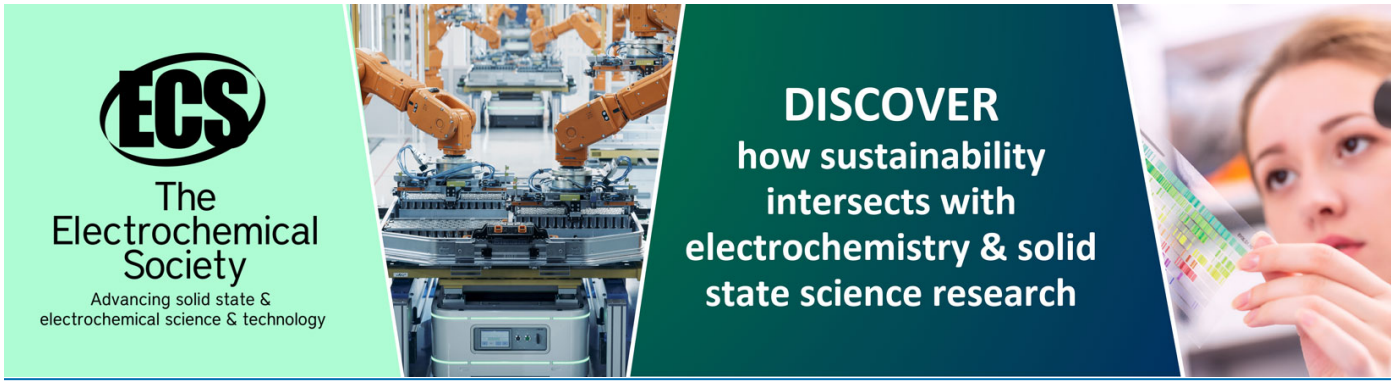
Improved YOLOV8 Network and Application in Safety Helmet Detection



Improved YOLOV8 Network and Application in Safety Helmet Detection

Junjie Bao[a*], Shihua Li[b], Guanglong Wang[c], Jianmin Xiong[d], and Sailai Li[e]

CHN ENERGY CHANGYUAN JINGMEN POWER GENERATION CO., LTD.

E-mail: [a] 12002402@ceic.com; [b] 851784592@qq.com; [c] 12041647@ceic.com;

[d] 12058971@ceic.com; [e] 12083846@ceic.com

Abstract: This paper proposes a research method to enhance the accuracy and real-time capability of helmet detection in complex industrial environments, aiming to address the engineering challenges of poor robustness and significant occurrences of both false positives and false negatives in existing detection methods. In this study, the C2F (faster version of CSP Bottleneck with two convolutions) module and FE (FasterNet with EMA) module are integrated into the network architecture of YOLOV8 to form a new attention mechanism module called C2F-FE. This module enhances the model's perception of safety helmet targets by fusing feature information from different levels and incorporating attention mechanisms while reducing computational overhead. Furthermore, the model is trained and optimized on publicly available safety helmet datasets. Experimental results demonstrate that the improved model exhibits stronger robustness, achieving an accuracy rate of 94.6% and a mAP50 of 99.1% for safety helmet detection in complex construction scenarios, with an inference time of 0.7 ms.

1. Introduction

The benefits of artificial intelligence to human society are increasingly prominent [1]. Safety helmets effectively reduce the risk of head injuries. However, due to workers' negligence, improper wearing of safety helmets always happens. Therefore, developing a system that can be widely deployed to detect in real-time and alert the wearing condition of safety helmets for workers becomes crucial. In the past few years, deep learning techniques have significantly progressed in computer vision. The YOLO (You Only Look Once) series of single-stage object detection models have shown great potential in safety helmet detection in industrial settings, compared to multi-stage methods such as RCNN, Fast-RCNN, and RPN, as the YOLO series offers faster detection speed. YOLOV8 [2], as the SOTA model in the YOLO series, inherits the advantages of real-time performance, multi-scale feature fusion, and simplicity from previous versions. It further improves accuracy, inference speed, and scalability. However, it has relatively low FLOPS efficiency, redundant computations, and memory access and requires improvement in balancing model size and accuracy. This study aims to integrate the EMA attention mechanism [3] and PConv [4] into YOLOV8 to develop an object detection technology that can detect and alert the wearing condition of safety helmets in real time so that it can improve workers' safety in the industrial production process. The primary contributions of this research can be summarized as follows:

1) The C2F-FE module is used for the YOLOV8 framework, improving the accuracy, real-time performance, and generalization ability of safety helmet detection;

2) Training and evaluating the YOLOV8 model before and after the improvements and comparing their performance are necessary;

3) An efficient and accurate safety helmet detection system is developed by adopting the improved YOLOV8 object detection model.

2. Method

2.1. The architecture of the modified YOLOV8

In this section, we optimize the architecture of the YOLOV8 object detection network to achieve better performance in safety helmet detection. As shown in Figure 1, the modified YOLOV8 network architecture includes the backbone, neck, and head components. The backbone network generates three different scale feature maps, which are then fed into the neck for feature fusion. The fusion process involves upsampling and concatenation operations, enabling a more comprehensive and enriched feature representation. This helps improve the model's detection, classification, and segmentation capabilities in the prediction head. By leveraging the YOLOV8 architecture, we aim to accurately improve the model's detection capability on safety helmets in various industrial scenarios.

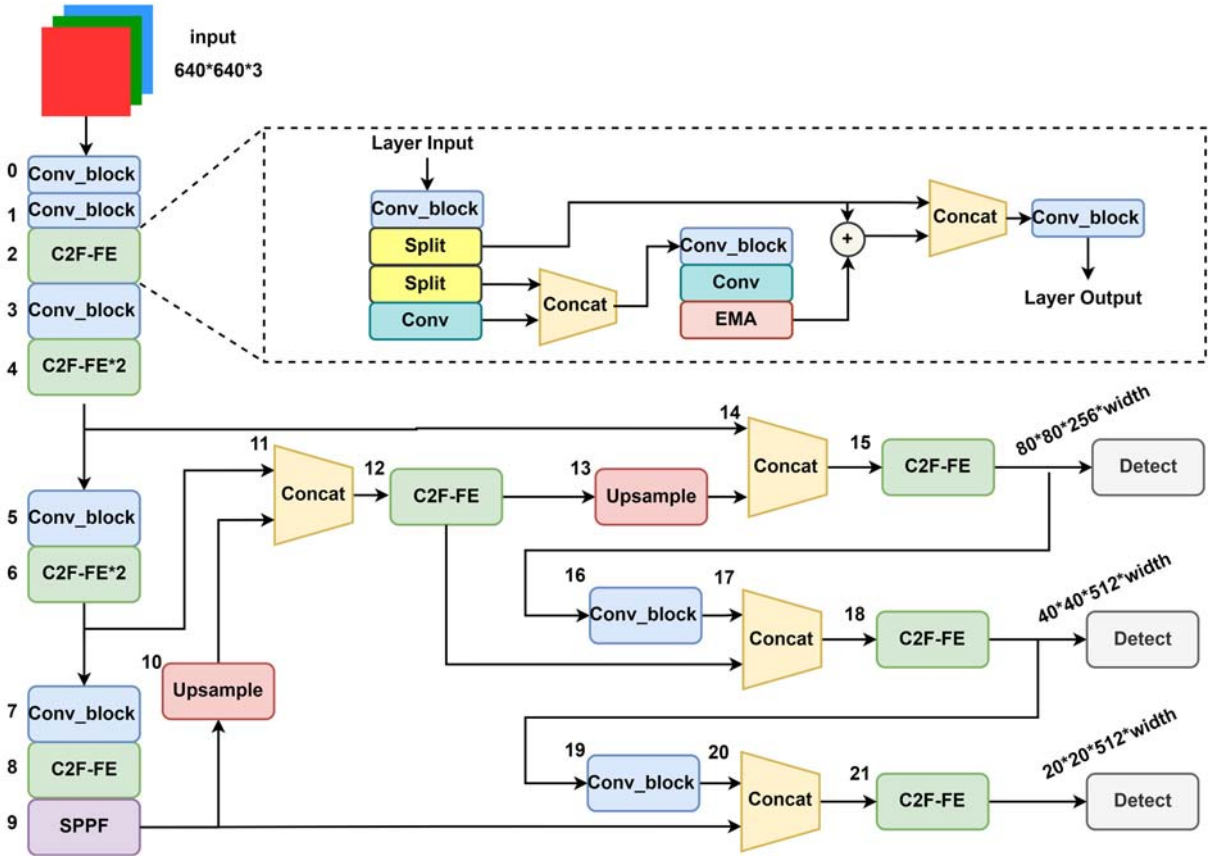


Figure 1. The architecture of the modified YOLOV8

2.2. Improved method

In this study, we introduce improvements to the YOLOV8 model by replacing the original C2F module with the C2F-FE module. The FasterNet within the C2F-FE module helps reduce redundant computations and memory access, enabling more efficient extraction of spatial features. It achieves higher running speeds across a diverse array of devices than other neural networks without sacrificing the precision of different visual tasks. In the FasterNet Block, partial convolutions (PConv) are applied, where only a few input channels are involved in feature extraction while the remaining channels remain

unchanged, as shown in Figure 2. PConv exhibits lower floating-point operations (FLOPs) than conventional convolutions while achieving higher FLOPs than depth-wise/group convolutions.

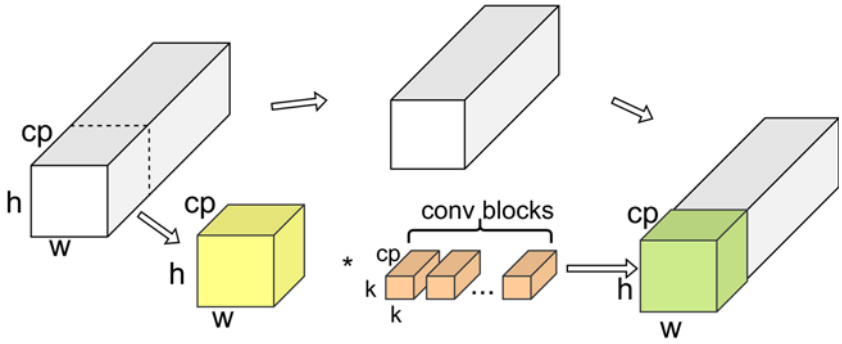


Figure 2. Partial convolution (PConv)

At the same time, this study introduces the EMA mechanism. Compared to other attention mechanisms such as CBAM, NAM [5], and CA [6], EMA performs better and demonstrates higher efficiency regarding required parameters. This module aims to maintain channel information while reducing computation workload. EMA divides the channel dimension into sub-features and evenly distributes spatial semantic features. Extensive trials on widely adopted benchmarks like ImageNet, MS COCO, and VisDrone2019 datasets [7] for object detection demonstrate that EMA achieves superior performance without altering the network depth. The EMA module is integrated into the FasterNet structure, resulting in the C2F-FE module, as depicted in Figure 3.

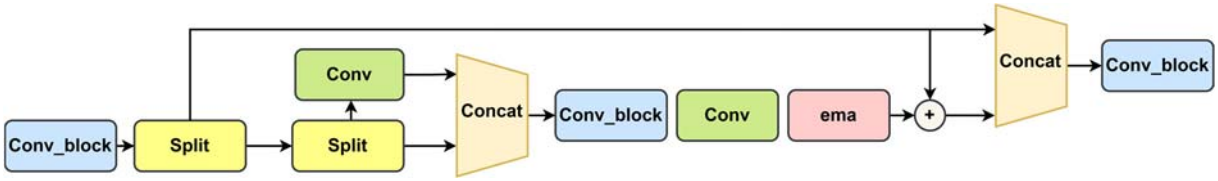


Figure 3. C2F-FE module

3. Experiments

3.1. Experimental details

The experiments were conducted on a computer with an Intel Xeon (R) W-2145 CPU and NVIDIA GeForce RTX 2080Ti GPU with 11 GB of VRAM, running the Ubuntu 18.04 operating system. PyTorch version 1.12.0, CUDA version 12.0, and Python-Opencv version 4.6.0 are used. The experimental code was modified based on the Ultralytics-YOLO framework, which offers ease of use, excellent performance, and scalability. It can be utilized for various object detection tasks, including image classification, object localization, and object segmentation. Additionally, the Ultralytics-YOLO framework supports multiple hardware platforms, including CPU, GPU, and TPU.

3.2. Dataset preparation

In the experiment, the Hard Hat Workers dataset [8] was used for model training and performance testing. The dataset consists of 7, 041 samples with corresponding labels and is divided into training and test sets in a 3:1 ratio. The dataset supports multiple formats, including Pascal VOC [9], COCO, and YOLOV8.

3.3. Evaluation metrics

The precision (accuracy) of the predicted bounding boxes is calculated by using the following formula:

$$precision = \frac{tp}{tp + fp} \tag{1}$$

where tp is a binary array indicating whether the detection results are correct; fp, short for false positive, is an occurrence in object detection where unrelated objects are incorrectly classified as target objects, indicating a detection result that is not accurate. Additionally, the formula for calculating the recall of the predicted bounding boxes is:

$$recall = \frac{tp}{n_1 + eps} \tag{2}$$

where the variable n_1 represents the number of labels. The default value for eps is $1e^{-16}$, which is used to avoid division by zero. To ensure that the recall curve and precision curve start at (0, 0) and end at (1, 1) respectively, sentinel values are introduced. The modified recall and precision curves are referred to as mrec (modified recall) and mpre (modified precision) respectively. Average Precision (AP) can be computed based on these curves:

$$\mathrm{AP} = \int_0^1 interp(x, \mathrm{mrec}, \mathrm{mpre})dx \tag{3}$$

where the Numpy [10] function np. interp (x, mrec, and mpre) performs linear interpolation based on the reference points mrec and their corresponding values mpre. It calculates the interpolated result at the given target point x. After that, by setting an Intersection over Union (IoU) threshold, we can obtain metrics such as mAP50 and mAP50-95, which provide a more comprehensive evaluation of the model's performance. These metrics consider the precision and recall values at different IoU thresholds to assess the model's accuracy across a range of overlap criteria. Additionally, the Generalized Intersection over Union (GIoU) loss function is used in this experiment. GIoU considers the position, size, and shape differences between bounding boxes. Therefore, GIoU is a suitable choice for the safety helmet detection task. The specific calculation is as follows:

$$\mathrm{GIoU} = \mathrm{IoU} - \frac{\mathrm{area}\left(B \cup \hat{B}\right) - \mathrm{area}(B)}{\mathrm{area}\left(B \cup \hat{B}\right)} \tag{4}$$

where IoU represents the intersection over the union between the predicted bounding box B and the ground truth labeled box $\widehat{\mathrm{B}}$. It is calculated as the ratio of the intersection area between the two boxes to their union area. The function calculates the area of the corresponding box.

3.4. Experimental results

The following experiments were conducted with 100 training iterations, using training images of size and a batch parameter of 100. Neural network models were trained and tested to compare the detection performance of YOLOV8 before and after the improvements.

Table 1. YOLOv8 model performance

Class	Precision	Recall	mAP50	mAP50-95
All	0.705	0.745	0.747	0.511
Head	0.852	0.960	0.962	0.675
Helmet	0.872	0.968	0.980	0.689
Person	0.393	0.306	0.300	0.170
Speed	1.2 ms			

Table 2. YOLOv8 improved model performance

Class	Precision	Recall	mAP50	mAP50-95
All	0.831	0.802	0.842	0.610
Head	0.915	0.968	0.976	0.719
Helmet	0.946	0.979	0.991	0.738
Person	0.63	0.459	0.559	0.373
Speed	0.7 ms			

Table 3. Different models for safety helmet detection

Models	Precision	Recall	mAP50	Average FPS
RPN	0. 597	0.894	0.779	101
Fast-RCNN	0.672	0.881	0.774	97
YOLOV5	0.893	0.872	0.846	105
Ours	0.946	0.979	0.991	135

When examining the data in Table 1 and Table 2, it can be deduced that the improved YOLOV8 demonstrates enhanced accuracy compared to the original version, while reducing the inference time. Table 3 presents the comparison results with other existing methods. The testing performance on four different industrial scenes is illustrated in Figure 4.

Figure 4. Improved version of YOLOv8 in safety helmet detection

4. Conclusion

In this research paper, we tackle the challenges of low FLOPS, redundant computations, and a limited balance between model generalization and accuracy in existing helmet detection algorithms. We propose an improved helmet detection method based on YOLOv8, which involves refining the YOLOv8 network architecture, incorporating the EMA attention mechanism and PConv into YOLOv8's bottleneck module, handling the helmet dataset, and training the model with specific parameters. Experimental results demonstrate that our improved helmet detection method performs excellently in testing accuracy and speed. The precision is improved by 0.126, mAP50 is increased by 0.095, and the inference speed is reduced. In future work, we will collect more positive and negative samples from real-world scenarios to enhance the generalization capability of the network. Additionally, we plan to deploy this algorithm on monitoring devices or detection robots for practical applications.

References

[1] Zarbin M. (2020). Artificial Intelligence: Quo Vadis? Trans. Vis. Sci. Tech., 9 (2): 1-1. https://doi.org/10.1167/tvst.9.2.1.

[2] Jocher G., Chaurasia A., and Qiu J. (2023). YOLO by Ultralytics (Version 8.0.0) [Computer software]. https://github.com/ultralytics/ultralytics.

[3] Ouyang D., He S., Zhang G., et al. (2023). Efficient Multi-Scale Attention Module with Cross-Spatial Learning. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Rhodes Island. 1-5. https://doi.org/10.1109/icassp49357.2023.10096516.

[4] Chen J., et al. (2023). Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. arXiv, May 21, 2023. Accessed: Jul. 05, 2023. https://doi.org/10.48550/arXiv.2303.03667.

[5] Liu Y., Shao Z., Teng Y., et al. (2021). NAM: Normalization-based attention module. arXiv preprint arXiv: 2111.12419, 2021. https://doi.org/10.48550/arXiv.2111.12419.

[6] Gu R., Wang G., Song T., et al. (2020). CA-Net: Comprehensive attention convolutional neural networks for explainable medical image segmentation. IEEE transactions on medical imaging, 2020, **40 (2)**: 699-711. https://doi.org/10.1109/TMI.2020.3035253.

[7] Du D., Zhu P., Wen L., et al. (2019). VisDrone-DET2019: The vision meets drone object detection in image challenge results. In: Proceedings of the IEEE/CVF international conference on computer vision workshops. Seoul, Korea (South). 0-0. https://doi.org/10.1109/ICCVW.2019.00030.

[8] Xie L. (2019). Hardhat URL. https://doi.org/10.7910/DVN/7CBGOS.

[9] Everingham M., Van Gool L., Williams C. K. I., Winn J., and Zisserman A. (2012). The PAS-CAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascalnetwork.org/challenges /VOC/voc2012/workshop/index.html.

[10] Harris C. R., Millman K. J., van der Walt S. J., Gommers R., Virtanen P., Cournapeau D., et al. (2020). Array programming with NumPy. Nature, 585:357–62. https://doi.org/10.1038/s41586-020-2649-2.