PSP-RTMDet: End-to-End Lightweight High-Precision Helmet Detector

Xueying Liao, Xingran Guo, Askar Rozi, Haizheng Yu, Abdukerim Haji

Abstract—**Industries like construction, mining and exploration enforce mandatory helmet regulations, leading to the widespread use of helmet detection algorithms in these fields. However, existing algorithms often need help with high parameter counts, complexity, poor real-time performance, and balancing accuracy and speed. We propose a new helmet detection algorithm based on Real-Time Object Detectors (RTMDet), PSP-RTMDet, to address the need for both high precision and lightweight real-time performance. The core innovation of this algorithm is the introduction of Partial Convolution (PConv), which replaces traditional convolution methods to reduce computational load and enhance the extraction of spatial features. The backbone network structure has been optimized using basic building blocks, and the global attention mechanism Parameter-Free Attention (SimAM) has been incorporated. This enhances the model's ability to extract critical features without adding extra parameters. To further minimize feature information loss for small targets during feature fusion and to boost their representation in the shallow feature map, we developed a new lightweight feature fusion module, Symmetric Positive Definite Convolution (SPD-RPAFPN), which improves the detection of dense helmet targets. We conducted experimental comparisons using the public Safety Helmet Wearing Dataset (SHWD). The results showed that our improved algorithm increased the total mAP-50 value by 3.88%, reaching 93.33%, compared to the baseline RTMDet model. Additionally, the number of model parameters and FLOPs was reduced by 9.33% and 12.44%, respectively. Moreover, compared to current mainstream algorithms such as YOLOv8, our method improved detection accuracy by 4.78%, demonstrating the PSP-RTMDet algorithm's effectiveness in helmet detection.**

Index Terms—**dense-detection, helmet-detection, RTMDet, lightweight-model.**

I. Introduction

ACCIDENTS at construction sites, such as falls from heights and object impacts, account for over 50% of fatalities. Safety helmets are crucial for protecting workers from these injuries, making the detection of helmet usage among construction personnel vital for site safety management and future intelligent construction platforms. Object detection, an essential aspect of computer vision research, identifies objects in images and provides information on their categories and positions. Safety helmet detection, a specific object detection application, is essential for intelligent monitoring systems at construction sites.

The single-stage safety helmet detection algorithm based on deep learning features a simple model structure and rapid detection speed, making it ideal for helmet detection tasks. Commonly used algorithms include the YOLO series, SSD, RetinaNet, EfficientDet, and RTMDet.

YOLO (You Only Look Once) [1] significantly improves detection speed. YOLOv3 is used for real-time safety helmet detection in complex scenarios [2]. By modifying the classifier for single-class detection, YOLOv3 achieves an 18-dimensional tensor output and an average detection speed of 35 frames per second. YOLOv5 offers high accuracy (92.4%) in detecting people with helmets in complex scenes. The SSD algorithm balances high accuracy and speed, using an improved VGG backbone to detect small objects with accuracy similar to Faster R-CNN efficiently but with speed comparable to YOLO [3]. RetinaNet addresses sample imbalance in single-stage detection with a weighted cross-entropy loss function, improving detection accuracy [4]. EfficientDet combines EfficientNet with a bi-directional feature pyramid network, enhancing generalization ability and addressing network depth, width, and resolution imbalances [5].

Another method integrates attentional mechanisms with Faster R-CNN [6], using a self-attention layer to capture global information and enhance feature richness. A YOLOv3-based algorithm incorporates an attention mechanism and a bidirectional feature pyramid for improved accuracy in helmet detection [7]. A pyramid attention network (PANet) is introduced to minimize false detections and reduce feature loss. Lyu et al. summarized and integrated the series with current improvement ideas to further advance the YOLO series to construct a new series: RTMDet [14]. This series enhances feature extraction, improves label assignment and data enhancement strategies, and balances computational efficiency and accuracy. RTMDet performs well across multiple tasks, including instance segmentation and rotating target detection, achieving state-of-the-art results. However, it must still work on balancing high accuracy and speed in practical helmet detection applications.

Despite the success of the algorithms above in safety helmet detection, challenges remain. Inaccurate bounding box localization poses difficulties in detecting miniature target safety helmets, leading to suboptimal performance and missed detections. YOLOv3 faces challenges in detecting overlapping safety helmets in crowded environments. Applying the YOLOv5 model to a small embedded safety hel-

Manuscript received July 2, 2024; revised November 26, 2024.

This work was supported by the National Natural Science Foundation of China (Grant nos. 11761066) and the doctoral foundation of Xinjiang University (Grant nos. 62031224735).

Xueying Liao and Xingran Guo these authors contributed equally to this work

Xueying Liao is a graduate student at the School of Mathematics and System Sciences, Xinjiang University, Urumqi, China (e-mail: 107552203498@stu.xju.edu.cn).

Xingran Guo is a graduate student at the School of Mathematics and System Sciences, Xinjiang University, Urumqi, China (e-mail: 107552203480@stu.xju.edu.cn).

Askar Rozi is an associate professor at the School of Mathematics and System Sciences, Xinjiang University, Urumqi, China (corresponding author to provide e-mail: kar@xju.edu.cn).

Haizheng Yu is an associate professor at the School of Mathematics and System Sciences, Xinjiang University, Urumqi, China (e-mail: yuhaizheng@xju.edu.cn).

Abdukerim Haji is a professor at the School of Mathematics and System Sciences, Xinjiang University, Urumqi, China (e-mail: abdukerimhaji@sina.com).

met detection device requires further refinement of the network architecture [8]. The SSD algorithm's detection speed decreases significantly with increased complexity, making applying safety helmet detection tasks with high-speed requirements challenging. RetinaNet lacks real-time detection speed in helmet detection [9]. EfficientNet suffers from decreased detection speed due to increased network complexity. Faster R-CNN uses an attentional mechanism that requires additional memory for storing attentional weights, leading to increased parameters, reduced detection speed, and unsatisfactory accuracy in dense worker helmet detection. For detecting dense images, a larger and more powerful receptive field in the Backbone can better adapt to dense feature extraction, such as object detection and instance segmentation. This approach aids in capturing and constructing surrounding features of key image points [10]. However, the computational cost of dilated convolutions [11] and non-local blocks [12] is usually high, making real-time applications challenging and imposing various limitations. YOLOv7-DSE [13], an improved YOLOv7 model for complex target scenes, detects objects with large-scale differences but requires advanced hardware and significant memory usage, hindering its deployment in standard embedded systems.

Building on previous research, we identified several challenges, including significant model parameters, slow training speeds, inaccuracies in dense helmet detection, and issues with small targets, deformable helmets, and stacked scenarios.

To address the need for high accuracy and real-time performance in lightweight models, we propose a new PSP-RTMDet, which builds upon the RTMDet baseline model. The core idea of PSP-RTMDet involves introducing a new Partial Convolution block (PConv) to extract spatial features more efficiently by reducing redundant computations and memory accesses. The PConv convolution applies a feature extraction scheme to part of the input channels, leaving the rest unprocessed while maintaining the same number of channels in the input and output feature maps. This approach reduces the computational burden compared to traditional convolution methods. We also optimize the basic building blocks in the backbone network structure and introduce the global attention mechanism SimAM. SimAM, which combines channel and spatial attention mechanisms, derives accurate 3-D attention weights for the feature map without adding extra parameters. This mechanism improves the model's ability to extract compelling features by highlighting important neurons based on their discharge patterns, which enhances the extraction of crucial feature information. Finally, to reduce the loss of small target feature information and increase its fusion proportion in the shallow feature map, we construct a new lightweight feature fusion module, SPD-RPAFPN. SPD-RPAFPN modifies the traditional FPN feature pyramid by reversing the top-down path to a bottom-up approach. This adjustment, combined with SPD-Conv for downsampling, retains a higher proportion of small target feature information while integrating higher-order semantic information.

Our experiments demonstrate the effectiveness of these innovations. PSP-RTMDet achieves high precision and lightweight helmet detection capabilities.

Our contributions are outlined as follows:

1) Novel Partial Convolution Block (PConv): We introduce a novel PConv block that replaces the original convolutional layers. This block enhances the efficiency of spatial feature extraction by reducing redundant computation and memory usage.
2) Optimized Backbone Network: We optimize the backbone network by enhancing the basic building blocks and incorporating a lightweight global attention mechanism. This improves the model's focus on the target region.
3) Improved Feature Fusion: We enhance the network's ability to fuse features by increasing small target feature information integration. This improvement is achieved without adding to the model's parameters.
4) Extensive Testing: Our model extensively tests the publicly available SHWD dataset. It achieves a 3.88% improvement in average accuracy compared to the baseline RTMDet and a 4.78% improvement compared to the state-of-the-art YOLOv8. Additionally, our model reduces the number of parameters and FLOPs by 9.33% and 12.44%, respectively.

The rest of the paper is organized as follows: Section II discusses the baseline model RTMDet. Section III presents the innovative approach of PSP-RTMDet. Section IV covers the experimental results, and Section V concludes the paper.

II. Review of Baseline Model RTMDet

RTMDet is a high-performance, low-latency, single-stage targeting algorithm. It fully utilizes large kernel deep convolution and a dynamic soft-label assignment strategy. As shown in Fig. 1, the model's structure includes four main components: a data enhancement module, a backbone network module, a feature fusion module, and a prediction module.

A. Network Architecture Analysis

Single-image data enhancement techniques such as scale-transformed random cropping and auto-cropping are used in the data enhancement module. Mixed-class data enhancement methods, including Mosaic and Mixup, are also applied through strong-weak two-phase training. The Cache adjusts the enhancement strength to improve the efficiency of mixed data enhancement.

The image enters the backbone network module after passing through the data enhancement module. This module includes Conv convolution layers, CSP convolution layers, and SPPFBottleneck. The CSP convolution layer consists of 3 Convs, n-th CSPNextBlocks with residual connections, and a channel attention mechanism. CSPNextBlock includes one Conv and one large kernel depth-separable convolution (DWConv) [15].

The feature fusion module is based on the PAFPN structure, which integrates a Feature Pyramid Network (FPN) [16] and a Path Aggregation Network (PAN) [17]. This module fuses and reconstructs abstract semantic information and shallow features from different scales, achieving complete multi-scale feature fusion. The fused features are then input to the prediction module for classification and regression.

In the prediction module, the classification and regression branches are decoupled. A SepBNHead [18] is used to share

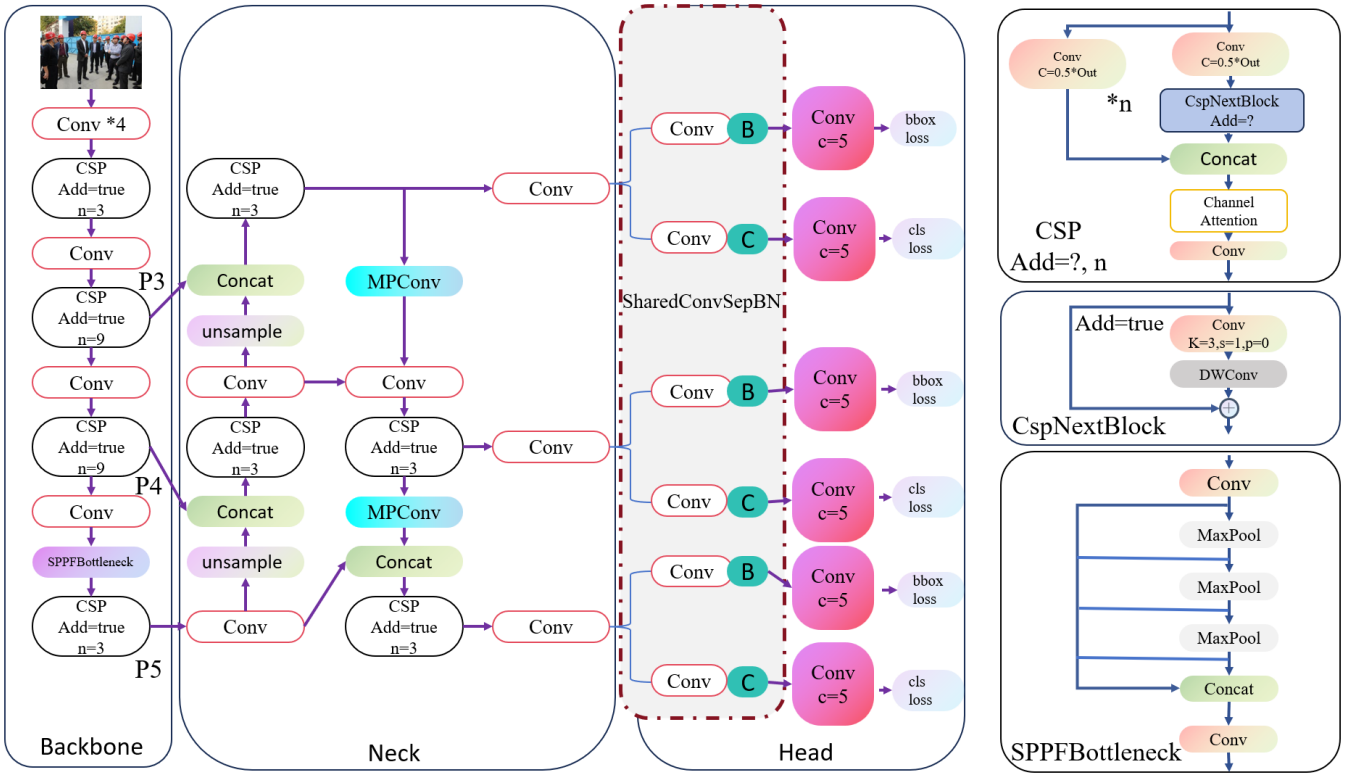


Fig. 1. RTMDet Network Structure

convolutional weights across different layers while independently calculating BatchNorm (BN) statistics to reduce the number of parameters. Finally, an unanchored frame algorithm generates the prediction frames. The final prediction value is obtained after applying non-maximum suppression and the loss function.

B. Label Assignment Strategy

To enhance the training of a single-object detection model for prediction tasks at various scales, RTMDet adopts a novel label assignment approach. This approach aligns with the boundaries of the actual annotated boxes for objects. Recent studies [19] have used dynamic label assignment rules as matching benchmarks. However, these methods are computationally expensive. To address this, RTMDet introduces a new soft label assignment rule, given by:

$$C = \alpha_1 C_{class} + \alpha_2 C_{\text{regress}} + \alpha_3 C_{\text{center}}\,, \tag{1}$$

where $\alpha_1 = 1$, $\alpha_2 = 3$, and $\alpha_3 = 1$ are the initial weight ratios for classification cost (C_{class}), regression cost (C_{regress}), and local prior cost (C_{center}), respectively. Previously, classification cost was calculated using binary labels. However, this approach includes bounding boxes with high classification scores but incorrect predictions. RTMDet addresses this issue by incorporating soft labels into the classification cost, as follows:

$$C_{class} = ce\left(p, y_{\text{soft}}\right) \times \left(y_{\text{soft}} - p\right)^2. \tag{2}$$

Here, ce denotes the cost mean weighting function, p is the probability of the correct value, and y_{soft} represents the soft-labeled value.

Inspired by GFL [20], RTMDet introduces a new mechanism. Soft labels are defined as the joint intersection (IoU) values between predicted boxes and ground truth boxes, which serve as training criteria for classification. This method reweights costs based on regression quality and resolves noise and match instability issues encountered in previous methods. Traditional IoU [21] can have values less than 1 for the best and worst matches, limiting the ability to distinguish between high- and low-quality matches. RTMDet uses the logarithmic rule of IoU as a regression cost to improve matching accuracy, enhancing the matching cost for low IoU values. RTMDet employs a soft centre rule rather than a fixed centre for regional costs, stabilizing matching between dynamic costs. The formulas are:

$$C_{\text{regress}} = -\log(\text{IoU}), \tag{3}$$

$$C_{\text{center}} = \alpha^{|X_{\text{pre}} - X_{gt}| - \beta}. \tag{4}$$

Here, α and β are parameter-matching values for a given soft centre region.

Recent research [22] highlights the use of cross-sample techniques [23] to enhance data. Despite their effectiveness, these techniques introduce challenges, such as the need to load multiple images, which slows down training. Additionally, generated samples may contain noise, affecting model performance. RTMDet proposes improved rules for MixUp and Mosaic with a caching effect. MixUp generates new training samples by linearly interpolating features and labels of two samples, enhancing model generalization. Mosaic stitches together multiple randomly selected images to help the model learn complex scenes. By using a caching mechanism, RTMDet reduces the time cost of blending images to that of single-image input, alleviating data loading

requirements. This rule is controlled by the cache length and output method, with larger values corresponding to the original non-caching rule and smaller values enabling repeated augmentation.

RTMDet will integrate PConv enhancements, SimAM attention, and a novel downsampling strategy for SPD-RPAFPN based on these improvements.

III. Proposed PSP-RTMDet

In this section, we present the improved structure of the article. Subsection III-A introduces the concept of PConv. Subsection III-B discusses the optimized backbone, while subsection III-C covers the improved feature fusion.

A. Fusion Embedding for Lightweight Convolutional PConv

The FasterNet convolutional neural network [24] introduces a new Partial Convolutional Block (PConv) that extracts spatial features more efficiently. This is achieved by simultaneously reducing redundant computations and memory accesses. PConv convolution applies a regular feature extraction scheme to some input channels while leaving the rest unprocessed. This approach maintains the same number of input and output feature map channels. In contrast, standard convolution is a feature extraction method that keeps the number of convolution kernel channels equal to the number of input feature map channels. The following section compares the performance of PConv convolution with standard convolution, as illustrated in Fig. 2.

For an input $I \in \mathbb{R}^{c\times h\times w}$, PConv uses c_n filters, while ordinary convolution (Conv) uses c filters. The memory usage of PConv is given by:

$$Pconv_{mem} = h \times w \times 2c_n + k^2 \times c_p c_n \approx h \times w \times 2c_n \quad (5)$$

The FLOPs for PConv are calculated as follows:

$$F_{Pconv} = h \times w \times k^2 \times c_p c_n \quad (6)$$

The memory usage of a Conv convolution is:

$$Conv_{mem} = h \times w \times 2c + k^2 \times cc^2 \approx h \times w \times 2c \quad (7)$$

The FLOPs for Conv are:

$$F_{conv} = h \times w \times k^2 \times c^2 \quad (8)$$

From the above equations (5), (6), (7), and (8), we derive:

$$\frac{Pconv_{mem}}{Conv_{mem}} \approx \frac{c_n}{c} \quad (9)$$

$$\frac{F_{Pconv}}{F_{conv}} = \left(\frac{c_n}{c}\right)^2 \quad (10)$$

PConv replaces standard convolution by convolving only the c_n channels. It does not aim to resist the un-convolved channels; instead, it uses the Fourier transform to extract unique features. This reduces memory usage and operational complexity, contributing to a lighter model. Additionally, PConv effectively extracts information from the feature map by retaining all feature channels through convolutional transformations and combining these with standard convolution to extract features.

B. Improvement of Basic Building Blocks CSP

This section focuses on the improvements made to the CSP section, explicitly introducing the SimAM global attention mechanism [25] and optimising parameter values in the CSPNextBlock.

1) SimAM Integration of Attention Mechanisms: Due to the wide variety of helmet images and the significant interference from complex backgrounds, it is essential to focus effectively on important regions. Attention mechanisms are widely used in deep learning to enhance feature extraction and reduce the dispersion of target information. However, most current attention mechanisms typically assign weights along the channel or spatial dimensions to improve model performance. Classical approaches, such as BAM [26] and CBAM [27], calculate 1-D channel weights and 2-D spatial weights, respectively. These methods then set hyperparameters for parallel or serial combinations to form reasonable global attention. Generally, these attention mechanisms do not realistically simulate the information selection process during visual processing and often introduce extra parameters into the model.

SimAM is a mechanism that combines channel and spatial attention, as shown in Fig. 3. It derives accurate 3-D attention weights for feature maps without adding additional parameters, enhancing the model's ability to extract compelling features. SimAM draws on neuroscience principles to differentiate the importance of feature information at various locations and successfully achieve attention. It suggests that information-rich neurons exhibit different firing patterns compared to surrounding neurons. Activated neurons tend to inhibit their neighbours, known as spatial inhibition. Thus, neurons with spatial inhibitory effects should be assigned higher importance.

To identify important feature information straightforwardly, SimAM defines an energy function that measures the linear separability of a single feature from all other features in the same channel. The energy function for each feature is defined as follows:

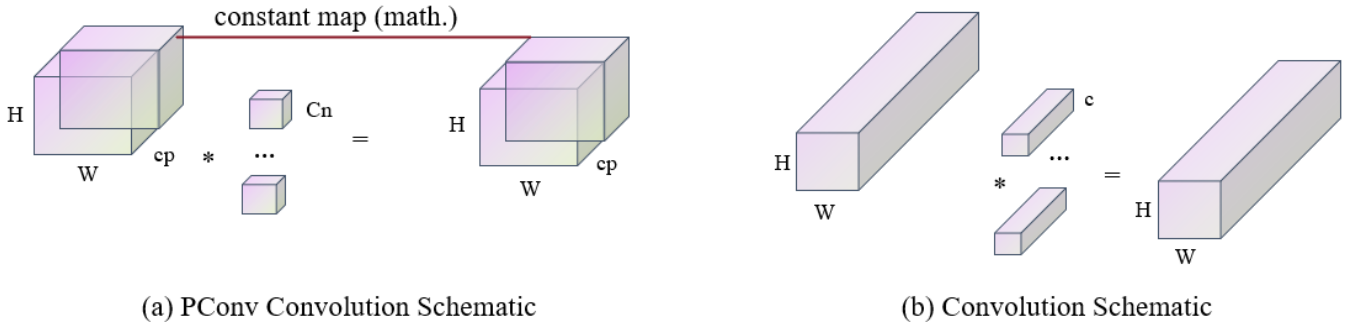
$$e_t(w_t, b_t, y, x_i) = \frac{1}{M-1}\left(\sum_{i=1}^{M-1}\left(1-(w_t x_i + b_i)\right)^2\right) + \left(1-(w_t t + b_t)\right)^2 + \lambda w_t^2 \quad (11)$$

In this equation, t and x_i represent the channel's target feature and other features, respectively. w_t and b_t are the linear transformation weight and bias of t, t is the spatial dimensionality ordinal, λ is a hyperparameter, and M is the total number of features in a single channel. The transformation weights and biases are expressed as follows:

$$w_t = \frac{2(t-u_t)}{(t-u_t)^2 + 2\sigma_t^2 + 2\lambda} \quad (12)$$

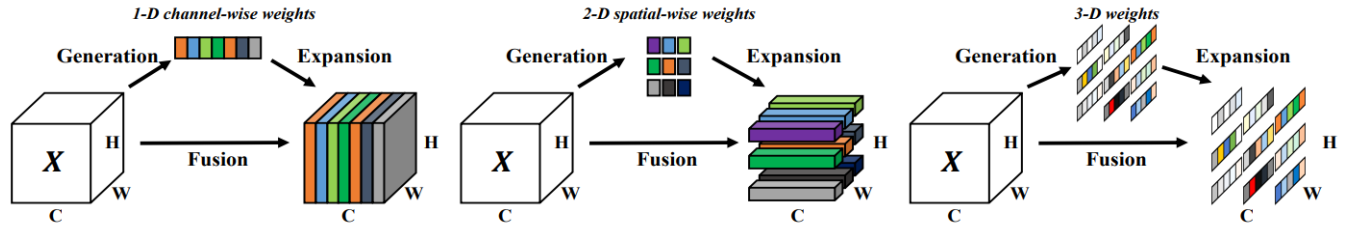
$$b_t = \frac{1}{2}(t+u_t)w_t \quad (13)$$

Here, $u_t = \frac{1}{M-1}\sum_{i=1}^{M-1} x_i$ and $\sigma_t^2 = \frac{1}{M-1}\sum_{i=1}^{M-1}(x_i - u_t)^2$ represent the mean and variance of all feature information except t. By calculating w_t, b_t, and the mean and variance of all features in the channel, the minimized energy formula is obtained as follows:



(a) PConv Convolution Schematic (b) Convolution Schematic

Fig. 2. Comparison of the Structure of PConv and Convolution



(a) Channel-wise attention (b) Spatial-wise attention (c) Full 3-D weights for attention

Fig. 3. Comparison of Different Attention Mechanisms

$$e_t^* = \frac{4(\sigma_t^2 + \lambda)}{(t - \mu_t)^2 + 2\sigma_t^2 + 2\lambda} \tag{14}$$

In this case, $\mu_t = \frac{1}{M}\sum_{i=1}^{M} t_i$ is the mean value of the feature information t. According to Equation (8), as μ_t increases, e_t^* decreases, indicating that the feature information t is less important in the channel. This implies that the greater the difference between feature information and background features, the more crucial it is for image processing. Therefore, the importance of each feature can be determined by $1/e_t^*$.

2) CSPNextBlock: While the basic building block of traditional CSPDarkNet consists of various convolution operations, the current YOLOv7-v8 models incorporate RepVGG's reparameterized convolution module into their basic units. This reparameterized module enhances performance without increasing inference computation by employing a multi-branch structure during training, which is then fused into a single branch for inference. However, it faces challenges such as high training costs and difficulties with quantization, necessitating alternative methods to compensate for quantization errors.

To reduce training costs and improve the feature extraction capability of the basic unit, RTMDet draws inspiration from RepLKNet [28]. Key concepts include the efficiency of large kernel deep convolution, the importance of constant skip connections for large kernel convolutions, the role of small kernel reparameterizations in addressing optimization issues, and the effectiveness of large kernel convolutions for downstream tasks. Additionally, large kernel convolution remains effective for small images. As a result, the CSPNextBlock is constructed with the structure shown in Fig. 4.

However, the introduction of large kernel depth-separable convolution (DWConv) in the basic unit increases the overall depth of the model compared to CSPDarkNet, resulting in slower inference speeds. To address this issue, the number of basic units at different resolution layers is adjusted based on the experimental techniques used in ConvNeXt [29]. After conducting several experiments, it was determined that using a 5×5 DWConv with C2-C5 adjusted to 3-6-6-3 blocks allows the model to achieve the best balance between computational efficiency and accuracy in the helmet target detection task.

Finally, to verify the effectiveness of the improved basic building block, the enhanced algorithm is compared with YOLOv8 and RTMDet using the Grad-CAM [30] visualization technique to assess its focus on the target area. The results, shown in Fig. 5, demonstrate that the improved module enhances the network's ability to concentrate on the target area and extract critical feature information.

C. Improved Neck Module

Since the feature map shrinks during the downsampling process in RTMDet's feature fusion, fine-grained information is lost, resulting in decreased detection accuracy at low resolutions. SPD-Conv [31] is used in the feature fusion module instead of the original downsampling module to address this issue. SPD-Conv reduces the loss of fine-grained information while enhancing feature details. It incorporates a traditional image transformation technique [32] within the CNN, consisting of a space-to-depth layer and a non-stepwise convolutional layer.

The SPD-Conv process is illustrated in Fig. 6. The specific steps are as follows: First, the feature map $X(S \times S \times C_l)$ is divided into N parts along the channel dimension, resulting in sub-feature maps of size S/N by S/N, with the number of channels unchanged. These sub-features are then combined along the channel direction to form the feature map X_1, so $X(S \times S \times C_l) \rightarrow X_1\left(\frac{S}{N} \times \frac{S}{N} \times N^2 C_l\right)$. Finally, X_1 is convolved using a non-strided convolutional layer ($stride = 1$) to obtain the feature map X_2, i.e., $X_1\left(\frac{S}{N} \times \frac{S}{N} \times N^2 C_l\right) \rightarrow X_2\left(\frac{S}{N} \times \frac{S}{N} \times C_2\right)$.

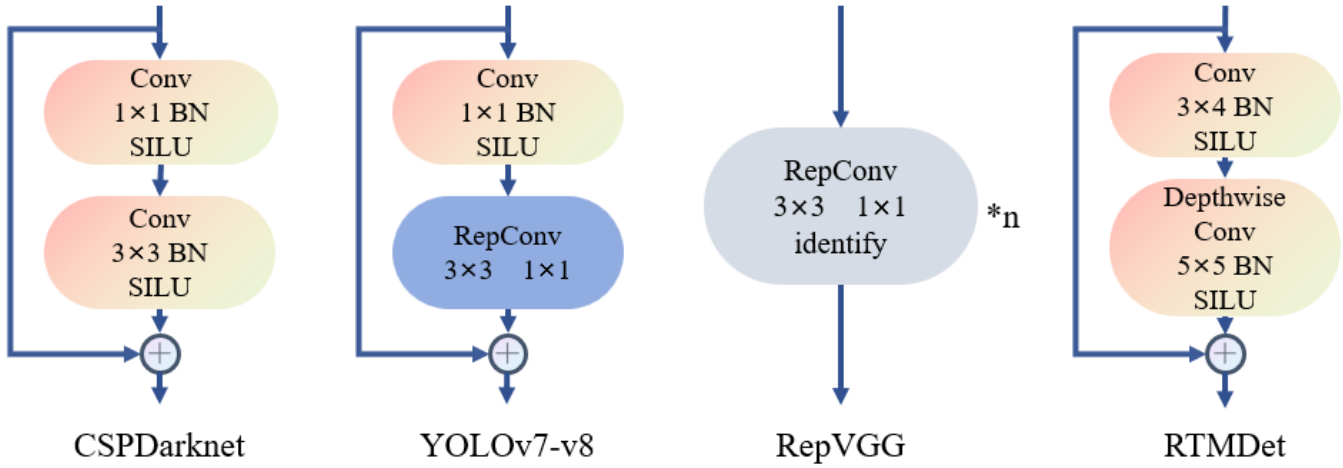


Fig. 4. Several Different Comparisons of Basic Building Blocks

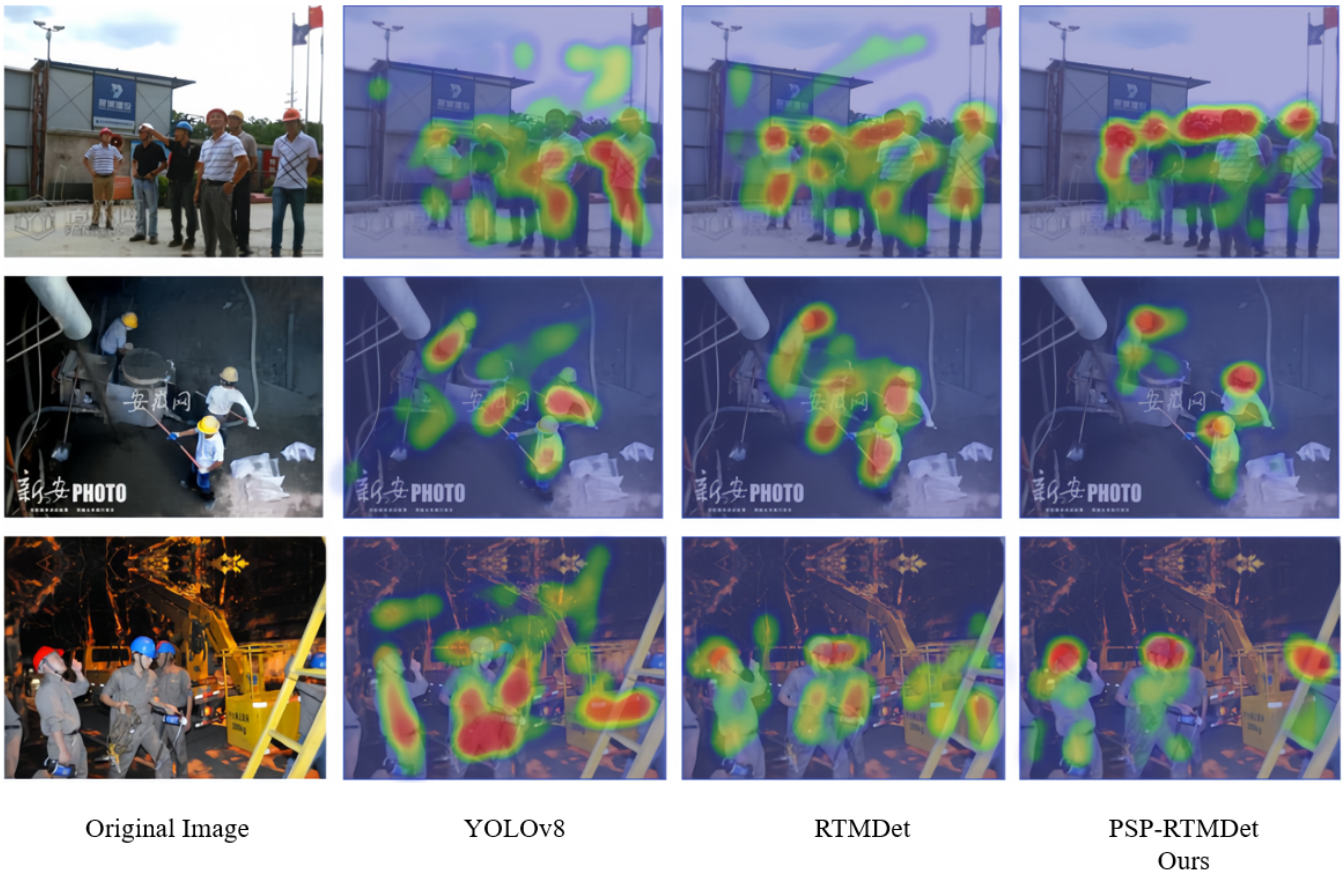


Fig. 5. Attention Effect Comparison Heat Maps

The helmet image features a smaller, more densely arranged target size set against a more extensive background. As convolutions increase in the backbone network, the receptive field gradually enlarges. This enhances the richness of higher-order semantic information but also results in the loss of small target feature information. The small feature information in the shallow feature map is crucial for such targets. It is essential to minimize unnecessary processing loss and increase the proportion of feature information fusion.

The SPD-RPAFPN network structure is illustrated in Fig. 7. By reversing the direction of the traditional feature pyramid network (FPN), the top-down path is converted to a bottom-up approach: (P5 >P4 >P3) becomes (P3 >P4 >P5). With this change in the feature pyramid direction, fusion from the shallow layer to the deeper layer occurs earlier. However, downsampling is required to reduce the size, which can lead to the premature loss of small target feature information. To address this, SPD-Conv is employed for downsampling. After feature pyramid fusion, although the feature information in the shallow feature map P3 remains unchanged, the deeper feature maps P4 and P5 have already incorporated the shallow feature information. When the subsequent lightweight network adjusts the direction of the current response back to top-down, the P4 and P5 maps, which have fused shallow feature information, can flow back to the P3 layer. This allows the shallow feature

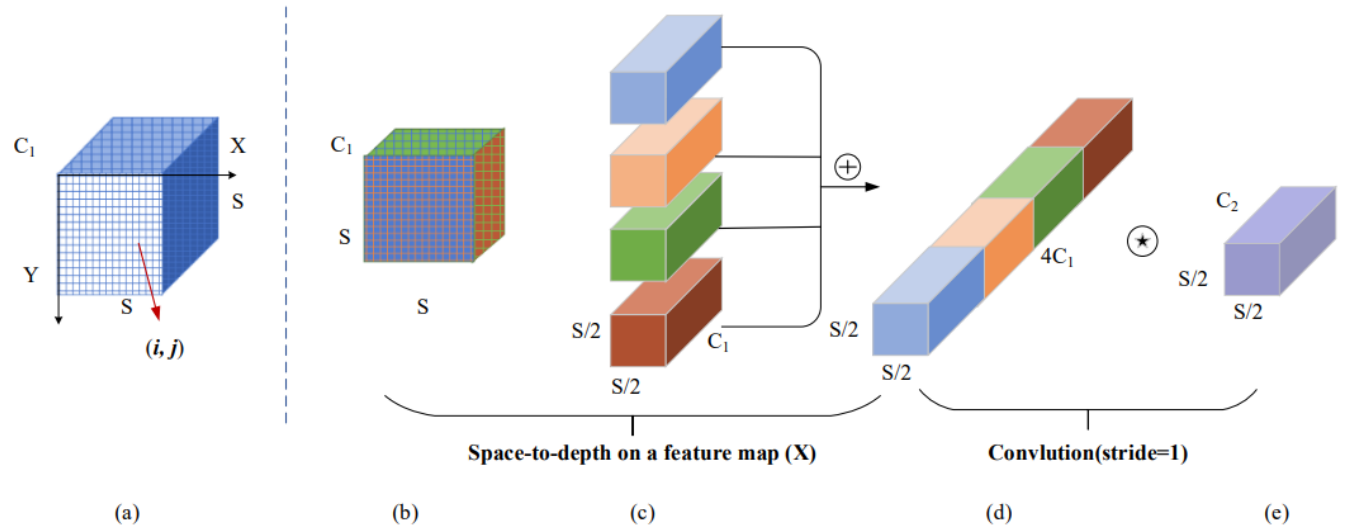


Fig. 6. SPD-Conv Network Architecture Diagram

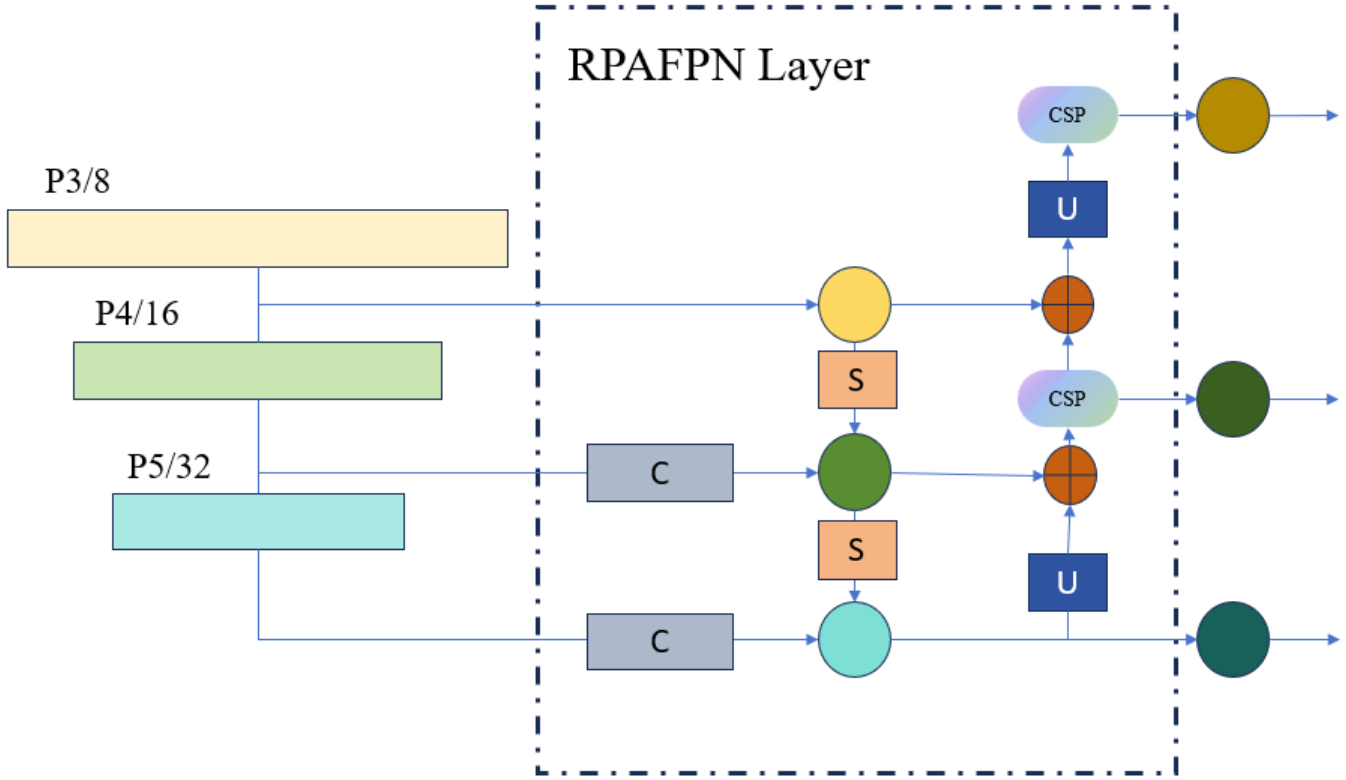


Fig. 7. SPD-RPAFPN Network Architecture Diagram

map to represent a more significant proportion of the feature information for small targets while enabling the fusion of higher-order semantic information.

This entire process enhances the role of shallow feature maps in detecting small targets within helmet image-dense scenes. Additionally, it reduces the number of parameters and FLOPs by one-eighth and one-fifth, respectively, compared to the original RTMDet. This reduction is due to the integration of PConv's lightweight convolutional blocks, early dimensionality reduction of feature maps, and decreased reliance on the basic building block CSP, resulting in a lighter and more efficient overall module.

The original RTMDet structure is modified in the improved algorithm by replacing traditional convolutional blocks with the PConv lightweight convolutional block. The basic building block in the backbone network is optimized, and the lightweight global attention mechanism SimAM is introduced. A new fusion method, SPD-RPAFPN, is constructed in the feature fusion module, enhancing the processing of feature information and improving helmet targets' detection capability. The improved RTMDet network structure is shown in Fig. 8.

IV. Experiments

In this module, we conduct extensive experiments on the models. Subsection IV-D focuses on validation ablation experiments and baseline model comparisons for each innovative module. Subsection IV-E presents a comprehensive

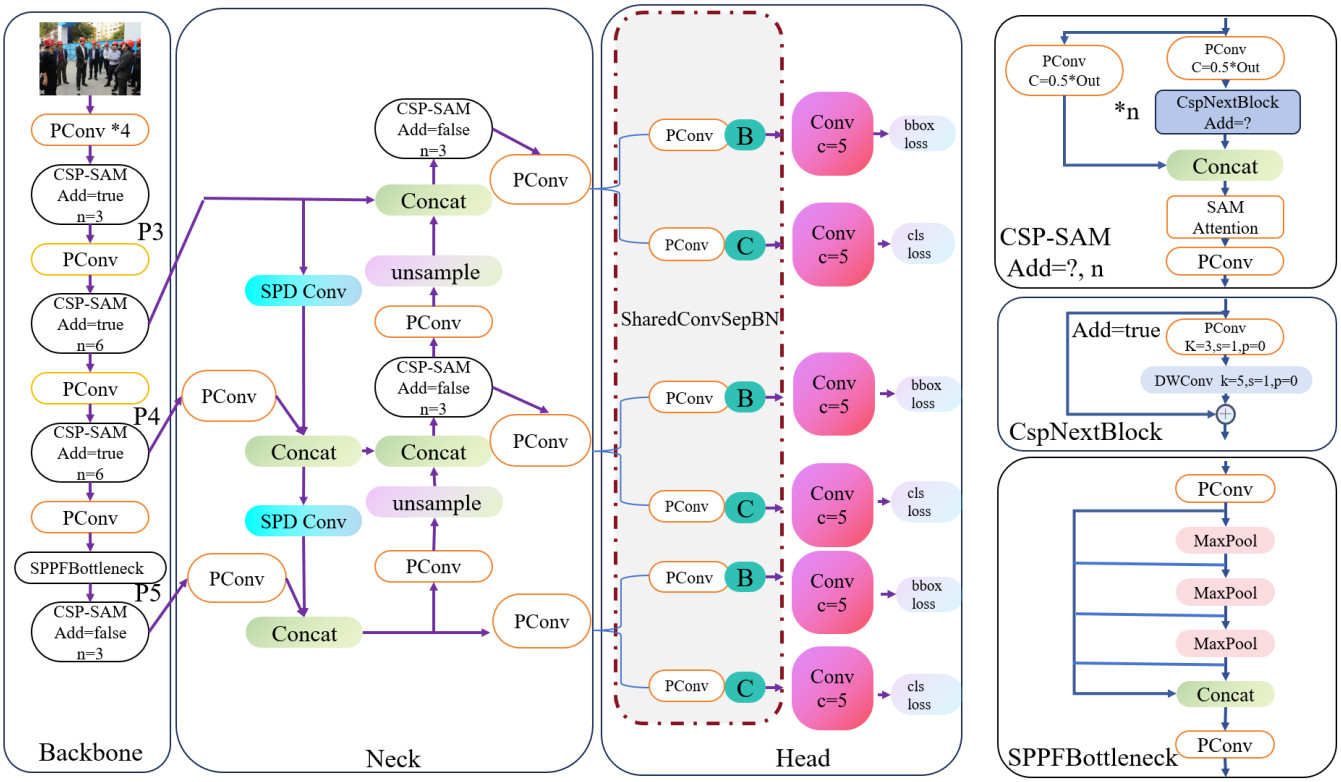


Fig. 8. Our PSP-RTMDet Network Structure Diagram

Fig. 9. Example of a Partial Image of the Dataset

range of comparison experiments with state-of-the-art detection models. Finally, subsection IV-F provides a detailed visualization and analysis of the differences between the baseline and improved models.

A. Settings

The system configuration used in this study is Linux Ubuntu 20.04. The CPU model is Intel Xeon Platinum 8255C, and the GPU model is NVIDIA GeForce RTX 3080 Ti. The system has 40 GB of memory and uses Python 3.8.0 as the programming language. The deep learning framework employed is PyTorch 1.10.0, with GPU acceleration enabled via CUDA 11.3.

B. Dataset

In this paper, we used the public dataset Safety-Helmet-Wearing Dataset (SHWD) [33] for the experiments. The SHWD dataset is designed explicitly for helmet-wearing detection. It contains a total of 7,581 images, which include helmet-wearing head objects (positive samples) and non-helmet-wearing head objects (negative samples). Positive samples are sourced from Google and Baidu and are manually labelled using LabelImg. In this context, positive samples refer to heads wearing helmets, while negative samples refer to heads without helmets.

We randomly divided the SHWD dataset into training, testing, and validation sets in a ratio of 7:1:2. The training set consists of 5,457 images, the test set includes 607 images, and the validation set contains 1,517 images. The SHWD dataset includes two categories: hat and person. Examples of images from the SHWD dataset are shown in Fig. 8.

Analysis reveals that most data in the dataset consists of small targets. The feature information of these small targets could be more comprehensive, making it easier to locate and classify them accurately. Additionally, these targets often need to be detected in small-scale images. Therefore, the target detection algorithm must be capable of effectively searching for and locating objects at this dense, small target scale.

C. Evaluation Indicators

This document uses several evaluation metrics: Precision, Recall, F1 Score, mean Average Precision (mAP), Intersection over Union (IoU), floating point operations, and the number of parameters. The evaluation is based on the number of positive samples, with an IoU threshold set at 0.5.

Specifically, mAP@0.5 represents the average precision at an IoU threshold of 0.5. In contrast, mAP@0.5:0.95 represents the average precision over IoU thresholds ranging from 0.5 to 0.95 in increments of 0.05. Floating point operations are computed based on the number of operations, while the number of parameters is counted based on the model parameters. The specific definitions of the performance metrics are as follows:

1) Precision: Measures the ratio of correctly predicted positive samples to the total predicted positive samples, i.e., the number of true positive samples divided by the total number of predicted positive samples. As shown in formula (15):

$$P = \frac{TP}{TP + FP} \tag{15}$$

2) Recall: Measures the ratio of correctly predicted positive samples to the total positive samples, i.e., the number of true positive samples divided by the total positive samples. As shown in formula (16):

$$R = \frac{TP}{TP + FN} \tag{16}$$

3) Average Precision (AP) he method for calculating AP is as follows, as shown in Equation (17):

$$AP = \int_0^1 PR\, dr \tag{17}$$

where AP is the area under the P-R curve of a certain class.

4) Mean Average Precision (mAP): Used to evaluate the target detection task, considering the precision and recall of different categories. By calculating the area under the precision-recall curve for each category, the average precision for each category is obtained, and finally, the average of all categories is taken. As shown in formula (18):

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{18}$$

D. Ablation Experiments

To demonstrate the lightweight and high efficiency of the PConv convolution, we conducted a comparison test to evaluate the performance of five types of convolutions: Conv, Depth-Conv, GSConv, PConv, and DCNV2. All tests were performed under the same experimental conditions, assessing metrics such as running time, frames per second (FPS), and the number of parameters. The results of the experiments are presented in Table I.

As shown in Table I, the total running time of the PConv convolution is 10.478 seconds, significantly lower than the running times of Conv (21.487 seconds), Depth-Conv (21.652 seconds), GSConv (23.836 seconds), and DCNV2 (97.254 seconds). The average time per operation for PConv reaches 0.00218 seconds, comparable to DCNV2 but much lower than that of the other three convolution types. Additionally, the FPS of the PConv convolution reaches 435, indicating that it processes image data more quickly than the other four convolution types. The parameter count and floating point operations (FLOPs) for PConv are 7.862K and 4.317G, respectively. This significantly reduces the computational burden associated with the more extensive computational requirements of the other convolutions. PConv demonstrates the best real-time performance and the lowest number of parameters and computations of the five convolutions tested. This suggests that the embedding based on PConv convolution is well-suited for lightweighting the backbone network of RTMDet.

TABLE I
COMPARATIVE PERFORMANCE OF MULTIPLE CONVOLUTIONS

Methods	Time/s	Average Time	FPS	FLOPs	Param(K)
Conv	21.487	0.00438	221	76.78G	146.23
Depth-Conv	21.652	0.00441	219	8.931G	16.694
GSConv	23.836	0.00485	200	38.962G	74.102
DCNV2	97.254	0.01932	49	15.776G	29.777
PConv	10.478	0.00218	435	4.317G	7.862

To validate the proposed algorithm for helmet image detection in complex scenes, we used the original RTMDet network as the baseline for ablation experiments on the SHWD dataset. The environment and parameter settings were kept unchanged. A checkmark (✓) indicates that the corresponding module was added to the model, while bold text indicates the optimal results for each column. The results are shown in Table II.

The first row of Table II displays the results of the baseline RTMDet model, which serves as a benchmark for comparison in subsequent experiments. The baseline model has 52.19M parameters, 51.02G FLOPs, and an mAP-50(%) of 89.45%. Each improvement point is evaluated independently by adding PConv, SAM-Attention, and Spd-RPAFAN. The results for RTMDet(1), RTMDet(2), RTMDet(3), RTMDet(4), RTMDet(5), and RTMDet(6) are as follows:

RTMDet(1): By optimizing the conventional convolution and introducing PConv, the model's parameters are reduced by 1.85M, FLOPs decrease by 3.23G, and the mAP-50(%) improves by 1.8%. RTMDet(2): This version optimizes the baseline architecture by replacing the original attention mechanism with SimAM. This change results in a reduction of 0.87M parameters and an improvement in the mAP-50(%) by 1.32%. RTMDet(3): We improved the feature fusion module by constructing Spd-RPAFAN. This modification reduces the parameter count by 0.3M and FLOPs by 5.52G while enhancing accuracy. RTMDet(4): By combining PConv and SimAM, this model achieves a detection accuracy of 92.15% with 3.25M fewer parameters and 3.23G fewer FLOPs, representing a 2.7% improvement over the baseline model. RTMDet(5): This model optimizes both the traditional convolution and the feature fusion module, reducing 2.83M parameters and 6.35G FLOPs while improving accuracy by over 1%. RTMDet(6): This model optimizes the baseline architecture and simultaneously improves the feature fusion module, resulting in significantly lower parameter counts and improved accuracy.

Finally, by integrating all three improvements, the mAP-50(%) is enhanced to 93.33%, with reductions of 4.87M parameters and 6.35G FLOPs. This represents an overall improvement of 3.88%, demonstrating the effectiveness and good compatibility of each enhancement.

TABLE II
ABLATION EXPERIMENTS FOR IMPROVED PROCESSES

Methods	PConv	SAM-Attention	Spd-RPAFPN	Params	FLOPs	mAP-50(%)
RTMDet				52.19M	51.02G	89.45
RTMDet(1)	✓			50.34M	47.79G	91.25
RTMDet(2)		✓		51.32M	51.02G	90.77
RTMDet(3)			✓	51.89M	45.50G	90.64
RTMDet(4)	✓	✓		48.94M	47.79G	92.15
RTMDet(5)	✓		✓	49.36M	44.67G	91.32
RTMDet(6)		✓	✓	49.81M	45.50G	91.57
PSP-RTMDet(Ours)	✓	✓	✓	47.32M	44.67G	93.33

E. Comparative Experiments with Advanced Detection Models

In this experimental section, we conducted model comparison experiments using the ResNet [34] family pre-trained on ImageNet, along with our modified CSPNeXt as alternative networks for ablation studies. The experiments involved comparing models with different backbones, including the ResNet family and our improved CSPNeXt. Additionally, in the Neck module, we performed comparative experiments on multiscale feature extraction using standard Feature Pyramid Networks (FPNs) [16] and our proposed novel Neck layer.

We ensured a fair assessment of our PSP-RTMDet for the comparison experiments against current state-of-the-art object detection models. These models include YOLOX [19], YOLOv7 [35], YOLOv8 [36], ViTDet [37], Conditional DETR [38], Sparse R-CNN [39], and RTMDet [14]. These state-of-the-art models perform well across various tasks. Our goal is to validate the advantages of our model, specifically for helmet detection.

As shown in Table III comprehensively compares the performance of our proposed PSP-RTMDet in ablation experiments. In modelling, we compared the baseline RTMDet with our proposed PSP-RTMDet. The backbones used for the investigation are CSPNeXt and Improved CSPNeXt. We perform ablation experiments to evaluate the advantages of the innovative modules and verify their superiority. We compare the experimental results by modifying the optimizer, learning rate (LR), momentum, and weights.PSP-RTMDet achieves the best detection performance in the Improved CSPNeXt configuration with each innovative module, the SGD optimizer, and LR = 0.01. Table IV, our model achieves a mAP-50(%) of 93.33% after utilizing the improved CSPNeXt backbone module, surpassing the performance of other state-of-the-art models. For instance, the YOLOv8 model achieves a mAP-50(%) of 88.55%, indicating that our model outperforms YOLOv8 by 4.78%. Additionally, our model reaches mAP(bbox), mAPs(bbox), mAP75(bbox), and Average Recall (AR) values of 93.21%, 93.29%, 93.27%, and 93.31%, respectively, all of which are significantly higher than those of other advanced models. Our model has a parameter count of only 47.32M and 44.67G FLOPs. This significantly alleviates the computational burden compared to state-of-the-art models, which typically have at least 60M parameters and over 50G FLOPs. Furthermore, in terms of real-time detection, our model achieves an impressive inference FPS of 29.8, which far exceeds YOLOX's 17.5. This demonstrates that our model has significantly stronger practical real-time performance in application scenarios.

The results highlight the excellent convergence of the PSP-RTMDet model and its robust real-time detection capabilities. From the experimental data, it is evident that PSP-RTMDet not only achieves substantial improvements in accuracy and real-time performance over the baseline model but also outperforms the latest YOLO and DETR series models in average accuracy and average recall. This improvement can primarily be attributed to the introduction of PConv, which alleviates the computational pressure associated with traditional convolution, thereby enhancing its suitability for real-time helmet detection scenarios. Additionally, by optimizing the basic building blocks in the backbone network structure and incorporating the global attention mechanism SimAM, the model's ability to extract critical feature information is improved without increasing the number of parameters. Moreover, to minimize the loss of small target feature information and enhance its representation in the shallow feature map during the fusion process, we constructed a new lightweight feature fusion module, SPD-RPAFPN. The experimental results demonstrate that PSP-RTMDet significantly reduces FLOPs while maintaining strong real-time inference performance. This balance of parallelism and testing speed without compromising accuracy underscores the robust application of our model in real-time helmet detection scenarios.

F. Visual Analysis of Detection Effects

To intuitively illustrate the differences between the improved model and the baseline RTMDet in terms of detection performance, we selected images of occluded targets and small targets viewed from a distance for comparison. The results are presented in Fig. 10 and Fig. 11. Observing the detection outcomes, it is evident that PSP-RTMDet outperforms RTMDet in detecting helmets with occlusion, as shown in Fig. 10. In detecting small, distant targets, RTMDet fails to identify them correctly, whereas PSP-RTMDet successfully detects them, as shown in Fig. 11.

In intensive target detection tasks, PSP-RTMDet and RTMDet experience varying degrees of missed detection due to image resolution. However, PSP-RTMDet has a lower error rate than RTMDet, especially in cases of shallow resolution. RTMDet exhibits a significantly higher misdetection and missed detection rate than FEVYOLOv8n, as illustrated in Fig. 12.

Comparing the detection performance of PSP-RTMDet and RTMDet across different helmet detection scenarios reveals that the improved PSP-RTMDet achieves lightweight performance while reducing the number of parameters and computation. It maintains relatively high accuracy and adapts to complex scenarios, meeting the expected detection goals.

V. CONCLUSION

To address the limitations of previous helmet detection models, such as high accuracy, which is hampered by complex network structures and high parameter counts that hinder the deployment of embedded devices, limiting real-world applicability, as well as the trade-off between extreme lightweight and reduced accuracy, we propose PSP-RTMDet. This model aims to balance high detection accuracy with real-time performance for monitoring helmet usage on construction sites. PSP-RTMDet builds on RTMDet by integrating PConv to reduce computational pressure from traditional convolutions, optimizing backbone network components, and

TABLE III
OPTIMAL PARAMETER RATIO EXPERIMENT

Method	Backbone	Optimizer	LR	Momentum	weight decay	bbox mAP	AP	bbox mAP 75	bbox mAP s	AR
RTMDet	CSPNeXt	SGD	0.001	0.8	0.001	77.45	79.13	78.55	78.57	78.01
RTMDet	CSPNeXt	SGD	0.005	0.9	0.0001	89.34	89.45	89.37	89.43	89.43
PSP-RTMDet	Improved CSPNeXt	AdamW	0.02	-	0.0001	23.13	42.15	23.38	28.39	41.49
PSP-RTMDet	Improved CSPNeXt	SGD	0.02	0.9	0.0001	67.98	92.27	73.84	70.09	74.68
PSP-RTMDet	Improved CSPNeXt	SGD	0.05	0.9	0.0001	26.49	45.87	28.79	31.72	44.46
PSP-RTMDet	Improved CSPNeXt	SGD	0.001	0.9	0.0001	33.49	56.72	33.98	41.88	53.79
PSP-RTMDet	Improved CSPNeXt	SGD	0.0001	0.9	0.0001	91.37	91.78	91.47	91.22	91.52
PSP-RTMDet	Improved CSPNeXt	SGD	0.01	0.9	0.0001	93.21	93.33	93.27	93.29	93.31

TABLE IV
COMPARATIVE EXPERIMENTS ON ADVANCED DETECTION MODELS

Method	Backbone	mAP-50(%)	mAP(bbox)	mAP_s(bbox)	mAP_75(bbox)	AR	Params	FLOPs	Inference FPS
YOLOX	ResNet-50	87.45	87.34	87.43	87.39	87.43	64.18M	82.12G	17.5
Sparse R-CNN	ResNet-50	88.21	88.11	88.21	88.15	88.19	67.62M	72.54G	19.8
YOLOV8	CSPDarknet	88.55	88.44	88.53	88.49	88.53	69.21M	81.22G	17.5
YOLOV7	ResNet-50	88.73	88.62	88.71	88.660	88.71	69.82M	52.13G	26.3
Conditional DETR	ResNet-50	89.01	88.89	88.98	88.95	88.99	99.21M	99.52G	14.9
ViTDet	ResNet-50	89.21	89.09	89.18	89.15	89.19	68.52M	68.15G	26.1
RTMDet	CSPNeXt	89.45	89.34	89.43	89.37	89.43	52.19M	51.02G	27.1
PSP-RTMDet(Ours)	Improved CSPNeXt	93.33	93.21	93.29	93.27	93.31	47.32M	44.67G	29.8

(a) RTMDet (b) PSP-RTMDet (Ours)

Fig. 10. Comparison of the Detection Effect of Occlusion Types

(a) RTMDet (b) PSP-RTMDet (Ours)

Fig. 11. Comparison of the Detection Effect of Small Target Types in the Vision

(a) RTMDet (b) PSP-RTMDet (Ours)

Fig. 12. Comparison of Dense Target Detection Effectiveness

incorporating the lightweight global attention mechanism SimAM to enhance feature extraction. It also introduces the SPD-RPAFPN fusion approach to improve handling small and dense targets, effectively integrating helmet features in crowded environments.

We evaluate PSP-RTMDet comprehensively for efficiency, real-time performance, and inference speed through extensive ablation and model comparison experiments on the SHWD dataset. The results indicate that the model achieves a detection accuracy of 93.3%, with a 9.33% reduction in parameters, a 12.44% decrease in FLOPs, and an FPS of 29.8. The mAP-50 improves by 3.88% compared to RTMDet. Additionally, PSP-RTMDet significantly outperforms mainstream advanced algorithms in detection accuracy and efficiency, demonstrating the algorithm's effectiveness. Our research contributes to high-precision, real-time helmet detection for construction site workers.

Although the model performs exceptionally well, its ability to detect helmets under extreme environmental conditions still requires improvement. Future research will focus on enhancing detection capabilities in such challenging scenarios.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[2] C. Feng, Y. Zhong, Y. Gao, M. R. Scott and W. Huang, "TOOD: Task-aligned One-stage Object Detection," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021, pp. 3490-3499, doi: 10.1109/ICCV48922.2021.00349.

[3] Liu, W., Dragomir Anguelov, D. Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu and Alexander C. Berg. "SSD: Single Shot MultiBox Detector." European Conference on Computer Vision (2015).

[4] T. -Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, "Focal Loss for Dense Object Detection," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 2999-3007, doi: 10.1109/ICCV.2017.324.

[5] M. Tan, R. Pang and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 10778-10787, doi: 10.1109/CVPR42600.2020.01079.

[6] Qian Y, Wang B. A new method for safety helmet detection based on convolutional neural network. PLoS One. 2023 Oct 13;18(10):e0292970. doi: 10.1371/journal.pone.0292970. PMID: 37831687; PMCID: PMC10575485.

[7] Li, Tianchen, Dong Li, Ming-ju Chen, Hao Wu and Yi-cen Liu. "High precision detection method of safety helmet based on convolution neural network." Chinese Journal of Liquid Crystals and Displays (2021): n. pag.

[8] Z. Yi, G. Wu, X. Pan and J. Tao, "Research on Helmet Wearing Detection in Multiple Scenarios Based on YOLOv5," 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 2021, pp. 769-773, doi: 10.1109/CCDC52312.2021.9602337.

[9] Wang Y S, Gu Y, Feng X, et al. Research on Detection Method of Helmet Wearing Based on Attitude Estimation[J]. Appl. Res. Comput, 2021, 38(3): 937-940.

[10] Luo W, Li Y, Urtasun R, et al. Understanding the Effective Receptive Field in Deep Convolutional Neural Networks[J]. Advances in Neural Information Processing Systems, 2016, 29.

[11] F. Yu, V. Koltun and T. Funkhouser, "Dilated Residual Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 636-644, doi: 10.1109/CVPR.2017.75.

[12] X. Wang, R. Girshick, A. Gupta and K. He, "Non-local Neural Networks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 7794-7803, doi: 10.1109/CVPR.2018.00813.

[13] Ren, Jiaxin, Cui, Wenhua, Tao, Ye and Shi, Tianwei, "YOLOv7-DSE: An Efficient Safety Equipment Detection Network," IAENG International Journal of Computer Science, vol. 51, no. 6, pp572-581, 2024

[14] Lyu, Chengqi, Wenwei Zhang, Haian Huang, Yue Zhou, Yudong Wang, Yanyi Liu, Shilong Zhang and Kai Chen. "RTMDet: An Empirical Study of Designing Real-Time Object Detectors." ArXiv abs/2212.07784 (2022): n. pag.

[15] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 1800-1807, doi: 10.1109/CVPR.2017.195.

[16] T. -Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature Pyramid Networks for Object Detection," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 936-944, doi: 10.1109/CVPR.2017.106.

[17] S. Liu, L. Qi, H. Qin, J. Shi and J. Jia, "Path Aggregation Network for Instance Segmentation," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 8759-8768, doi: 10.1109/CVPR.2018.00913.

[18] G. Ghiasi, T. -Y. Lin and Q. V. Le, "NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 7029-7038, doi: 10.1109/CVPR.2019.00720.

[19] Ge, Zheng, Songtao Liu, Feng Wang, Zeming Li and Jian Sun. "YOLOX: Exceeding YOLO Series in 2021." ArXiv abs/2107.08430 (2021): n. pag.

[20] X. Li, C. Lv, W. Wang, G. Li, L. Yang and J. Yang, "Generalized Focal Loss: Towards Efficient Representation Learning for Dense Object Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 3, pp. 3139-3153, 1 March 2023, doi: 10.1109/TPAMI.2022.3180392.

[21] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. 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), Long Beach, CA, USA, 2019, pp. 658-666, doi: 10.1109/CVPR.2019.00075.
[22] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo and J. Choe, "CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 6022-6031, doi: 10.1109/ICCV.2019.00612.
[23] Zhang H, Cisse M, Dauphin Y N, et al. Mixup: Beyond Empirical Risk Minimization[J]. arXiv preprint arXiv:1710.09412, 2017.
[24] J. Chen et al., "Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks," 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 2023, pp. 12021-12031, doi: 10.1109/CVPR52729.2023.01157.
[25] Yang, Lingxiao, Ru-Yuan Zhang, Lida Li and Xiaohua Xie. "SimAM: A Simple, Parameter-Free Attention Module for Convolutional Neural Networks." International Conference on Machine Learning (2021).
[26] Xuanhao Q I, Min Z H I. Review of Attention Mechanisms in Image Processing[J]. Journal of Frontiers of Computer Science & Technology, 2024, 18(2): 345.
[27] Woo, S., Park, J., Lee, JY., Kweon, I.S. (2018). CBAM: Convolutional Block Attention Module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds) Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science(), vol 11211. Springer, Cham.
[28] X. Ding, X. Zhang, J. Han and G. Ding, "Scaling Up Your Kernels to 31×31: Revisiting Large Kernel Design in CNNs," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022, pp. 11953-11965, doi: 10.1109/CVPR52688.2022.01166.
[29] S. Woo et al., "ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders," 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 2023, pp. 16133-16142, doi: 10.1109/CVPR52729.2023.01548.
[30] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 618-626, doi: 10.1109/ICCV.2017.74.
[31] Sunkara, R., Luo, T. (2023). No More Strided Convolutions or Pooling: A New CNN Building Block for Low-Resolution Images and Small Objects. In: Amini, MR., Canu, S., Fischer, A., Guns, T., Kralj Novak, P., Tsoumakas, G. (eds) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2022. Lecture Notes in Computer Science(), vol 13715. Springer, Cham.
[32] M. S. M. Sąjjadi, R. Vemulapalli and M. Brown, "Frame-Recurrent Video Super-Resolution," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 6626-6634, doi: 10.1109/CVPR.2018.00693.
[33] D. Peng, Z. Sun, Z. Chen, Z. Cai, L. Xie and L. Jin, "Detecting Heads using Feature Refine Net and Cascaded Multi-scale Architecture," 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 2018, pp. 2528-2533, doi: 10.1109/ICPR.2018.8545068.
[34] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.
[35] 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," 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 2023, pp. 7464-7475, doi: 10.1109/CVPR52729.2023.00721.
[36] Rahman, S.; Rony, J.H.; Uddin, J.; Samad, M.A. Real-Time Obstacle Detection with YOLOv8 in a WSN Using UAV Aerial Photography. J. Imaging 2023, 9, 216.
[37] Li, Yanghao, Hanzi Mao, Ross B. Girshick and Kaiming He. "Exploring Plain Vision Transformer Backbones for Object Detection." ArXiv abs/2203.16527 (2022): n. pag.
[38] D. Meng et al., "Conditional DETR for Fast Training Convergence," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021, pp. 3631-3640, doi: 10.1109/ICCV48922.2021.00363.
[39] P. Sun et al., "Sparse R-CNN: End-to-End Object Detection with Learnable Proposals," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2021, pp. 14449-14458, doi: 10.1109/CVPR46437.2021.01422.
[40] K. Han and X. Zeng, "Deep Learning-Based Workers Safety Helmet Wearing Detection on Construction Sites Using Multi-Scale Features," in IEEE Access, vol. 10, pp. 718-729, 2022, doi: 10.1109/ACCESS.2021.3138407.