2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

BS-YOLO:A Multi-scale Object Detection Model for Complex Environments

Zhaoyang Liu[1][0009-0001-0797-3337] and Guangying Jin[2](✉)

[1] Dalian Maritime University, Dalian 116026, China
liuzhaoyang04@gmail.com
[2] Dalian Maritime University, Dalian 116026, China
guangying.jin@dlmu.edu.cn

Abstract. Helmet detection is a critical component of road safety. However, existing detection algorithms face challenges in accuracy and recall, particularly when dealing with multi-scale objects in complex environments. To address these issues, this study proposes an improved YOLOv8-based model, denoted as BS-YOLO. Firstly, drawing inspiration from StarNet, we propose two modules, namely StarFuseBlock and StarSPPF, to enhance the feature extraction capability of the model at shallow layers. Secondly, to mitigate the loss of texture features of small and medium-sized objects during the feature propagation process, we propose ResBiBlock, which captures global features through residual connections and Biformer Attention. Finally, MPDIoU is employed as a superior alternative to CIoU to enhance computational efficiency and provide greater robustness in situations where predicted boxes do not align with ground-truth boxes. To validate the performance of the proposed model, a series of extensive experiments were performed on the TWHD dataset.Results show that the BS-YOLO yields a 2.2% increase in precision, a 2.8% improvement in recall, and a 2.2% enhancement in mAP compared to the YOLOv8 baseline. The experimental results indicate that the proposed improvements effectively enhance the performance of the baseline model, particularly in terms of robustness when dealing with multi-scale objects in complex scenarios.

Keywords: YOLOv8, Object Detection, Star Operation, Biformer Attention.

1 Introduction

Road safety continues to be a critical issue. Helmets, as a fundamental component of personal protective equipment, play an essential role in enhancing personal safety. By effectively absorbing impact forces during collisions, helmets reduce the risk of head injuries. However, in practice, factors such as elevated temperatures and discomfort often lead individuals, particularly cyclists, to neglect wearing helmets. This compromises their protection in the event of a collision or fall, particularly at high speeds, thereby increasing the likelihood of injury. Consequently, efficient monitoring of helmet usage has become an urgent issue that requires attention. There are two primary approaches to helmet usage detection: manual monitoring and computer vision-based detection. The manual approach is hindered by challenges such as high costs, low efficiency, and limited accuracy. Computer vision-based detection, while promising, faces several challenges: first, helmets are relatively small, and existing algorithms often struggle with detecting small objects; second, due to variations in the height and distance of monitoring equipment relative to the target, helmets appear at multiple scales in the captured images, complicating feature extraction; finally, the diversity of helmet colors and the complex nature of real-world road environments, which often involve occlusions and various types of interference, further complicate the algorithm's accuracy.

With the advancement of computer vision, the maturation of related algorithms, and improvements in hardware performance, deep learning algorithms have increasingly been applied in engineering practice across various industries. The YOLO (You Only Look Once) series are prominent one-stage object detection models, renowned for their ability to balance detection speed and accuracy. YOLOv8 utilizes the Darknet-53 backbone for feature extraction and the CIoU loss function. Additionally, its anchor-free detection head reduces computational overhead, resulting in improved detection accuracy and faster inference speed. YOLOv8 also demonstrates excellent compatibility with various hardware platforms and superior robustness in complex scenarios. As a result, YOLOv8 has demonstrated strong performance in various computer vision tasks, including segmentation, tracking, object detection, classification, and pose estimation[1].

Therefore, this paper selects YOLOv8 as the baseline for the study. However, due to the small size of helmet targets, the presence of complex interference factors in detection scenarios, and the multi-scale nature of the targets, directly applying YOLOv8 fails to meet practical requirements. To enhance detection accuracy for helmet detection in real-world scenarios, several improvements have been made to YOLOv8. The main contributions of this paper are as follows:

- Based on the efficient feature mapping to high-dimensional space provided by StarNet, the Star Operation is used to optimize the SPPF structure. Additionally, the StarFuseBlock is proposed to enhance feature extraction in shallow networks while maintaining computational efficiency. The StarModules partially alleviate the challenge of insufficient high-level semantic representation for small and medium-sized objects in shallow network layers, while also addressing feature degradation in deeper layers.
- ResBiBlock is proposed, leveraging Biformer Attention to focus on global features, while residual connections are used to fuse global and local features, enhancing the network's ability to capture relevant features.
- To simplify the computation process and improve the accuracy of bounding box regression, MPDIoU is introduced as the loss function.

2 Related Work

Object Detection. Object detection is a key application in computer vision, involving the classification and localization of visual objects. Early research on helmet detection leveraged machine learning and computer vision techniques: Rubaiyat et al. combined frequency-domain information with the HOG human detection algorithm, using CHT (color-based features and Hough Transform) for helmet detection [2]; Pang et al. conducted feature extraction using the DOD framework and applied a linear SVM for image classification, achieving higher efficiency compared to traditional frameworks[3].

The advent of two-stage algorithms marked a significant breakthrough in object detection. Algorithms such as R-CNN[4], Fast R-CNN[5], Faster R-CNN[6], R-FCN[7], and Mask R-CNN[8] raised detection accuracy to new levels. Two-stage object detection algorithms first extract Regions of Interest (ROIs) from the image, followed by classification of each ROI using a classifier, improving both classification and regression accuracy. However, the large number of overlapping proposal boxes generated during ROI extraction results in redundant feature computation, which significantly slows down detection speed. Despite improvements in detection speed with Fast R-CNN and Faster R-CNN, two-stage algorithms still incur substantial computational overhead.

To address the issues of redundant computations and slow detection speed, Joseph et al. introduced the You Only Look Once (YOLO) algorithm. As a pioneer in one-stage algorithms, YOLO performs classification and regression on anchor boxes directly after a single feature extraction, significantly improving detection speed at the cost of some accuracy. Concurrently, Liu et al. [9]proposed the Single-Shot Multi-box Detector (SSD), another classic one-stage algorithm. SSD incorporates multi-reference and multi-scale detection techniques, enhancing both speed and accuracy, particularly for small object detection. Over the years, the YOLO series has made substantial advancements in computer vision, offering new research directions.

Attention Mechanism. In computer vision, the attention mechanism is a dynamic selection process that adaptively assigns weights to input features, enabling the model to focus on the most critical regions of the image. Early work by Mnih et al. introduced RAM[10], which used policy gradients to predict important regions and update the network end-to-end. Jaderberg et al. proposed the spatial attention model STN[11], which used affine transformations to select important input regions. Hu et al. introduced SE-Net[12], a classic channel attention model that computes feature map channel weights via fully connected layers. Woo et al. proposed CBAM[13], which considers both spatial and channel information, enhancing feature extraction without increasing network complexity. The Transformer[14] architecture further advanced attention mechanisms by introducing self-attention, allowing the model to capture long-range dependencies across all elements of the sequence simultaneously. Subsequent models, such as ViT[15] and Swin Transformer[16], further extended attention mechanisms, advancing their application in vision tasks.

3 Methods

Overall Structure. This section will elaborate on the details of the proposed improvements. Our improvements are primarily based on the YOLOv8 network structure.

The modified model structure is shown in **Fig. 1**. To enhance target feature capture, we integrate StarBlock into the SPPF module, forming StarNet, which improves the model's ability to extract complex features, particularly for helmets in complex scenarios. All C2f modules in the feature extraction network are modified by incorporating Biformer attention, enabling the model to focus on relevant regions and improving small object feature extraction while reducing background interference. Lastly, IoU is calculated based on the minimum point distance between bounding boxes, simplifying computation while accounting for relevant factors from existing loss functions.

Star Modules. SPPF is an efficient pooling structure inspired by spatial pyramid pooling, which captures multi-scale features by applying pooling at different scales. In SPPF, the input feature map is pooled three times with varying kernel sizes and then concatenated to retain both local and global information.

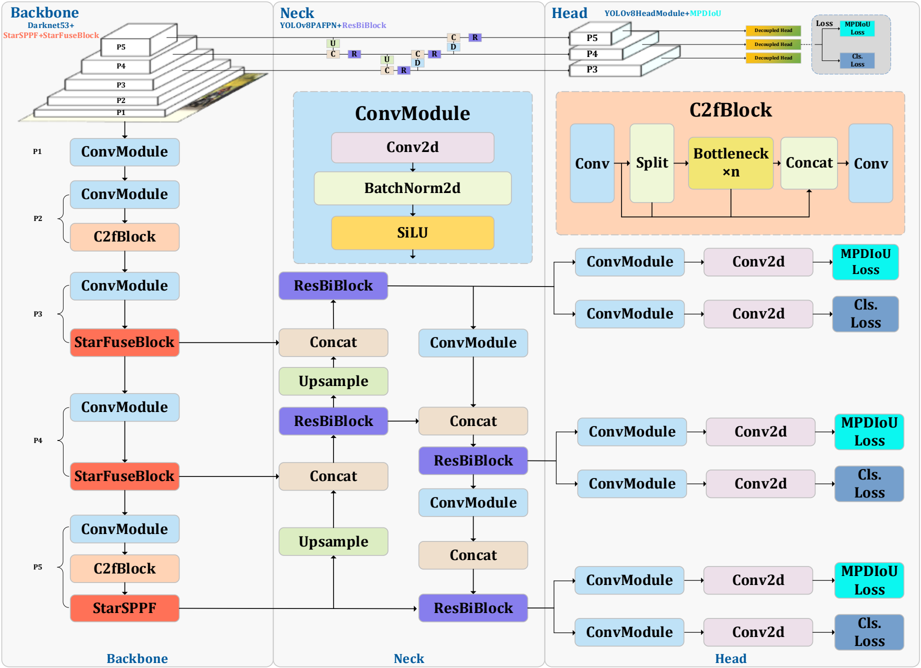


Fig. 1. Overall Structure of BS-YOLO.

However, constrained by image resolution, the features of smaller objects are typically retained only in the shallow layers of the network. Although repeated convolutions can capture their edge details, this process often results in the loss of high-level semantic features. To enhance the model's ability to map inputs to high-dimensional non-linear features without adding extra computational burden, we introduce StarNet[17]. StarNet is a simple and efficient feature extraction network that leverages the Star Operation to project input features into a high-dimensional nonlinear space, thereby enhancing the network's overall feature extraction capacity. Specifically, for small objects, the Star Operation(element-wise multiplication) enables the acquisition of high-level semantic information prior to feature degradation, and facilitates its transmission to deeper network layers. For one output channel, a star operation can be written as $(\boldsymbol{w}_1^T\boldsymbol{x}) * (\boldsymbol{w}_2^T\boldsymbol{x})$, where $\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{x} \in R^{(d+\mathbb{1})\times\mathbb{1}}$. Through the statistical analysis of the quadratic and cross terms in the star operation, we obtain $(d+1)+\frac{(d+1)\times d}{2}$ distinct combinations, which represent $\frac{(d+1)(d+2)}{2}$ implicit feature dimensions in the high-dimensional space. When multiple layers of the Star Operation are stacked in a neural network, the implicit high-dimensional feature space increases exponentially.

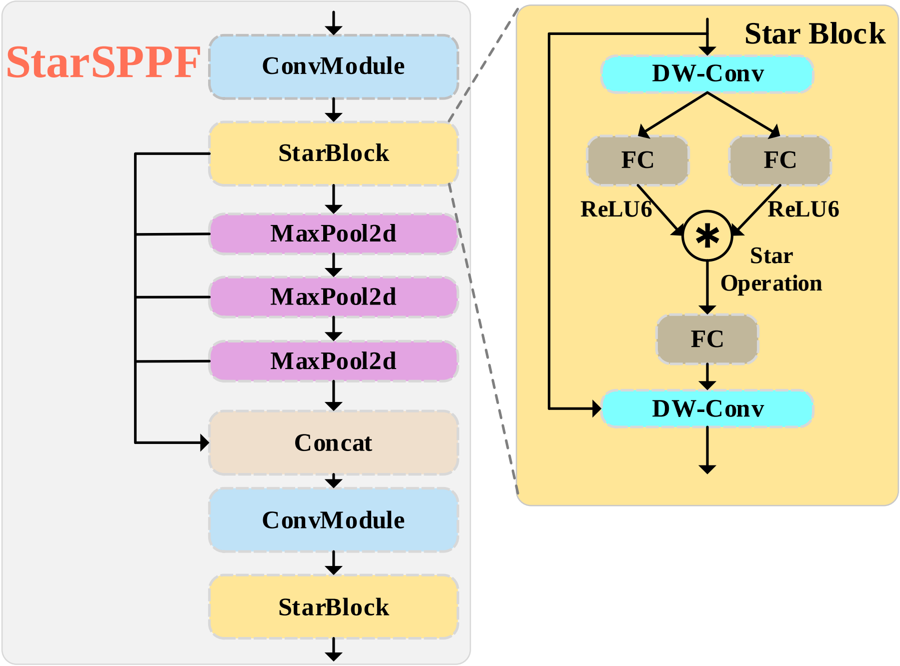


Fig. 2. Details of StarSPPF. In StarNet, the feature maps activated through fully connected layers are fused via star operation(element-wise multiplication).

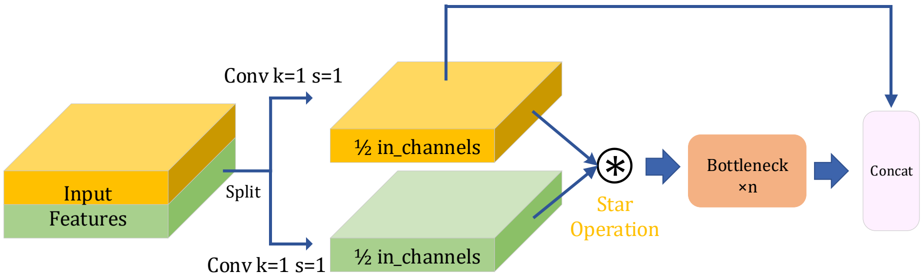


Fig. 3. Feature Transmission Path in StarFuseBlock

Therefore, even in simple network structures, the Star Operation can lead to significant performance improvements.

StarNet is formed by stacking multiple StarBlocks. In the SPPF, we insert StarBlocks at the head and tail of the structure to create a compact StarNet architecture, as shown in **Fig. 2**, StarBlock replaces the traditional ConvModule (Conv2d, BatchNorm,SiLU) convolution module with DWConv (Depth-Wise Convolution). DWConv significantly reduces both the parameter count and computational overhead through the use of grouped convolution. Subsequently, the feature maps are passed through two 1×1 convolutional layers, denoted as FC1 and FC2. Next, the feature map processed by FC1 is activated using the ReLU6 function and combined with the output of FC2 through the star operation. Finally, the initial input is fused with the result of the star operation via a residual connection.

In StarFuseBlock, the input feature channels are equally split into two parts, as shown in **Fig. 3**. These two parts are processed via the Star Operation and then passed through n bottleneck layers for further transformation. Meanwhile, the other half of the original input is retained and merged with the processed features via a residual connection, followed by a convolutional layer for feature fusion.

ResBiBlock. The attention mechanism is the core of the Transformer architecture[14], enabling the model to focus more on the key features in the input. In the classic Transformer, self-attention is used, which involves applying a linear transformation to the same input and then using it to compute the attention matrix, expressed as:

$$Attention(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{T}}{\sqrt{C}}\right)\boldsymbol{V} \tag{1}$$

The queries **Q**, keys **K**, and values **V** are obtained by applying linear transformations to the same input **X**. The $\sqrt{\mathrm{C}}$ is introduced to mitigate the vanishing gradient problem. However, due to the need to compute attention weights between each pixel (query) and all other pixels (keys), the self-attention mechanism incurs high memory usage and computational cost. To address this, various improvements to Transformer attention, such as local window, axial stripe, dilated window, and deformable attention, have been proposed to compute sparse attention, thereby reducing both computational and storage overhead.

Biformer[18] achieves dynamic and query-adaptive sparse attention computation through the Bi-Level Routing Attention (BRA) mechanism. The key idea is to filter out the majority of irrelevant query-key pairs in the input image, ensuring that each query attends only to a limited number of key-value pairs.

As shown in **Fig. 4**, the input **X** is reshaped into S^2equally sized regions. For the reshaped input $\mathbf{X}^{\mathrm{r}}$, linear transformations are applied using the matrices $\mathbf{W}^{\mathrm{q}}$,$\mathbf{W}^{\mathrm{k}}$,$\mathbf{W}^{\mathrm{v}}$, generating the initial **Q**,**K**,**V** matrices:

$$\boldsymbol{Q} = \boldsymbol{X}^{r}\boldsymbol{W}^{q}, \boldsymbol{K} = \boldsymbol{X}^{r}\boldsymbol{W}^{k}, \boldsymbol{V} = \boldsymbol{X}^{r}\boldsymbol{W}^{r} \tag{2}$$

Subsequently, the region-wise averages of **Q** and **K** are computed, and the correlation adjacency matrix $\mathbf{A}^{\mathrm{r}}$ is obtained through matrix multiplication:

$$\boldsymbol{A}^{r} = \boldsymbol{Q}^{r}(\boldsymbol{K}^{r})^{T} \tag{3}$$

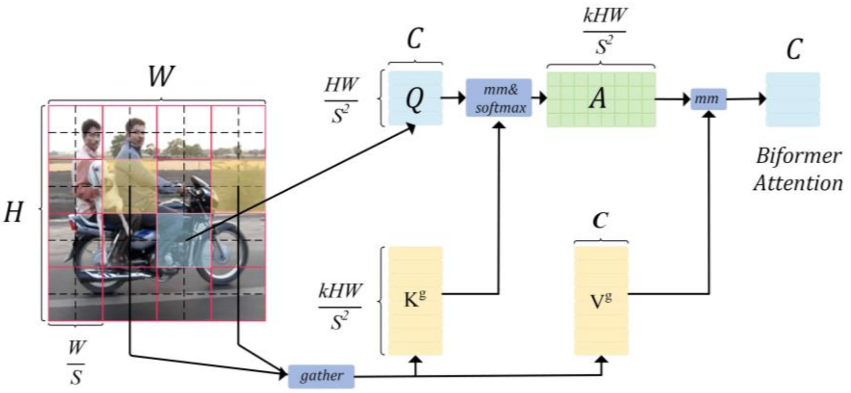


Fig. 4. Biformer Attention

Based on the correlation between Q and K, the top-k regions are selected row-wise from $\mathbf{A}^{r}$, denoted as $\mathbf{I}^{r}$, which represent the regions that **Q** needs to focus on the most. The specific operation is as follows:

$$\boldsymbol{I}^{r} = topkIndex(\boldsymbol{A}^{r}) \tag{4}$$

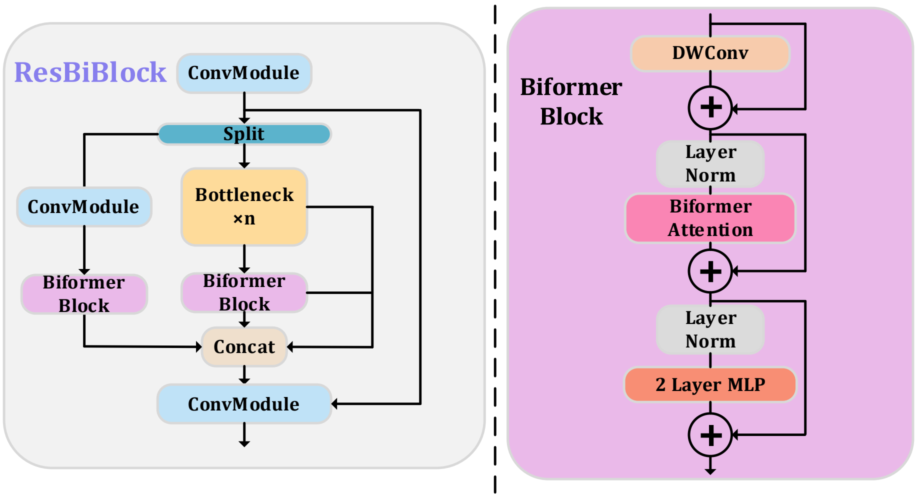


Fig. 5. Details of ResBiBlock.

Subsequently, based on $\mathbf{I}^{r}$, the corresponding keys and values from **K** and **V** within the selected regions are extracted to obtain $\mathbf{K}^{g}$ and $\mathbf{V}^{g}$.Finally, by inputting **Q**, $\mathbf{K}^{g}$ and $\mathbf{V}^{g}$ into (1), the Biformer Attention is computed, as described in (5) and (6).

$$\boldsymbol{K}^{g} = gather(\boldsymbol{K}, \boldsymbol{I}^{r}), \boldsymbol{V}^{g} = gather(\boldsymbol{V}, \boldsymbol{I}^{r}) \tag{5}$$

$$Biformer\ Attention = Attention(\boldsymbol{Q}, \boldsymbol{K}^{g}, \boldsymbol{V}^{g}) \tag{6}$$

As shown in **Fig. 5**, Biformer Blocks encode relative positional information via a 3×3 Depth-Wise Convolution and apply the BRA module to compute attention, which is then fused with features through a two-layer MLP. Each submodule fuses the previous and current outputs to preserve multi-scale information. ResBiBlock partitions the input into two branches. One branch undergoes a convolutional operation before entering the Biformer Block to capture global context from shallow features, while the other branch is processed through multiple Bottleneck layers to extract high-dimensional semantic features. Finally, the two branches are merged with the original input through residual connections and fused via a convolution operation. The long-range dependencies inherent in the self-attention mechanism allow the network to effectively incorporate shallow-layer information, thereby ensuring a richer representation of semantic information.

MPDIoU. BBR(Bounding Box Regression) is a crucial component in object detection, and a well-designed loss function can improve the convergence speed and accuracy of BBR. However, existing loss functions do not fully utilize the geometric properties of bounding box regression. Therefore, we introduce the Minimum Point Distance-Based IoU loss[19].

MPDIoU Loss uses the minimum point distance to describe the similarity between the predicted and ground truth boxes. It incorporates all the relevant factors considered by existing loss functions, such as the overlap or non-overlap area, the center point distance, and the deviations in width and height, while simplifying the computation.**Fig. 6** illustrates the parameters of MPDIoU, where w and h represent the width and height of the input image, respectively.

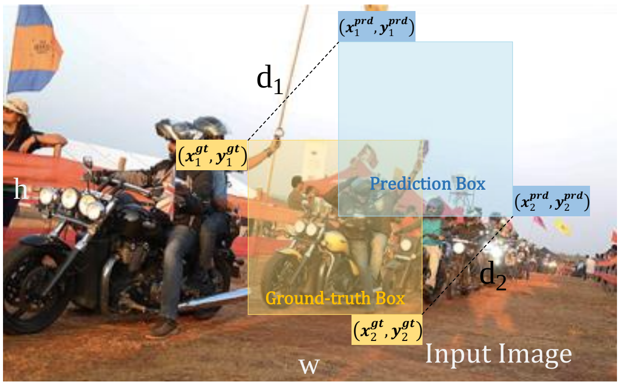


Fig. 6. MPDIoU

(x_1^A, y_1^A), (x_2^A, y_2^A) represent the coordinates of the top-left and bottom-right corners of box A, and (x_1^B, y_1^B),(x_2^B, y_2^B) correspond to those of box B. d_1^2 and d_2^2 represent the Euclidean distances between A and B's top-left and bottom-right corners, respectively. Their calculations are as follows:

$$d_1^2 = (x_1^B - x_1^A)^2 + (y_1^B - y_1^A)^2 \tag{7}$$

$$d_2^2 = (x_2^B - x_2^A)^2 + (y_2^B - y_2^A)^2 \tag{8}$$

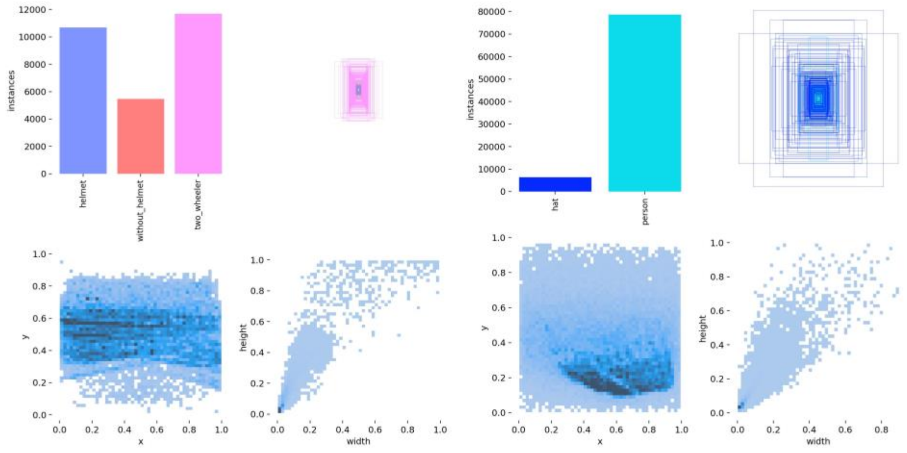
Based on the above definitions, the MPDIoU and MPDIoU Loss are defined as in (9) and (10):

$$MPDIoU = \frac{A \cap B}{A \cup B} - \frac{d_1^2}{w^2 + h^2} - \frac{d_2^2}{w^2 + h^2} \tag{9}$$

$$L_{MPDIoU} = 1 - MPDIoU \tag{10}$$

4 Experiments

4.1 Datasets and Evaluation Metrics



(a) **(b)**

Fig. 7. Data distribution of TWHD (a) and SHWD (b), showing the number of objects per category (top-left), ground truth box sizes (top-right), center locations (bottom-left), and width-height distribution (bottom-right).

To verify the effectiveness of BS-YOLO, experiments are conducted on the TWHD and SHWD datasets. The TWHD dataset is a high-quality image collection for helmet detection in two-wheeled vehicle riders, containing 5,448 images. It includes 4,710 re-annotated images from the OSF dataset and 738 images from the Bike Helmet dataset to enhance background diversity. Annotations follow the Pascal VOC format with three categories: two-wheeler, helmet, and without_helmet. The dataset covers diverse conditions—weather, lighting, environment, and traffic—across various scenes with differences in brightness, occlusion, density, camera angles, rider poses, and helmet appearances. The SHWD dataset is designed for safety helmet and human head detection, containing 7,581 images with 9,044 helmet-wearing (positive) and 111,514 non-helmet (negative) head instances. It is widely used for evaluating helmet detection performance due to its large scale and clear annotations. The statistical distributions of the TWHD and SHWD datasets are presented in **Fig. 7**.

In object detection tasks, model performance is primarily evaluated using metrics such as P (Precision), R (Recall), and Mean Average Precision (mAP). Precision (P) is the proportion of correctly predicted positive instances out of the total positive instances, and is calculated as follows:

$$P = \frac{TP}{TP+FP} \tag{11}$$

where TP represents the number of True Positive samples, and FN represents the number of False Negative samples. Recall (R) is the proportion of correctly predicted positive samples out of all the actual positive samples, and its formula is given as follows:

$$R = \frac{TP}{TP+FN} \tag{12}$$

where FP represents the number of samples predicted as False Positive. Average Precision (AP) is calculated as the area under the Precision-Recall curve for each class, and mAP is used to evaluate the average precision across multiple classes, and is calculated as follows:

$$AP = \int_0^1 P(R)\ dR \tag{13}$$

$$mAP_{50} = \frac{\sum_{i=1}^{C} AP_i}{C} \tag{14}$$

where C represents the total number of categories in the labels. mAP@50 refers to the mAP calculated with an IoU threshold of 50, meaning a prediction is considered a valid detection only when the Intersection over Union (IoU) between the predicted box and the ground truth box is greater than or equal to 50. And mAP@50:95 represents the mean average precision across multiple IoU thresholds, ranging from 0.5 to 0.95. It provides a comprehensive evaluation of the model's performance under varying levels of matching strictness.

4.2 Experiment Settings

The experiment was conducted on an Ubuntu 22.04 operating system. The hardware configuration includes a 12-core CPU, 12GB RAM, and an RTX 3080Ti GPU with 12GB of VRAM. The deep learning framework used is PyTorch version 2.1.0, with CUDA version 12.1. During the YOLOv8 training process, the input image size was set to 640×640, with a maximum of 300 epochs and a batch size of 16 images. No pretrained weights were used, and the training utilized the SGD optimizer with a learning rate of 0.01.

4.3 Ablation Experiments

To evaluate the effectiveness of each proposed improvement in this paper, an ablation study was conducted on the baseline using the TWHD dataset. **Table 1.** provides a detailed description of the performance of different improvements and their combinations. The experimental results indicate that each innovation contributes to performance improvement to varying extents. When combined, these innovations lead to even more significant enhancements. The use of Star Modules (StarSPPF, StarFuseBlock) enables the network to map shallow features to a high-dimensional feature space, resulting in a 1.1% increase in mAP.

The introduction of the ResBiBlock module enhances the model's attention to important features through Biformer Attention, leading to a 0.6% improvement in mAP. The incorporation of MPDIoU fully leverages the geometric properties of bounding box regression, effectively handling both overlapping and non-overlapping cases between predicted and ground truth boxes, while simplifying the computation, which increases mAP by 0.1%. Furthermore, combinations of these improvements lead to further performance gains. In particular, when StarSPPF and ResBiBlock are combined, the multi-scale high-dimensional features generated by StarSPPF, together with the excellent long-range attention from Biformer, allow the model to focus more on the crucial parts of the high-dimensional features, thus strengthening its feature extraction capability in complex scenarios. This combination results in a 1.8% improvement in mAP, which also demonstrates the feasibility of the proposed improvements. When these three improvements are integrated, ResBiBlock establishes long-range attention on the high-dimensional features generated within StarSPPF, while MPDIoU directs the model to mitigate biases in feature extraction, ultimately resulting in a 2.2% increase in mAP.

Table 1. Results of The Ablation Experiments

YOLOv8	Star Modules	ResBiBlock	MPDIoU	P	R	mAP50
√				0.841	0.769	0.829
√	√			0.843	0.792	0.840
√		√		0.845	0.773	0.835
√			√	0.842	0.770	0.830
√	√	√		0.856	0.797	0.847
√	√		√	0.847	0.776	0.842
√		√	√	0.841	0.769	0.837
√	√	√	√	**0.863**	**0.797**	**0.851**

4.4 Comparison Experiments of Attention Mechanism

To evaluate the impact of combining StarSPPF with ResBiBlock, experiments were conducted by substituting the Biformer Attention with alternative attention mechanisms at the same position,the results are presented in **Table 2.**

When the attention mechanism is replaced from Biformer Attention to CBAM, SENet, or SaE, although the model performance improves relative to the baseline, it consistently underperforms compared to ResBiBlock. This can be attributed to the self-attention mechanism in Biformer Attention, which effectively captures long-range dependencies between features, whereas the aforementioned attention mechanisms primarily focus on the importance of channel-wise or spatial information. However, the SE and SaE attention mechanisms focus solely on important features along the channel dimension. Although CBAM combines both channel and spatial attention, it still struggles to capture long-range dependencies between distant features. Therefore, ResBiBlock captures a greater number of high-dimensional features generated by StarSPPF and StarFuseBlock compared to CBAM, SENet, and SaE, resulting in better precision, recall, and mAP.

Table 2. Comparison Experiments of Attention Mechanism

Attentions	P	R	mAP
StarModuels+CBAM[20]+MPDIoU	0.844	0.769	0.827
StarModules+SENet[12]+MPDIoU	0.824	0.772	0.819
StarModules+SaE[21] +MPDIoU	0.843	0.771	0.833
StarModules+ResBiBlock+MPDIoU	**0.863**	**0.797**	**0.851**

4.5 Comparison Experiments

To demonstrate the advantages and effectiveness of the proposed improvements to the YOLOv8 model, we conducted multiple comparative experiments on the same dataset, comparing the proposed model with other well-known models. In terms of model selection, we first chose the classic One-Stage algorithm SSD[9] and the Two-Stage algorithm Faster R-CNN[6]. Additionally, we compared various models within the YOLO series, including YOLOv5, YOLOv8-v12, as well as the transformer based RT-DETR[22] and recently published D-FINE[23].An overview of the experimental data is provided in **Table 3.**

Table 3. Results of The Comparison Experiments on TWHD

Models	P	R	mAP	Params(M)	GFLOPs
SSD	0.717	0.222	0.321	26.3	62.8
Faster R-CNN	0.273	0.409	0.337	137.1	370.2
YOLOv5	0.823	0.781	0.827	9.1	23.8
YOLOv8	0.841	0.779	0.834	11.1	28.4
YOLOv9	0.818	0.789	0.828	**7.2**	26.7
YOLOv10	0.834	0.773	0.826	8.0	24.5
YOLOv11	0.838	0.777	0.825	9.4	21.3
YOLOv12	0.820	0.776	0.821	9.2	**21.2**
RT-DETR	0.769	0.742	0.785	32.0	103.4
D-FINE	0.797	0.762	0.813	10.0	25.0
BS-YOLO	**0.863**	**0.797**	**0.851**	14.2	31.1

SSD is an efficient and lightweight object detection model that uses the classic VGG16 as its backbone network, achieving good accuracy and speed under typical scenarios. However, in complex scenes, SSD's feature extraction capability faces challenges, resulting in a decrease in both precision and recall. As a Two-Stage anchor-based algorithm, Faster R-CNN has slower inference speed and lower precision but performs better in recall.YOLOv5 further improves model adaptability and detection accuracy in complex scenes by using Mosaic data augmentation and CIoU loss function, leading to significant improvements over traditional algorithms.YOLOv8, a high-performance and widely used algorithm in the YOLO series, employs SPPF for multi-scale feature pooling and fusion, and replaces C3 with C2f for enhanced feature extraction.

The model proposed in this paper also uses YOLOv8 as the baseline, but demonstrates significantly superior performance compared to other models on the same dataset. The proposed model enhances feature extraction in complex scenarios by generating additional high-dimensional space with StarModules(StarSPPF and StarFuse-Block), while incorporating Biformer Attention in the Neck to improve the model's focus on important features. Both modifications enhance the model's ability to extract features of multi-scale objects in complex scenarios. However, when combined, the long-range attention mechanism of Biformer Attention and the high-dimensional features extracted by StarModules synergistically improve detection performance. The introduction of MPDIoU further enhances the model's robustness in complex scenarios, leading to improvements in both precision and recall. The performance gap between YOLOv5 and YOLOv8–v12 on the TWHD dataset is within 1.3%. YOLOv9, in particular, incorporates the PGI module to mitigate gradient loss and adopts the lightweight GELAN architecture. YOLOv10–v12 preserve lightweight characteristics while balancing speed and accuracy. The Transformer-based RT-DETR requires substantial computational resources and a large number of parameters, yet its ability to extract features in complex environments is limited, resulting in the poor effect. Although D-FINE is relatively lightweight, its actual performance still falls short compared to the YOLO series.

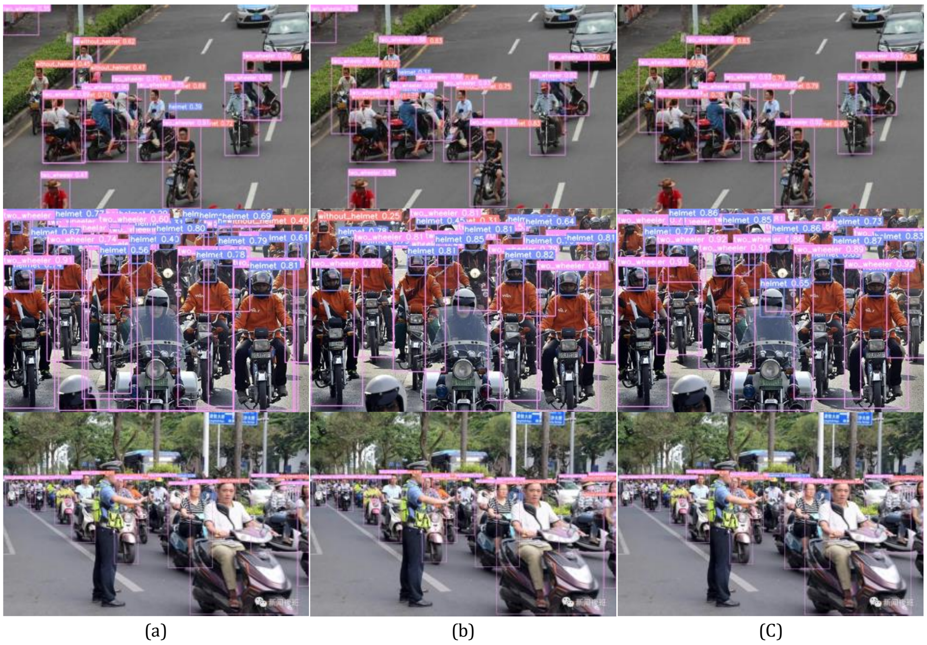
To further evaluate the performance of BS-YOLO, we conducted comparative experiments on the SHWD dataset against several well-performing YOLO-series models and Transformer-based models. The results of the experiments are presented in **Table 4.** Experimental results indicate that BS-YOLO outperforms algorithms such as YOLOv8 and D-FINE on the SHWD dataset. Among the evaluated models, BS-YOLO achieves the highest performance in terms of precision, mAP@50, and mAP@50:95, demonstrating its strong effectiveness on the SHWD dataset and highlighting its superior generalization capability.

Table 4. Comparison Experiments on SHWD

Models	P	R	mAP50	mAP50:95
YOLOv8	0.919	**0.832**	0.889	0.563
YOLOv12	0.908	0.835	0.889	0.565
D-FINE	0.893	0.791	0.856	0.512
BS-YOLO	**0.922**	0.831	**0.894**	**0.568**

4.6 Visual Results on the TWHD dataset

BS-YOLO achieves a precision of 86.3%, recall of 79.7%, and mAP of 85.1% on the TWHD dataset. **Fig. 8** presents the visualization results of the algorithm on the TWHD dataset, demonstrating its superior helmet detection capability. In the visualization results, a comparison is made between the performance of the best-performing YOLO model, YOLOv8, the recently released lightweight detector D-FINE, and BS-YOLO. In the images, D-FINE and YOLOv8 exhibit varying degrees of missed and false detections, with D-FINE showing more severe false detections compared to YOLOv8. In contrast, BS-YOLO does not produce any false detections and demonstrates higher confidence in the correct detections. It can be noted that, when confronted with complex environments (e.g., occlusion, lighting variations), BS-YOLO demonstrates strong performance and robustness in detecting multi-scale objects.



(a) (b) (C)

Fig. 8. Visual Results of D-FINE(a), YOLOv8(b), BS-YOLO(c) on TWHD

5 Conclusion

The helmet detection task presents various challenges, including but not limited to complex scenes and multi-scale targets. To address these challenges, this paper designs an innovative model based on YOLOv8 for helmet-wearing detection in complex traffic environments. In BS-YOLO, the SPPF structure is optimized based on the StarNet concept, resulting in the proposed StarSPPF. Additionally, a feature fusion module, StarFuseBlock, is designed. Both modules adopt the Star Operation as the core mechanism, enabling the mapping of low-dimensional features extracted from shallow layers to a higher-dimensional feature space. This effectively mitigates the insufficient high-level semantic representation of small and medium objects and alleviates the issue of feature loss in deeper layers of the network. In ResBiBlock, the input undergoes multi-path feature extraction with the integration of Biformer Attention, and is subsequently fused with the original input via residual connections. This design effectively enhances the network's ability to focus on globally significant features. Moreover, the long-range modeling capability of self-attention enables the deep layers to retain awareness of shallow-level features. Finally, the MPDIoU loss function is introduced to improve the model's robustness in complex environments. Despite these improvements, the proposed model still has some limitations. Although the performance improvement of BS-YOLO is notable, it comes at the cost of increased parameters and computational complexity, with Biformer Attention contributing significantly to the overhead. Future work may explore approaches such as model pruning or knowledge distillation to reduce model size while maintaining performance. Alternatively, more efficient mechanisms that retain the benefits of self-attention with substantially lower computational cost could be investigated.

References

1. Vijayakumar, A., Vairavasundaram, S.: YOLO-based Object Detection Models: A Review and its Applications. Multimed Tools Appl 83, 83535–83574 (2024).
2. Rubaiyat, A.H.M., et al.: Automatic detection of helmet uses for construction safety. In: Proc. 2016 IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW), pp. 135-142 (2016).
3. Pang, K., Zhang, Y., Yuan, K., Wang, K.: Distributed Object Detection With Linear SVMs. IEEE Transactions on Cybernetics 44(11), 2122–2133 (2014).
4. R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2014, pp. 580–587.
5. R. Girshick, "Fast R-CNN," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2015, pp. 1440–1448.
6. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2015, pp. 91–99.
7. Dai, Z., et al.: R-FCN: Object detection via region-based fully convolutional networks. In: Proc. Advances in Neural Information Processing Systems (NeurIPS), pp. 379-387 (2016).
8. He, K., et al.: Mask R-CNN. In: Proc. IEEE International Conference on Computer Vision (ICCV), pp. 2961-2969 (2017).
9. Liu, W., et al.: SSD: Single Shot MultiBox Detector. In: European Conference on Computer Vision (ECCV), pp. 21-37 (2016).
10. Mnih, V., et al.: Recurrent models of visual attention. In: Proc. 27th International Conference on Neural Information Processing Systems, vol. 2, pp. 2204-2212 (2014).
11. Jaderberg, M., et al.: Spatial transformer networks. In: Proc. 28th International Conference on Neural Information Processing Systems, vol. 2, pp. 2025 (2015).
12. Hu, J., et al.: Squeeze-and-excitation networks. IEEE Trans. Pattern Anal. Mach. Intell. 42(8), 2011-2023 (2020).

13. Woo, S., Park, J., Lee, J., Kweon, I.S.: CBAM: Convolutional block attention module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision-ECCV 2018, LNCS, vol. 11211, pp. 3–19. Springer, Cham (2018).
14. Vaswani, A., et al.: Attention is all you need. In: Proc. 31st International Conference on Neural Information Processing Systems, pp. 5998-6008 (2017).
15. Dosovitskiy, A., et al.: An image is worth 16×16 words: Transformers for image recognition at scale. In: Proc. 9th International Conference on Learning Representations (2021).
16. Liu, Z., et al.: Swin Transformer: Hierarchical vision transformer using shifted windows. In: Proc. IEEE/CVF International Conference on Computer Vision, pp. 10012-10022 (2021).
17. Ma, X., Dai, X., Bai, Y., Wang, Y., Fu, Y.: Rewrite the Stars. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5694-5703 (2024).
18. Zhu, L., Wang, X., Ke, Z., Zhang, W., Lau, R.W.: Biformer: Vision transformer with bi-level routing attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10323–10333 (2023).
19. Ma, S., Xu, Y.: MPDIoU: A loss for efficient and accurate bounding box regression. arXiv preprint arXiv:2307.07662 (2023).
20. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: CBAM: Convolutional block attention module. In: Proc. European Conference on Computer Vision (ECCV), pp. 3–19 (2018)
21. Narayanan, M.: SENetV2: Aggregated dense layer for channelwise and global representations. arXiv:2311.10807 (2023)
22. Zhao, Y., Lv, W., Xu, S., et al.: DETRs Beat YOLOs on Real-Time Object Detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 16965–16974 (2024)
23. Peng, Y., Li, H., Wu, P., et al.: D-FINE: Redefine Regression Task in DETRs as Fine-grained Distribution Refinement. arXiv preprint arXiv:2410.13842 (2024)