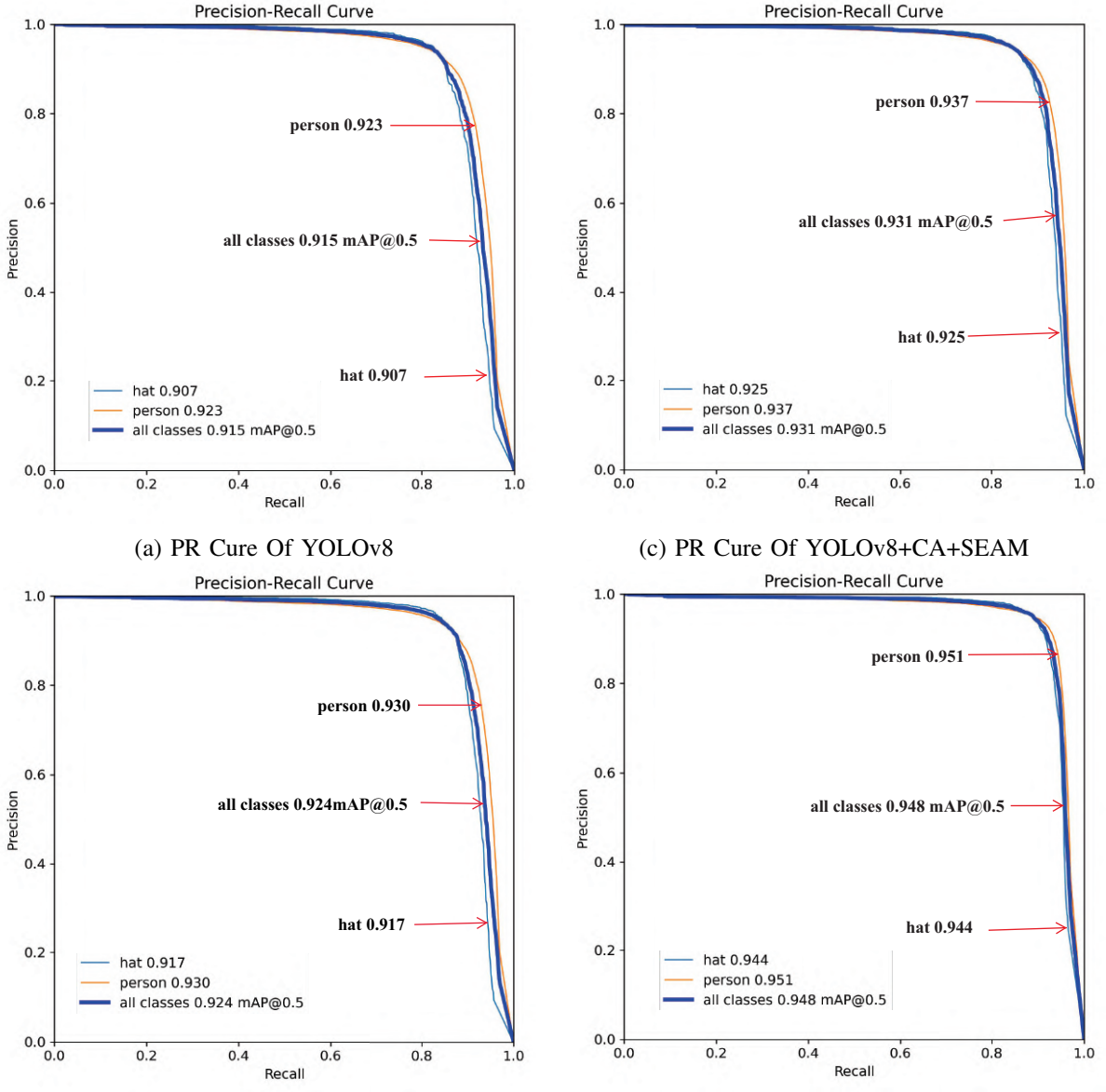
by the P-R curve and the coordinate axes, the higher the AP value, indicating better model performance. As shown in Figures, the improved model in this study has higher detection accuracy for both targets wearing safety helmets and those not wearing them, clearly outperforming other models.

3) To more intuitively compare the detection effects after adding various modules, Figure 14 shows the comparison result of mAP@0.5:0.95 metrics for the model after sequentially adding each module.

E. Comparison with other algorithms

Comparative studies were carried out to confirm the efficacy of the suggested object detecting system. A comparison was made between YOLO-CSS and various object identification models, including SSD, YOLOv3, YOLOv5s, YOLOv5l, YOLOv7, YOLOv8n, YOLOv8l, and Faster-RCNN. Similar datasets and experimental conditions were used in comparative studies. Mean average precision (mAP), weight size, model size, and floating-point operations per second (GFLOPs) were calculated and compared. A table called Table V displays the comparison results.

From Table V, it is evident that the mAP@0.5 metric of YOLO-CSS reached 94.89%, a significant improvement over the SSD and Faster-RCNN detection algorithms by 21.3% and 16.24%, respectively. Additionally, YOLO-CSS demonstrated considerable improvements over other YOLO series detection algorithms. Compared to the baseline model YOLOv8n used in this paper, YOLO-CSS achieved a 3.38% improvement. Although the improvement over the best-performing YOLOv8l algorithm was only 1.74%, but the parameters of YOLOv81 are several times that of YOLO-CSS, requiring more computation, which leads to a significant decrease in inference speed. This further demonstrates the superiority of the YOLO-CSS algorithm.



(a) PR Cure Of YOLOv8

(c) PR Cure Of YOLOv8+CA+SEAM

(b) PR Cure Of YOLOv8+CA

(d) PR Cure Of Our Proposed YOLO−CSS model

Fig. 13. A comparison of the PR curves for various models on the SHWD dataset with an IoU threshold set to 0.5.

TABLE III
COMPARISON OF EXPERIMENTAL RESULT WITH DIFFERENT LOSS FUNCTIONS

Loss Function	Precision(P)/%	Recall(R)/%	mAP@0.5/%	mAP@0.5:0.95/%
CIOU	0.904	0.866	0.908	0.581
DIOU	0.896	0.860	0.904	0.573
GIOU	0.902	0.863	0.899	0.570
EIOU	0.895	0.869	0.911	0.581
Wise-IOU	**0.906**	**0.871**	**0.914**	**0.583**

TABLE IV
ABLATION EXPERIMENT

CA	SEAM	SLIM_NECK	Wise−IoU	Precision(P)/%	Recall(R)/%	mAP@0.5/%	mAP@0.5:0.95/%	Size(Mb)
				92.27	86.65	91.51	57.48	16.23
✓				92.38	86.97	92.80	58.73	23.98
	✓			92.36	86.77	91.89	57.55	18.37
		✓		91.98	86.01	91.22	57.14	11.52
			✓	92.16	86.49	91.79	57.51	16.33
✓	✓			92.14	86.59	92.35	58.48	12.76
✓	✓	✓		92.61	87.08	93.11	59.71	13.07
✓	✓	✓	✓	**93.94**	**90.20**	**94.89**	**63.35**	13.53

TABLE V
THE COMPARISON BETWEEN THE MODEL PROPOSED IN THIS PAPER AND OTHER MODELS

Algorithm	Backbone	Params/M	Size/Mb	mAP@0.5/%	mAP@0.5:0.95/%	GFLOPs	Inference time(ms)
SSD	VGG16	43.85	98.53	71.59	35.25	59.67	87
Faster-RCNN	ResNet50	58.16	126.25	78.65	35.59	226.56	118
YOLOv3 [33]	DarkNet53	61.92	116.89	87.25	52.86	71.67	83
YOLOv5s	CSPDarkNet	7.06	13.85	89.41	54.97	17.20	27
YOLOv5l	CSPDarkNet	18.65	127.65	91.12	57.52	41.65	68
YOLOv7 [34]	CSPDarkNet	5.75	35.13	90.25	56.52	105.63	12
YOLOv8n	C2f	13.88	16.23	91.51	57.48	33.52	27
YOLOv8l	C2f	89.73	60.28	93.15	59.70	67.46	95
YOLO-CSS(Ours)	C2f with CA	11.01	29.53	**94.89**	**63.35**	25.79	19

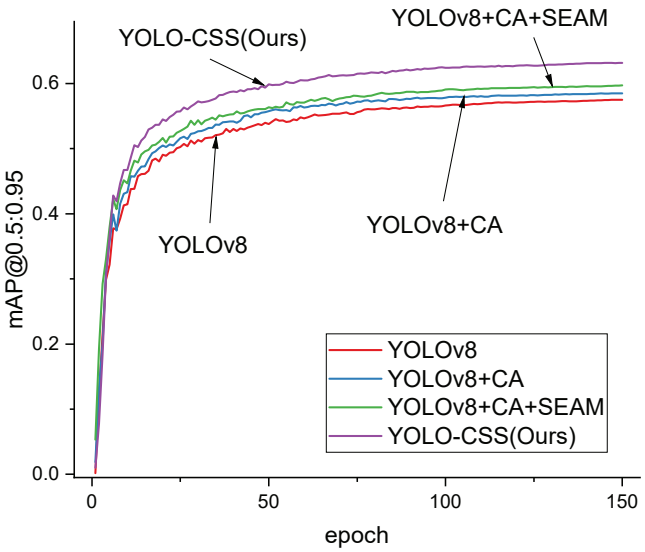


Fig. 14. Ablation experiment result.

F. Visualization results and discussion

To demonstrate the advantages of our algorithm, we selected various scenarios for comparative experiments, including complex scenes, densely populated targets, distant small targets, low-light conditions, and occluded targets. As shown in Figures 15 to 20, each image includes (a) the original real-world scene, (b) the detection results of the original YOLOv8 model, and (c) the detection results of the proposed YOLO-CSS model.

As shown in Figure15, in scenarios with complex backgrounds, dense targets, and occlusion, YOLO-CSS detects all targets, while YOLOv8n has one missed detection, indicated by a yellow triangle. YOLOv8n exhibits severe missed detections in complex backgrounds, but YOLO-CSS accurately identifies small targets with higher confidence. As shown in Figure 16, in scenarios with complex backgrounds and distant small targets, YOLOv8n misidentifies the crane arm as a hat, indicated by a yellow triangle, whereas YOLO-CSS detects all targets correctly. Figure 17 shows a low-light nighttime scene where YOLOv8n fails to detect the target inside the vehicle, while YOLO-CSS correctly identifies it. In Figure 18 and Figure 19, both scenarios involve distant small targets and occlusion. The SEAM module in YOLO-CSS enables better detection of occluded targets. Therefore, YOLO-CSS correctly detects all targets, while YOLOv8n performs poorly in detecting occluded targets, as indicated by the yellow triangles. In Figure 20, both YOLOv8n and YOLO-CSS correctly detect all targets, but YOLO-CSS exhibits higher confidence, further demonstrating its superiority.

In summary, the proposed algorithm achieves significant performance improvement in detection compared to YOLOv8n by introducing and improving various modules. Notably, the algorithm excels in various scenarios, including real-world scenes, complex backgrounds, occluded targets, and distant small target detection. These improvements repre-

(a) (b) (c)

Fig. 15. Complex backgrounds and dense targets.

(a) (b) (c)

Fig. 16. Complex backgrounds and distant small targets.

(a) (b) (c)

Fig. 17. Small targets at low-light night.

(a) (b) (c)

Fig. 18. Distant small targets and occlusion.

(a) (b) (c)

Fig. 19. Small targets from an overhead perspective.

(a) (b) (c)

Fig. 20. Small targets with complex background.

sent substantial progress, ensuring excellent performance in diverse environments and effectively addressing challenges posed by complex environments and distant targets.

IV. Conclusion

Workplace safety is becoming increasingly dependent on helmet detection as deep learning technology advances. However, current helmet identification methods struggle to recognize small, obscured items and objects against complex backgrounds.We propose and implement YOLO-CSS, an enhanced algorithm, to address these challenges. Through ablation and comparison studies, we arrive at the following findings:

1) Introducing the CA (Coordinate Attention) module effectively improves accuracy. Experimental results show that the CA module outperforms the CBAM (Convolutional Block Attention Module) attention mechanism. CA focuses more on key areas and reduces interference from complex backgrounds.
2) Introducing the SLIM-NECK structure for feature fusion in the backbone network better integrates multi-scale features of targets with background information, significantly improving performance while reducing network size.
3) Adding the SEAM module enables multi-scale object detection, emphasizing target areas and reducing background influence, enhancing the model's ability to handle occluded objects.
4) Using the more effective Wise-IoU loss function enables the model to better handle occlusion and other challenges, significantly improving convergence speed and detection accuracy.

With dense and obscured objects, this enhancement lessens false positives and negatives. In conclusion, YOLOv8n-CSS performs better than YOLOv8n in every way, increasing mAP@0.5 by 3.38% to 94.89%. Furthermore, in situations with small items, dense objects, and complex settings, its detection performance outperforms other methods. Consequently, this approach satisfies the needs for precise and real-time helmet detection.

References

[1] J. H. Wu Jing, Liu Guifa, "Strengthen the supervision of special labor protection equipment such as safety helmets and promote the development of emergency industry," *China Safety Production*, vol. 14, no. 6, pp. 36–37, 2019. I

[2] Z. Jing, "Discussion on building safety management technology in smart cities," *Popular Standardization*, vol. 1, no. 23, pp. 84–86, 2021. I

[3] J. Platt, "Sequential minimal optimization : A fast algorithm for training support vector machines," *Microsoft Research Technical Report*, 1998. I

[4] L. Qirui, "Research and implementation of helmet video detection system based on human body recognition," Master's thesis, Chengdu: University of Electronic Science and Technology of China, 2017. I

[5] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2013. I

[6] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137–1149, 2015. I

[7] X. Shoukun, W. Yaru, G. Yuwan, L. Ning, Z. Lihua, and S. Lin, "Research on helmet wearing detection based on improved faster rcnn," *Computer Application Research*, vol. 37, no. 03, pp. 901–905, 2020. I

[8] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2015. I

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision*, 2015. I

[10] S. Hui, C. Xianqiao, and Y. Ying, "Improved helmet wearing detection method for yolo v3," *Computer Engineering and Applications*, vol. 55, no. 11, pp. 213–220, 2019. I

[11] Y. Yang and D. Li, "Lightweight helmet wearing detection algorithm of improved yolov5," *Computer Engineering and Applications*, vol. 58, no. 9, pp. 201–207, 2022. I

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, pp. 1904–1916, 2014. II-A

[13] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944, 2016. II-A

[14] A. Kirillov, R. B. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6392–6401, 2019. II-A

[15] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *ArXiv*, vol. abs/2107.08430, 2021. II-A

[16] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2017. II-B

[17] S. Woo, J. Park, J.-Y. Lee, and I.-S. Kweon, "Cbam: Convolutional block attention module," *ArXiv*, vol. abs/1807.06521, 2018. II-B

[18] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13 708–13 717, 2021. II-B

[19] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929, 2015. II-B

[20] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, pp. 336 – 359, 2016. II-B

[21] H. Li, J. Li, H. Wei, Z. Liu, Z. Zhan, and Q. Ren, "Slim-neck by gsconv: A better design paradigm of detector architectures for autonomous vehicles," *ArXiv*, vol. abs/2206.02424, 2022. II-C, II-C

[22] L. Sifre and S. Mallat, "Rigid-motion scattering for texture classification," *ArXiv*, vol. abs/1403.1687, 2014. II-C

[23] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807, 2016. II-C

[24] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6848–6856, 2017. II-C

[25] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *ArXiv*, vol. abs/1704.04861, 2017. II-C

[26] Z. Yu, H. Huang, W. Chen, Y. Su, Y. Liu, and X.-Y. Wang, "Yolo-facev2: A scale and occlusion aware face detector," *Pattern Recognit.*, vol. 155, p. 110714, 2022. II-D

[27] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," *ArXiv*, vol. abs/2006.04388, 2020. II-E

[28] Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo, "Enhancing geometric factors in model learning and inference for object detection and instance segmentation," *IEEE Transactions on Cybernetics*, vol. 52, pp. 8574–8586, 2020. II-E

[29] Z. Tong, Y. Chen, Z. Xu, and R. Yu, "Wise-iou: Bounding box regression loss with dynamic focusing mechanism," *ArXiv*, vol. abs/2301.10051, 2023. II-E

[30] S. H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. D. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 658–666, 2019. III-D

[31] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-iou loss: Faster and better learning for bounding box regression," in *AAAI Conference on Artificial Intelligence*, 2019. III-D

[32] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, "Focal and efficient iou loss for accurate bounding box regression," *ArXiv*, vol. abs/2101.08158, 2021. III-D

[33] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018. V

[34] C. Wang, A. Bochkovskiy, and H. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023, pp. 7464–7475. V