IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 December 2023, accepted 28 December 2023, date of publication 3 January 2024,
date of current version 11 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3349471

APPLIED RESEARCH

Real-Time Helmetless Detection System for Lift Truck Operators Based on Improved YOLOv5s

YUNCHANG ZHENG[1], **MENGFAN WANG**[1], **YICHAO LIU**[2], **CUNYANG LI**[1], **AND QING CHANG**[1]
[1]Hebei University of Architecture, Zhangjiakou, Hebei 075000, China
[2]Zhangjiakou Cigarette Factory Company Ltd., Zhangjiakou, Hebei 075000, China

Corresponding author: Qing Chang (cqing220@163.com)

This work was supported by the Basic Scientific Research Business Fund Project of Universities in Hebei Province under Grant 2022QNJS03, Grant 2022CXTD08, and Grant 2023JCTD05.

ABSTRACT Safety helmet plays a major role in protecting the safety of operators in industry, and several helmetless detection methods have been developed based on artificial intelligence. However, existing detection methods cannot work well for machinery operators in specific scenarios, such as factory environments, in which low accuracy and efficiency for small helmet targets could happen occasionally. Aiming to comprehensively handle these issues, this paper proposes a real-time helmetless detection method for lift truck operators based on improved YOLOv5s. Firstly, the lightweight multiscale attention EfficientViT is added to improve the detection accuracy for small-sized helmets. Secondly, the detector C^2 F-Net from Transformer structure is added to improve the predictability of challenging occurrences. In addition, the loss function is changed to Alpha-IoU, which further enhances the detection ability of small-sized targets. Finally, a real-time helmetless detection system is built with a set of well-designed detecting logic. The system effectively implements the proposed method and provides real-time monitoring and detection of helmetless lift truck operators. By conducting experiments on a self-created dataset derived from factory surveillance videos, this paper successfully validated the effectiveness of the proposed method. Specifically, the results showed that the proposed method increases the mAP (0.5) of the original algorithm in abnormal class by 6.7%, reaching 98.7%, and the mAP (0.5:0.95) is improved by 6.5%, indicating the proposed method shows significant improvement in real-time detection performance for operators not wearing safety helmets in specific scenarios.

INDEX TERMS YOLOv5, helmetless detection, EfficientViT, C^2F-Net, Alpha-IoU.

I. INTRODUCTION

Safety is a paramount concern in all industries, especially in factory environments with intricate machinery. Among the myriad risks confronted by machinery operators, head injuries stand out as a significant threat, primarily due to non-compliance with safety helmet usage protocols. Studies have shown that a substantial percentage of production accidents, approximately 67.95% of total incidents, can be attributed to the lack of proper safety helmet usage [1]. This compelling statistic highlights the urgent necessity to address the problem of non-compliance with safety regulations within factory settings.

Lift truck drivers, also known as lift truck operators, occupy pivotal roles in production operations, maneuvering heavy loads and assisting in various lifting tasks. Their involvement in these activities exposes them to substantial risks, particularly concerning head injuries. Wearing safety helmets stands as a fundamental requirement to mitigate the inherent dangers of head injuries for lift truck operators. However, insufficient safety awareness and non-compliance with safety helmet regulations persist as common challenges among construction operators, substantially elevating the likelihood of injuries. Consequently, there is immense practical value in implementing rigorous supervision of safety helmet usage among lift truck operators, facilitated by advanced safety helmet detection algorithms.

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

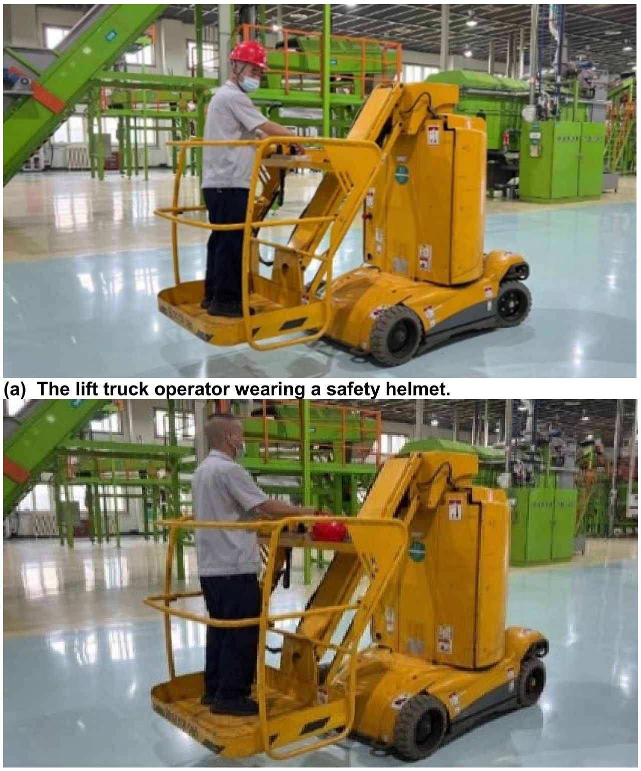
© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

With the development of artificial intelligence, deep learning-based object detection technology has made significant progress in various fields such as safety production supervision. Now, the field of object detection has seen an increasing research focus, with deep learning algorithms playing a crucial role. Two main types of computer vision algorithms, namely single-stage and two-stage detection, have been applied in this domain. Single-stage detection algorithms, including the YOLO series (such as YOLOv1 [2], YOLO9000 [3], YOLOv3 [4], YOLOv4 [5], YOLOv5 [6]), the SSD series (such as SSD [7],R-SSD [8], FSSD [9], and DSSD [10]), and EfficientDet [11], have been representative choices. On the other hand, two-stage detection algorithms, such as R-CNN [12], Fast R-CNN [13], Faster R-CNN [14] and Mask R-CNN [15], have also played dominant roles in this field.

However, both single-stage and two-stage detection algorithms may face challenges in capturing shallow features, which can result in the loss of semantic information, especially for small targets. To address this issue, researchers have proposed various approaches to combine shallow and deep features. In the year of 2017, Lin et al. conducted a seminal endeavor, unveiling the notion of feature pyramid network (FPN) [16] while providing an elaborate exposition of its network architecture and training methodology.

To enhance the inference speed and detection performance of YOLOv5s, and to achieve efficient and accurate object detection in resource-constrained environments, Han et al. introduced YOLOv5s GhostNet [17] in 2019. Howard et al. proposed in 2019 that MobileNetV3 [18] can be readily expanded and enhanced in the network architecture to accommodate various task and scenario demands. In pursuit of further improving model performance, attention mechanisms, like convolutional block attention module (CBAM) [19] and path aggregation network (PANet) [20], have also been utilized to enhance model performance by focusing on essential information, which have extensively optimized the YOLOv5's performance. In addition, Dillon Reis and colleagues proposed YOLOv8 [21] in 2023, which builds upon the YOLO algorithm. YOLOv8 enhances the feature expression in detecting small targets, employing a multiscale fusion approach and incorporating data augmentation strategies.

At the same time, many scholars have made efforts in the field of safety helmet detection. Jun et al. [22] modified the output dimension of the classifier in the YOLO based helmet detection algorithm to reduce the number of parameters. This algorithm has good real-time performance, but its accuracy is relatively low. Han et al. [23]currently use multi-scale detection methods to identify safety helmets and effectively improve recognition accuracy by adding a fourth dimension to predict a large number of small targets. However, their dataset only covers relatively standardized construction site environments, which may lead to performance degradation of the model when facing other types of construction sites or complex scenes. Wu et al. [24] have successfully addressed the challenges of helmet staining, partial occlusion, and multiple targets in low-resolution images by switching the backbone feature extraction network. Nevertheless, the algorithm imposes significant computational resource requirements. Tai et al. [25] aim to enhance the helmet recognition task by introducing a spatial attention module. This module aims to amplify the saliency of the spatial region where the target object is located, thereby capturing the helmet features more effectively. However, this approach may encounter certain constraints when dealing with small-sized or large-sized targets. Benjumea et al. [26] improved the YOLOv5 model to enhance its performance in detecting small targets, which is crucial for safety helmet wearing detection. They modified the model structure, creating a series of models called YOLO-Z, that exhibited impressive performance in detecting small targets. However, this improvement in mean average precision (mAP) came at the expense of a slower inference speed. Jin et al. [27] optimized the YOLOv5 algorithm for helmet-wearing detection, addressing issues that could affect detection accuracy. They used the k-means++ algorithm to improve the matching accuracy of prior anchor boxes, and added the depth wise coordinate attention mechanism to the backbone network, resulting in improved information dissemination between features. These enhancements resulted in higher detection accuracy, but increased computation complexity, potentially leading to longer training times or higher hardware requirements.



(a) The lift truck operator wearing a safety helmet.

(b) The lift truck operator without helmet.

FIGURE 1. Typical scenes of helmetless detection on the lift truck in factory environment.

However, as depicted in FIGURE 1, the current methods for helmetless detection face the following challenges on specific machinery such as lift trucks:

(1) Lift truck scale variations: Small targets on lift trucks can exhibit scale variations due to changes in distance and angle. At different distances and angles, small targets may appear on different scales, posing a challenge for detection algorithms.

(2) Complex environments: Due to the complex working environment in the factory workshop. The identification of personnel on lift trucks is challenged by obstacles, lighting conditions, similarities, and limitations in visibility. These factors can impact the accuracy and acquisition of comprehensive target information.

(3) Real-time requirements: Real-time performance is crucial during lift truck operations. Small target detection methods need to operate under real-time constraints to promptly identify and address potential safety hazards.

In order to address the mentioned issues, the main contributions of this paper include the following:

(1) To the best of our knowledge, we are the first researchers to detect helmetless people on specific machinery, especially on lift truck in factory environments. Due to the lack of public dataset for lift truck and operators, we have constructed a new dataset that includes three categories: lift trucks, operators wearing safety helmets, and operators not wearing safety helmets.

(2) By integrating EfficientViT's [28] into the backbone of YOLOv5s, not only can the detection accuracy for small helmets be significantly enhanced, but its impact extends to the field of detection system for lift truck operator. The efficient and powerful attention-based feature extraction capabilities of EfficientViT play a crucial role in improving the model's ability to capture fine-grained details, not only in the context of safety helmet detection but also in the detection system for lift truck operators. This integration allows the model to allocate computational resources more effectively, thereby enhancing the accuracy of detecting small and intricate details, including safety helmets in lift truck operator scenarios. As a result, the overall performance and reliability of the detection system for lift truck operators can be improved, ensuring a safer working environment and reducing the risk of accidents.

(3) By replacing the C3 module with the C^2F [29] module in the YOLOv5s model, the predictive capability improves while extending its impact on the detection system for lift truck operators. The C^2F module enhances the model's understanding of complex scenes by integrating information from low-level feature maps. This integration enables the model to accurately predict challenging events like partially obscured objects or objects with complex backgrounds commonly found in lift truck operator scenarios. The integration of the C^2F module makes the YOLOv5 model more effective in demanding real-world situations for lift truck operators. It improves object detection and prediction, providing valuable insights and enhancing situational awareness. Accurate detection of challenging events helps prevent accidents and ensures a safer working environment.

(4) By applying the Alpha-IoU Loss function, the YOLOv5 model's detection capability is enhanced for small targets relevant to the detection system for lift truck operators. This specific loss function assigns varying weights to the loss terms based on target size or difficulty, ensuring that smaller targets receive more attention during training. Consequently, the model becomes more proficient at detecting and predicting small objects or obstacles encountered in lift truck operator scenarios.

(5) We have developed an intelligent software system to implement helmetless detection, which has been applied in actual production environments. In the detection system, a set of well-designed warning logic was applied to detect helmetless operators on lift truck more accurately.

The rest of this paper is organized as follows. Section II introduces the original YOLOv5s network structure and its improvements and enhancements based on YOLOv5s. Section III introduces the experiment preparation and setup, and analyses the results of the experiment. Finally, the main conclusions are drawn in Section IV.

II. PRINCIPLES AND ENHANCEMENTS

A. YOLOv5 NETWORK STRUCTURE

YOLOv5 is a state-of-the-art object detection framework that builds upon the foundations of YOLOv3 and YOLOv4. It introduces continuous integration and innovation to enhance performance and accuracy. YOLOv5 comprises four different network models: YOLOv5s, YOLOv5m, YOLOv5l and YOLOv5x. This paper will focus on the YOLOv5s network model, which is the smallest in terms of network width and depth. The YOLOv5s network model consists of four main components: input, backbone, neck and output. Input is the initial stage where the input image is fed into the model. The backbone forms the core of the architecture and extracts hierarchical features from the input image. The neck component connects the backbone and output layers, performing further fusion and feature refinement. Lastly, the output module generates predictions by processing the refined features from the neck.

YOLOv5s is designed to balance speed and accuracy, making it useful for real-time applications with limited computational resources. It leverages advanced techniques such as anchor-free detection and focal loss to improve object detection performance.

1) INPUT

The image input serves as the starting point of the YOLOv5 network. During this stage, the input image undergoes Mosaic data augmentation, where four images are randomly combined through scaling, cropping, and arrangement. This technique increases dataset diversity and improves the network's robustness. The input image is then adaptively scaled to a fixed size, and the adaptive anchor box calculation is performed. Adaptive image scaling reduces the addition of black edges, improving the inference speed. Adaptive anchor box calculation enables the network to learn the optimal anchor box values based on the training data.

2) BACKBONE

The backbone network is responsible for extracting features from the input images. YOLOv5 offers flexibility in selecting different backbone network structures, such as Darknet [30], CSPDarknet [31] or EfficientNet [32]. Typically, the backbone network consists of convolutional layers and pooling layers, gradually capturing features at different scales, which are later used for object detection tasks.

Darknet is a classic backbone network structure that uses multiple convolutional layers and spatial sampling operations to effectively capture features at different levels. CSPDarknet introduces the cross-stage partial connections (CSP) module to further enhance feature representation. EfficientNet is a backbone network based on automated network structure search, achieving better performance with relatively small computational resources through optimizing network structure and depth scaling coefficients.

The backbone network gradually extracts features from images by stacking convolutional layers and reduces the spatial resolution of features using pooling layers. This is done to capture feature information at different scales, enabling effective detection of objects of different sizes during the object detection process. By utilizing different backbone network structures, YOLOv5 can adapt to different application scenarios and computational resources, providing more flexible and efficient feature extraction capabilities.

3) NECK

The integration and enhancement of the neck module further strengthen the capacity of feature representation. Common neck modules, such as the feature pyramid network (FPN) and the path aggregation network (PAN), utilize multiple convolutional layers, up sampling, and down sampling operations to fuse and connect feature maps from different levels of the backbone network. This facilitates the generation of feature maps with increased semantic information and multiscale feature representation. By merging and connecting feature maps from different levels of the backbone network, the neck module is able to synergize low-level and high-level feature information, thereby improving the capability of feature representation.

4) OUTPUT

The output detection head is responsible for object detection and prediction on the feature maps. YOLOv5 adopts an anchor-based approach to predict object bounding box positions and categories across different scales of feature maps. This is achieved through the utilization of convolutional and fully connected layers. These components collaborate in YOLOv5 to enable robust object detection. The incorporation of data augmentation, adaptive image scaling, and adaptive anchor box calculation enhances the network's performance and adaptability to different hardware and scenarios. the loss function CIoU_Loss of the anchor box is replaced with GIoU_Loss [33]. Weighted non-maximum suppression (NMS) [34] operation is adopted to filter target anchor boxes to improve the accuracy of target detection.

B. IMPROVED YOLOV5S DETECTION ALGORITHM

The original YOLOv5s algorithm utilizes a significant number of convolutional structures of Conv and Conv 3 × 3 in its backbone network and feature pyramids. As a consequence, it results in a high parameter count and slower detection speed. However, in regard to practical applications like mobile or embedded devices, it becomes challenging to implement large and intricate models. To meet the requirement of fast detection of whether the personnel operating on a lift truck are wearing safety helmets, the improved YOLOv5s method has made the following improvements to YOLOv5:

Backbone Replacement: In order to enhance the precision of small-sized helmet detection, we have implemented the EfficientViT methodology for refinement. This attention mechanism pays more attention to smaller helmet regions, improving the model's ability to accurately detect and classify helmets worn by operators.

C3 Module Replacement: In the neck of the original YOLOv5s model, the C3 module was replaced with the C^2F module. This replacement was made to improve the predictability of challenging events. The C^2F module incorporates advanced feature fusion techniques, enabling the model to capture more fine-grained details and effectively handle complex and demanding detection scenarios.

Alpha-IoU [35]Loss Function: To further enhance the detection ability of the original YOLOv5 model for small targets like safety helmets, we utilized the Alpha-IoU loss function. This loss function specifically addresses the challenges faced when detecting small objects. It encourages the model to prioritize accurate localization and classification of smaller targets, resulting in improved performance for detecting safety helmet usage.

Overall, the proposed method combines dataset construction, model architecture modification, loss function optimization, and alert mechanisms to enhance the detection of safety helmet usage by operators on lift trucks.

1) LIGHTWEIGHT MULTISCALE EFFICIENTVIT

Cai et al. proposed EfficientViT, which is an efficient vision transformer (ViT)architecture for high-resolution low computational visual recognition. As shown in FIGURE 2, EfficientViT replaces softmax attention with linear attention, and enhances its local feature extraction ability through deep convolution. EfficientViT maintains global and local feature extraction capabilities while enjoying linear computational complexity.

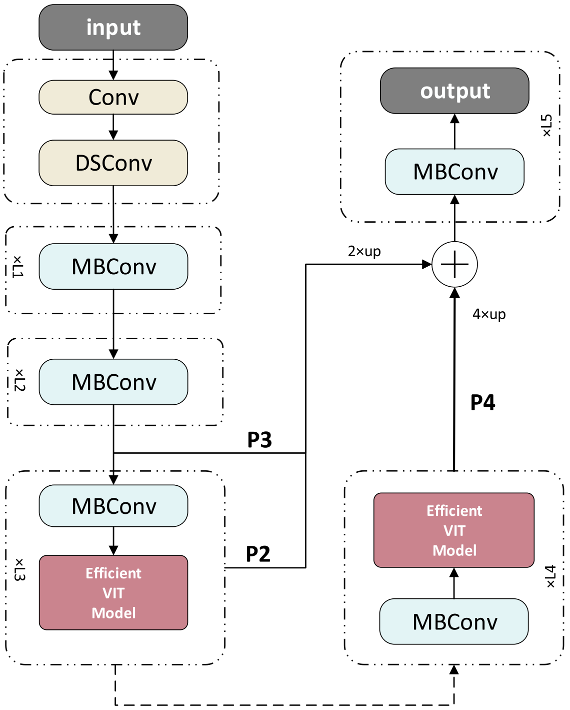


FIGURE 2. The structure diagram of efficientViT.

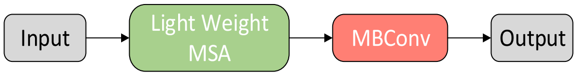


FIGURE 3. Illustration of efficientvit's building block.

Cai et al. build a new family of models based on the proposed lightweight multiscale attention (MSA) module. The core building block (denoted as 'EfficientViT Module') is illustrated in FIGURE 3. Specifically, an EfficientViT module comprises a lightweight MSA [36] module and an MBConv [37]. The lightweight MSA module is for context information extraction, while the MBConv is for local information extraction.

As depicted in FIGURE 4, MSA employs a method of processing input tokens. MSA initially acquires Q/K/V tokens through a linear projection layer. Subsequently, adjacent tokens are aggregated using lightweight small kernel convolution, resulting in multiscale tokens. These multiscale tokens are processed through global attention with ReLU activation, and the output is cascaded and passed to the final linear projection layer for feature fusion.

In order to enhance the multiscale learning capability of the ReLU-based global attention, the EfficientViT module proposes a method of aggregating information from neighboring Q/K/V tokens to obtain multiscale tokens. This information aggregation process is independent for each Q, K, and V in each head. EfficientViT solely utilizes small kernel convolution for information aggregation to avoid compromising hardware efficiency. Nevertheless, executing these aggregation operations independently proves to be less efficient on GPUs during practical implementation. Therefore, EfficientViT leverages the group convolution infrastructure in modern deep learning frameworks to consolidate all depth-wise convolutions (DWConv) into a singular DWConv and merge all 1 × 1 convolutions into a single 1 × 1 grouped convolution, where the number of groups is 3 × # heads and the channel count within each group is d.

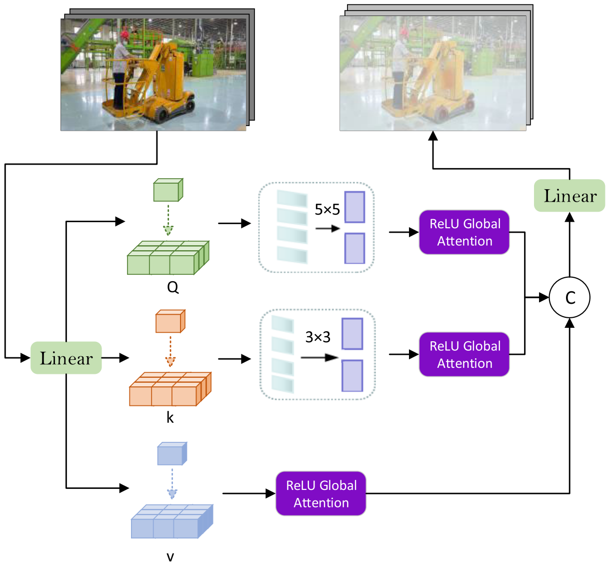


FIGURE 4. The proposed lightweight MSA.

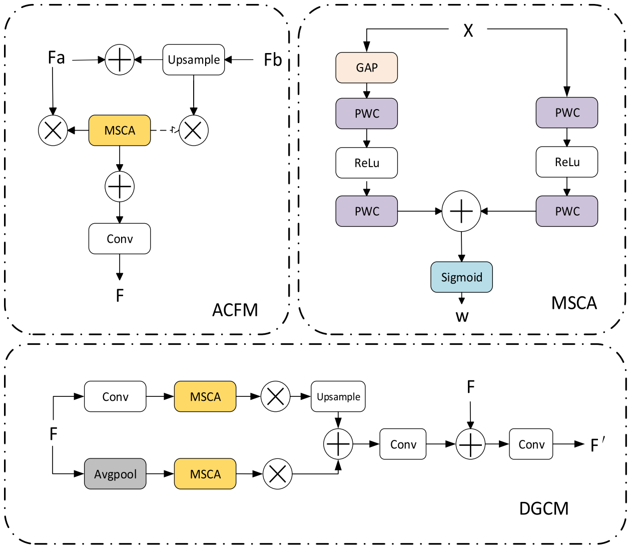
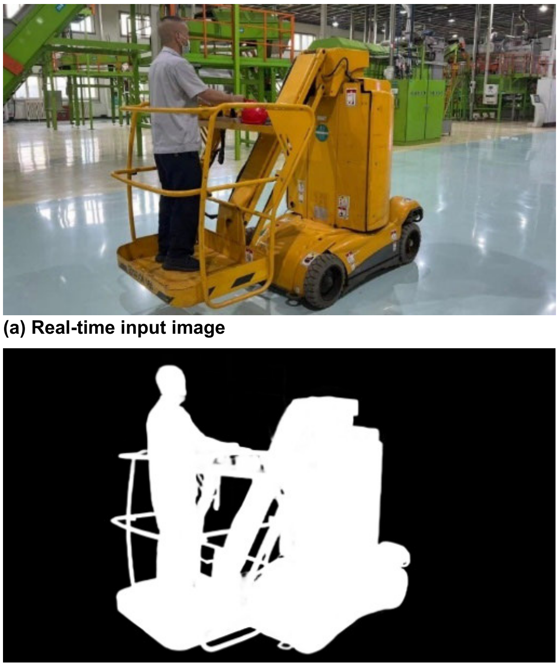


FIGURE 5. ACFM network (upper left), MSCA network (upper right) and dgcn network (below).

Upon obtaining the multiscale tokens, they undergo a global attention operation to extract multiscale global features. Finally, the features from different scales are concatenated along the head dimension and passed to the final linear projection layer for feature fusion.

2) CONTEXT-AWARE CROSS-LEVEL FUSION NETWORK (C^2F-NET)

Due to the low boundary contrast between objects and their surrounding environment in the production workshop



(a) Real-time input image

(b) Real-time output image

FIGURE 6. Real time photo of C2F module inspection.

of the factory, false detection may occur when detecting whether operators are wearing safety helmets on lift trucks. Sun et al. [29] proposed C^{2}F-Net to solve challenging COD tasks.

C^{2}F-Net is a method used to improve the detection performance of disguised objects, which combines cross-level features with context awareness. As shown in FIGURE 5, the detailed introduction of C^{2}F-Net's key components and operational processes are as follows:

1. Feature extraction: Use the Res2Net-50 model to extract the features of the input image and obtain five different levels of feature representation, represented as fi (i=1, 2..., 5). These features represent information of different granularity, from lower-level details to higher level abstract semantic information.

2. Receiving domain block (RFB): For each feature layer fi, we use an RFB module to expand its receptive field in order to capture richer feature information. The RFB module consists of five branches bk (k=1, 2, …, 5), with the first branch passing through 1 × 1 convolutional layer, reducing the number of channels to 64, and then passes through the (2k-(1) × (2k-(1) convolutional layer and 3 × 3 convolutional layers are used to process features and a specific expansion rate (2k-(1) is used when k>2.

3. Feature fusion: The features of the first four branches are concatenated and passed through 1 ×1 convolution operation, reducing the size of channels to 64. Then, the features of the fifth branch are added to it to obtain the final fusion feature. This can fuse features with different receptive fields and representation capabilities together, improving the detection performance of camouflaged objects.

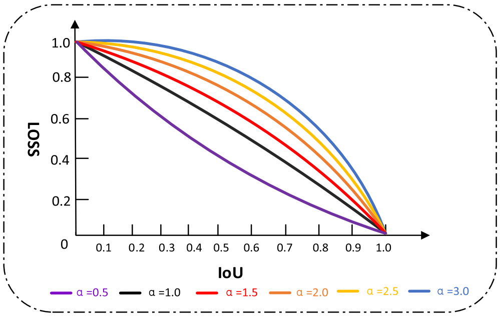


FIGURE 7. Correlation between IoU and Lα-IoU.

4. Attention guided cross-level fusion module (ACFM), as shown in the upper left image of FIGURE 5. In order to further integrate features from different levels, we introduced the ACFM module. The ACFM module utilizes attention mechanisms to dynamically adjust the weights of features at different levels, in order to better fuse and utilize their information.

5. Dual branch global context module (DGCM), as shown in FIGURE 5. In order to mine multiscale context information in fused features, we use the DGCM module. The DGCM module obtains more global and rich contextual information through two parallel branches, each branch has an MSCA module (upper right in FIGURE 5) utilizing global contextual information and convolution operations with different receptive fields.

Finally, the features obtained by combining ACFM and DGCM modules are used for detecting camouflaged objects. Through effective feature extraction, cross-level fusion, and the utilization of contextual information, C^{2}F-Net can improve the performance of camouflage object detection and achieve more accurate results. The input real-time photo and the output image detected by the C^{2}F module are shown in FIGURE 6.

3) ALPHA-IOU LOSS FUNCTION

YOLOv5, in its original implementation, utilizes the generalized intersection over union (GIoU) [33] to handle prediction boundary boxes. This approach effectively handles cases where there is no intersection between the predicted bounding box and the actual bounding box. However, GIoU suffers from fairness issues when dealing with bounding boxes of different scales and aspect ratios.

The Alpha-IoU loss function balances the impact of prediction errors on objects of different scales by introducing an adjustable parameter α. This enables the model to more accurately regress bounding boxes when dealing with multiscale targets. The Alpha-IoU loss function can be defined as follows:

The ordinary IoU loss, defined as $\mathcal{L}_{\alpha-\mathrm{IoU}} = 1 - IoU$, can be derived from the expression of the α-IoU loss function through the application of Box-Cox transformation:

$$\mathcal{L}_{\alpha-\mathrm{IoU}} = \frac{1 - IoU^{\alpha}}{\alpha}, \alpha > 0. \tag{1}$$

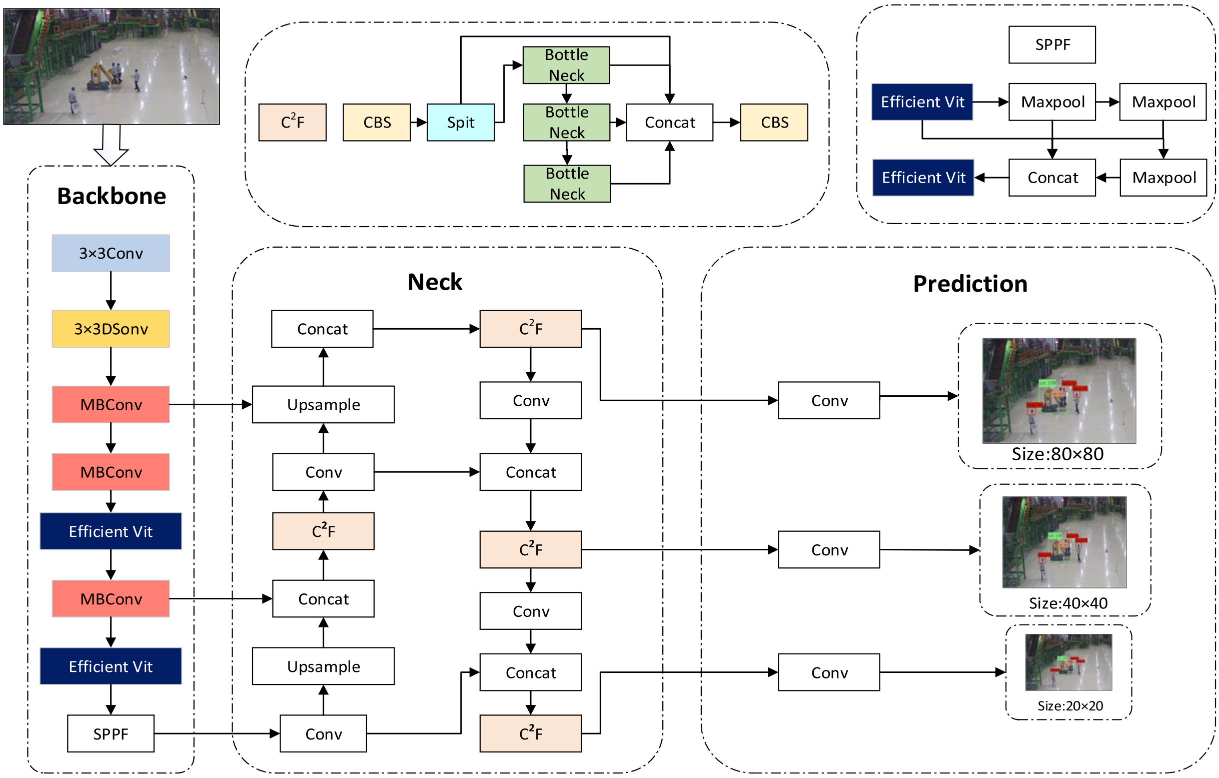


FIGURE 8. The improved YOLOv5s structure.

Utilizing the condition $\alpha > 0$ and simplifying the expression of the loss function $\alpha \to 0$, as in this case, the denominator α in equation (1) is just a positive constant in the objective. This gives us two cases of the α-IoU loss for $\alpha \to 0$ and $\alpha 0$, respectively:

$$\mathcal{L}\alpha - \text{IoU} = \begin{cases} -\log(IoU), & \alpha \to 0, \\ 1 - IoU^{\alpha}, & \alpha \nrightarrow 0. \end{cases} \tag{2}$$

Then, the α-IoU loss is extended for $\alpha \nrightarrow 0$ to a more general form by introducing a power regularization term into the formula:

$$\mathcal{L}_{\alpha\text{-IoU}} = 1 - IoU^{\alpha_1} + P^{\alpha_2}(B, B^{\text{gt}}) \tag{3}$$

In the formula: $\alpha_1 > 0$, $\alpha_2 > 0$, $P^{\alpha_2}(B, B^{\text{gt}})$ indicates that based on B and B^{gt}, this extension directly summarizes the existing IoU base loss to its α-IoU version. According to equation (4), the commonly used IoU losses can be included. $\mathcal{L}_{I\text{oU}}$, $\mathcal{L}_{\text{G}I\text{oU}}$, $\mathcal{L}_{\mathcal{D}I\text{oU}}$ and $\mathcal{L}_{\mathcal{C}I\text{oU}}$ use the same parameter α for IoU and penalty terms:

$$\begin{aligned}
\mathcal{L}_{I\text{oU}} &= 1 - IoU \Rightarrow \mathcal{L}_{\alpha-\text{IoU}} = 1 - IoU^{\alpha}, \\
\mathcal{L}_{\text{GIoU}} &= 1 - IoU + \frac{|\text{C}\backslash(\text{B} \cup \text{B}^{gt})|}{|\text{C}|} \Rightarrow \mathcal{L}_{\alpha-\text{GIoU}} \\
&= 1 - IoU^{\alpha} + (\frac{|\text{C}\backslash(\text{B} \cup \text{B}^{gt})|}{|C|})^{\alpha}, \\
\mathcal{L}_{\text{DIoU}} &= 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} \Rightarrow \mathcal{L}_{\alpha\text{-DIoU}} \\
&= 1 - IoU^{\alpha} + \frac{\rho^{2\alpha}(\text{b}, \text{b}^{gt})}{c^{2\alpha}}, \\
\mathcal{L}_{\text{CIoU}} &= 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \beta\nu \Rightarrow \mathcal{L}_{\alpha-\text{CIou}} \\
&= 1 - IoU^{\alpha} + \frac{\rho^{2\alpha}(\text{b}, \text{b}^{gt})}{c^{2\alpha}} + (\beta\nu)^{\alpha},
\end{aligned} \tag{4}$$

where C in $\mathcal{L}_{\text{GIoU}}$ denotes the smallest convex shape enclosing B and B^{gt};b and b^{gt} in $\mathcal{L}_{\text{DIoU}}$ denote central points of B and B^{gt} with $\rho(\cdot)$ being the Euclidean distance and c being the diagonal length of the smallest enclosing box; and in $\mathcal{L}_{\text{C}I\text{oU}}$, $\nu = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2$, $\beta = \frac{\nu}{(1-IoU)+\nu}$. When α =1, the transformation form of $\mathcal{L}_{I\text{oU}}$, $\mathcal{L}_{\text{G}I\text{oU}}$, $\mathcal{L}_{\mathcal{D}I\text{oU}}$ and $\mathcal{L}_{\text{C}I\text{oU}}$ becomes the prototype.

As shown in FIGURE 7, when $\alpha > 1$, learn easy examples first; when IoU =1, the difficult examples will gradually begin to learn; when $0 < \alpha < 1$, it tends to decrease the final performance, reducing the loss and gradient of high IoU objects will lead to worse localization of the object. The different forms of $\mathcal{L}\alpha$-IoU are adapted to the IoU value of the target, which provides greater flexibility for model training and can achieve different levels of target box regression accuracy.

4) THE IMPROVED YOLOV5s

The structure of the improved YOLOv5s method is shown in FIGURE 8. Firstly, we substitute the backbone of YOLOv5s with the EfficientViT network architecture. Once the image is

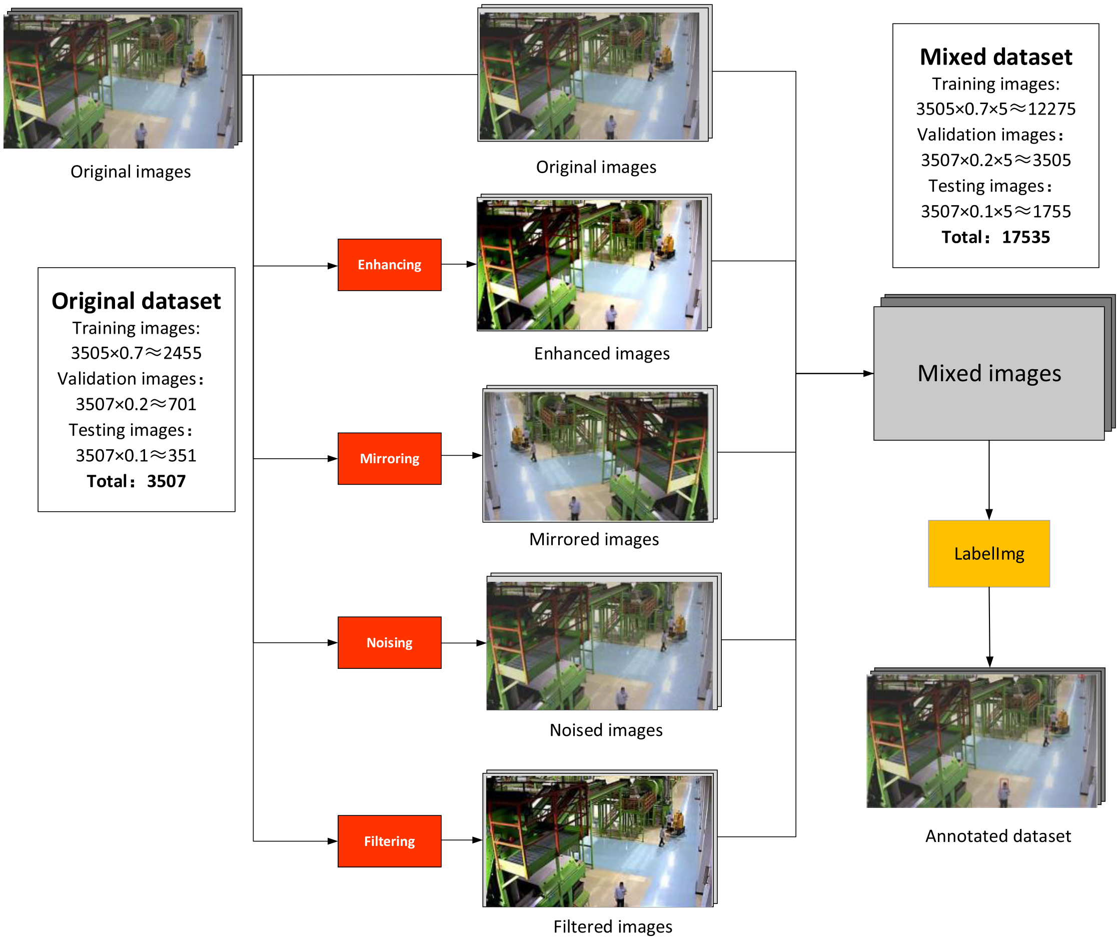


FIGURE 9. Image pre-processing and annotation.

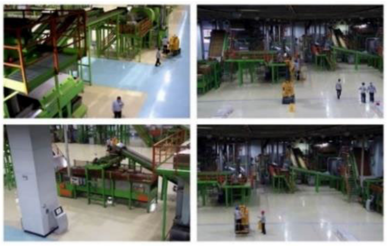


FIGURE 10. Sample images extracted from surveillance videos.

fed into the model, multiple layers of convolutional networks will be utilized for feature extraction and image classification. As for the neck section0, the C3 structure is replaced by C^2F. In the output phase, we replace GIoU with Alpha-IoU.

III. EXPERIMENTS AND DISCUSSION

The experimental procedure can be mainly categorized into three phases: dataset generation, model training, and object detection. In the initial stages, we employed image preprocessing techniques to construct a comprehensive dataset by gathering images. Subsequently, by fine-tuning the parameters and training the models, we obtained the model weights for our algorithm. Finally, utilizing the meticulously trained weights, we conducted helmetless detection and conducted a comparative analysis of detection results from various methodologies.

A. DATASET AND PRE-PROCESSING

Due to the unavailability of publicly accessible dataset, the experimental dataset was sourced from internal shots. The location of the shots is a workshop scene within a specific enterprise factory. After processing, a grand total of 3507 images were acquired. The distribution of these images for training, validation, and testing was performed in a random manner, maintaining a ratio of 7:2:1. Some sample images are shown in FIGURE 10.

To tackle the problem of uneven sample distribution and ensure dataset alignment with real-world situations, we have employed four image preprocessing methods for each data group. These methods include image enhancing, mirroring, noising and filtering. Following the image preprocessing stage, we have curated a custom dataset comprising a total of 17535 images, consisting of both original and preprocessed

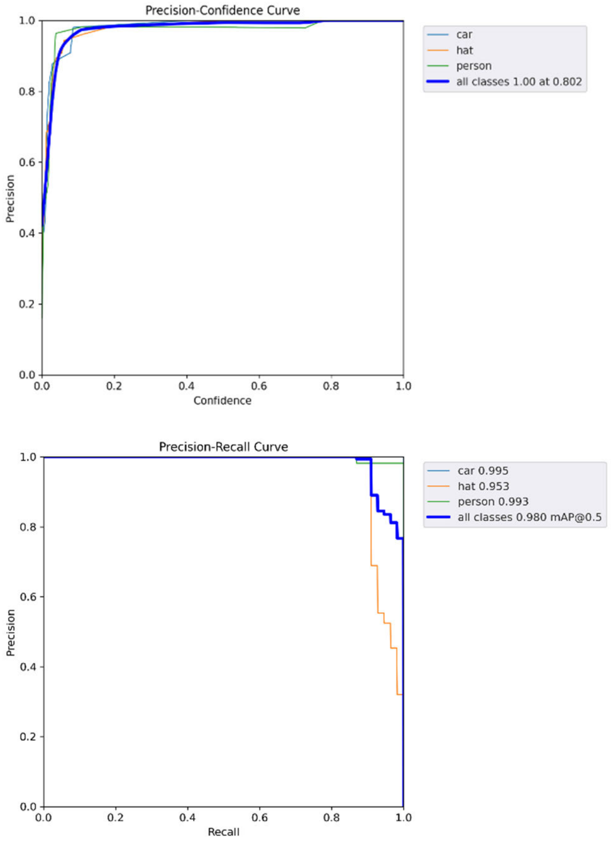


FIGURE 11. The precision-confidence curve and precision-recall curve.

images. The annotation of the dataset was carried out using the LabelImg software. As shown in FIGURE 9, to enhance the evaluation of object detection performance in complex environments, the dataset was annotated into three categories: person, hat and car, which respectively represent operator, helmet and lift truck. The annotation information of the dataset is saved in XML files adhering to the PASCAL VOC format.

B. NETWORK TRAINING

In the realm of network training, a sophisticated experimentation and development platform is established utilizing Windows 10 (64-bit). The central processing unit (CPU) is configured with the cutting-edge 11th generation Intel (R) Core (TM) i7-13900K CPU, operating at a remarkable 5.40 GHz. Additionally, the graphics processing unit (GPU) boasts the commendable NVIDIA GeForce RTX 3080, endowed with a substantial 10GB of memory. The CUDA framework employed in this endeavor operates under the prestigious 11.3 version. Python, a prominent programming language, stands at version 3.8. The deep learning framework harnessing the power of artificial intelligence in these experiments is none other than PyTorch.

Before the network training, it is essential to fine-tune the hyperparameters to achieve the best model performance and avoid overfitting. The chosen batch size is 32, and a learning rate of 0.001 is assigned. The optimizer selected is Adam, and 150 iterations are performed. The precision-confidence curve and precision-recall curve are shown in FIGURE 11.

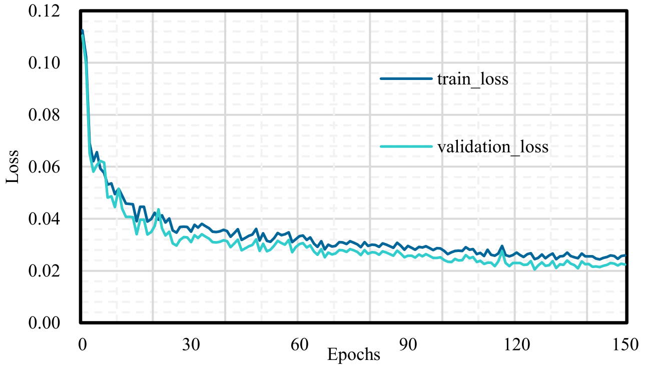


FIGURE 12. The loss curve of the training results.

FIGURE 12 illustrates a rapid decrease in the loss function value from 0 to 50 iterations, followed by a gradual decline from 50 to 150 iterations. After 150 iterations, the loss value stabilizes, indicating that the model has reached its optimal state.

C. EVALUATION INDICATORS

The evaluation metrics utilized in this paper comprise precision, recall, average precision (AP), and mean average precision (mAP). Precision refers to the likelihood that all positively predicted samples by the model are indeed positive samples. Recall represents the probability of the model correctly identifying positive samples among the total number of actual positive samples. Equations (5) and (6) respectively express precision and recall.

$$\text{Precision} = \frac{T_{\text{p}}}{T_P + F_P} \tag{5}$$

$$\text{Recall} = \frac{T_{\text{p}}}{T_P + F_N} \tag{6}$$

where TP (True Positives) represents the accurate prediction of positive samples by the model. Similarly, FP (False Positives) represents the incorrect prediction of positive samples, while FN (False Negatives) represents the incorrect prediction of negative samples. AP is a crucial performance metric that aims to eliminate the dependency on a single confidence threshold. It is calculated as the average precision under the precision-recall curve, as indicated by Equation (7). On the other hand, mAP is commonly utilized to assess the combined precision and recall results. It is computed by averaging the AP values across all the considered classes, and can be represented by Equation (8).

$$\text{AP} = \int_0^1 \text{Pr ecision}\,(t)\,\mathrm{d}t \tag{7}$$

$$\text{mAP} = \frac{1}{N}\sum_{\text{i}=1}^{N} AP_{\text{i}} \tag{8}$$

TABLE 1. Ablation experiments.

Model	+EfficientVit	+C^{2}F	+Alpha-IoU	Precision	Recall	mAP (0.5)	mAP (0.5:0.95)
YOLOv5s				87.6	84.9	92.0	80.2
-	√			93.1	92.6	94.5	85.2
-		√		92.3	93.2	94.1	84.1
-			√	89.9	92.8	94.4	84.3
-	√	√		94.3	**97.1**	96.3	84.7
Our Method	√	√	√	**99.8**	96.9	**98.7**	**86.7**

TABLE 2. The mAP (0.5) comparison of different methods through different epochs.

Model	Epoch=0	Epoch=30	Epoch=60	Epoch=90	Epoch=120	Epoch=150
SSD	0	0.88755	0.90125	0.91055	0.91661	0.91880
YOLOv5s	0	0.89050	0.91265	0.91884	0.91998	0.92036
YOLOv5s GhostNet	0	0.93654	0.95011	0.95804	0.96021	0.96701
YOLOv5s MobileNetV3	0	0.90367	0.92395	0.93702	0.94021	0.94068
YOLOv8	0	0.93256	0.94901	0.95458	0.96021	0.96356
Our Method	**0**	**0.95090**	**0.96758**	**0.97013**	**0.97889**	**0.98018**

Furthermore, experiments often utilize mAP (0.5) and mAP (0.5:0.95) as evaluation metrics. mAP (0.5) refers to the mean average precision with the Intersection over Union (IoU) threshold set to 0.5. On the other hand, mAP (0.5:0.95) represents the mAP calculated across a range of IoU thresholds from 0.5 to 0.95.

D. ABLATION EXPERIMENT

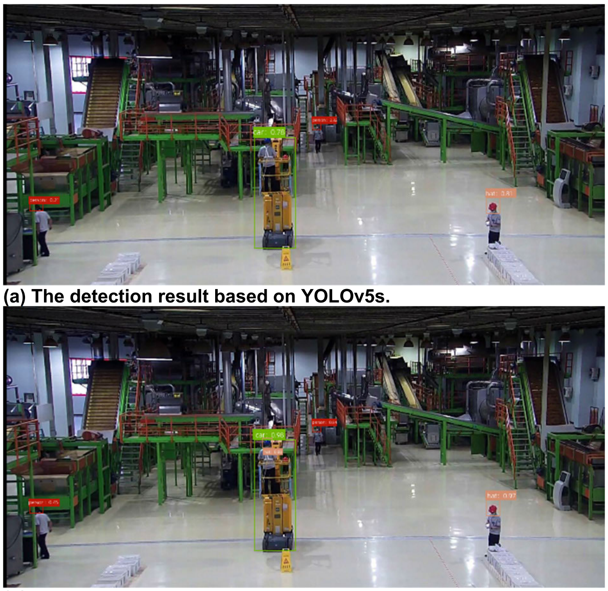
We conducted five sets of ablation experiments using the same environment and parameter settings, aiming to accurately evaluate the impact of each enhanced component on helmet detection. In these experiments, we used the YOLOv5s model as the baseline and evaluated it by comparing the experimental results. The experimental results are shown in TABLE 1.

Compared with the original YOLOv5s, when only introducing EfficientVit, the Precision increased by 5.5%, Recall increased by 7.7%, mAP (0.5) increased by 2.5%, mAP (0.5:0.95) increased by 5%.

Compared with the original YOLOv5s, when only introducing C^{2}F, the Precision increased by 4.7%, Recall increased by 8.3%, mAP (0.5) increased by 2.1%, mAP (0.5:0.95) increased by 3.9%.

Compared with the original YOLOv5s, when only introducing Alpha-IoU, the Precision increased by 2.3%, Recall increased by 7.9%, mAP (0.5) increased by 2.4%, mAP (0.5:0.95) increased by 4.1%.

Compared with the original YOLOv5s, when adding EfficientVit and C^{2}F modules, the Precision increased by 6.7%, Recall increased by 12.2%, mAP (0.5) increased by 4.3%, mAP (0.5:0.95) increased by 4.5%.



(a) The detection result based on YOLOv5s.

(b) The detection result based on our method.

FIGURE 13. Comparison of detection results between the original YOLOv5s and our method.

When all the enhancement strategies are implemented in combination, compared to YOLOv5s, the mAP (0.5) increases by 6.7%, the mAP (0.5:0.95) increases by 6.5%, the Precision increases by 12.2%, and the Recall increases by 12%. Although the improvement in recall was not as high as when only EfficientVit and C^{2}F modules were added, there were significant improvements in Precision, mAP (0.5), and mAP (0.5:0.95).

In conclusion, the results of the ablation experiments demonstrate the superiority of our method.

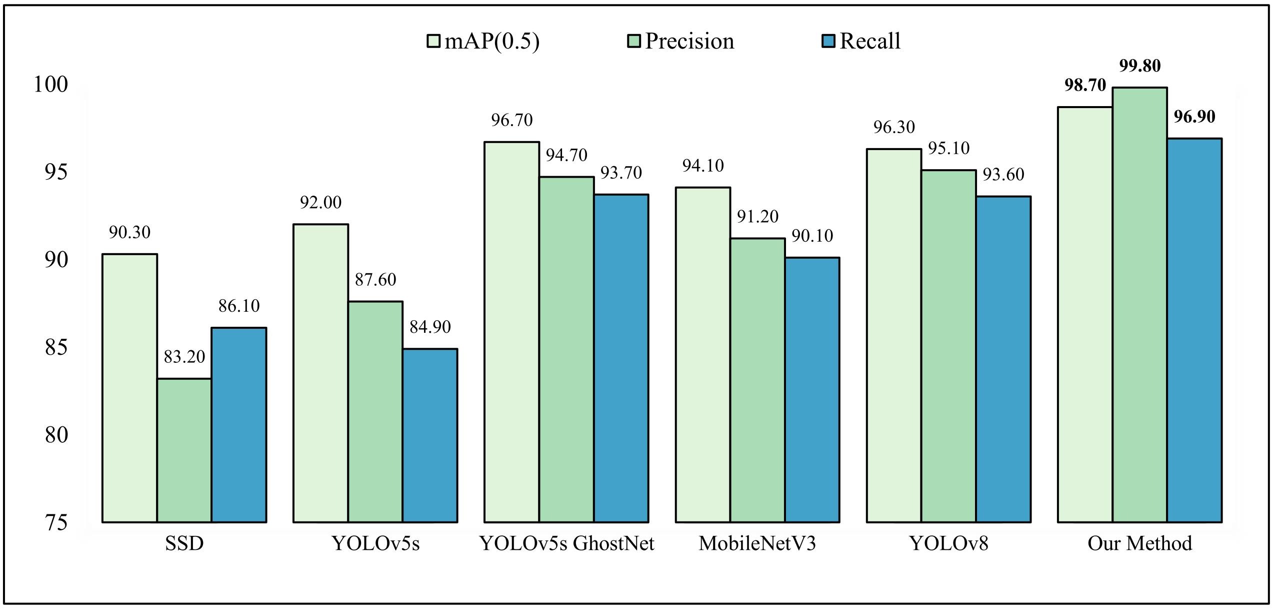


FIGURE 14. The mAP (0.5), Precision and Recall of different methods.

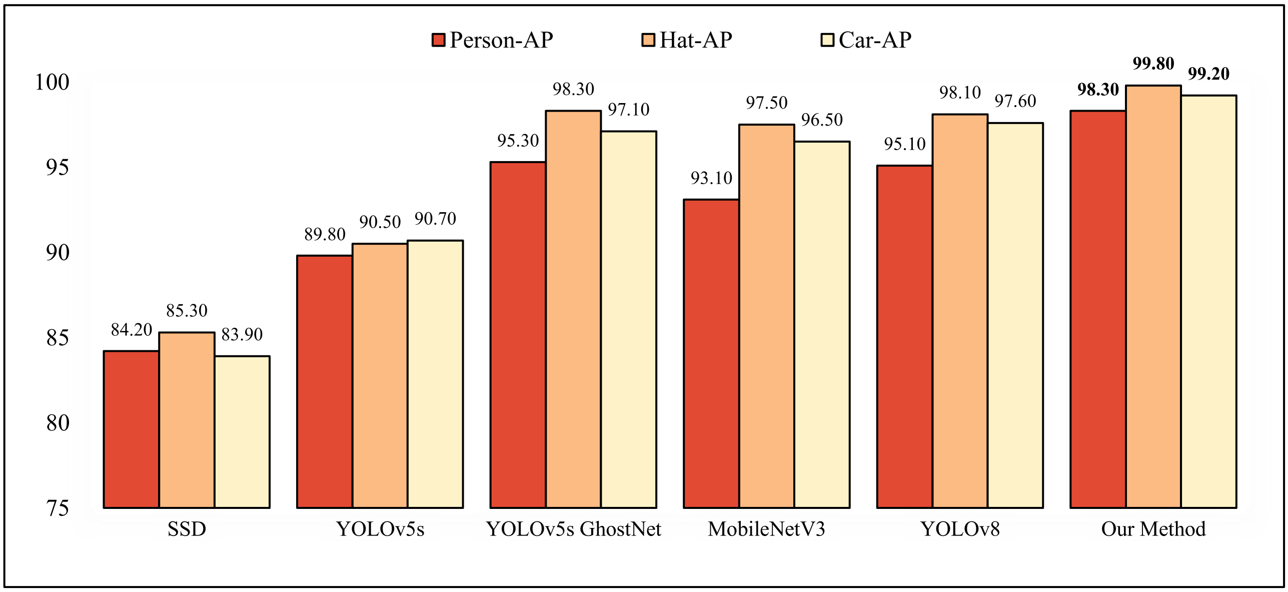


FIGURE 15. The Person-AP, Hat-AP and Car-AP of different methods.

TABLE 3. Comparison results of different methods.

Model	Person-AP	Hat-AP	Car-AP	mAP (0.5)	Model Size/MB
SSD	84.2	85.3	83.9	90.3	92.1
YOLOv5s	89.8	90.5	90.7	92.0	12.9
YOLOv5s GhostNet	95.3	98.3	97.1	96.7	7.79
YOLOv5s MobileNetV3	93.1	97.5	96.5	94.1	6.67
YOLOv8	95.1	98.1	97.6	96.3	22.5
Our Method	**98.3**	**99.8**	**99.2**	**98.7**	**6.03**

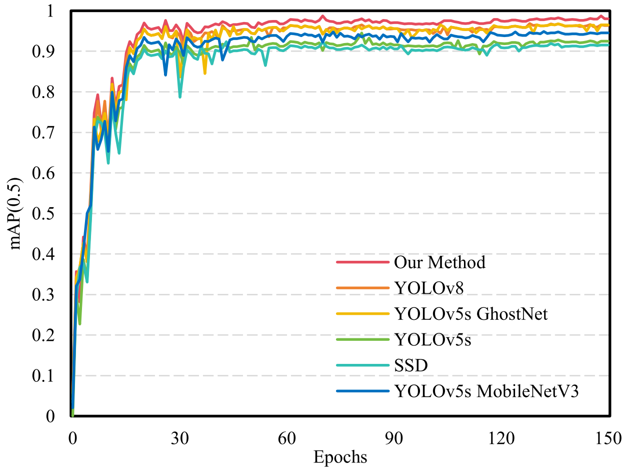


FIGURE 16. **The mAP (0.5) of related algorithms in the training process.**

In order to visually evaluate the detection performance of our approach, a comparison was conducted between the original YOLOv5s and our method. For evaluation, a subset of images from the dataset was randomly chosen, as shown in FIGURE 13. Under complex background conditions, the YOLOv5s model occasionally results in missed detection. For instance, in FIGURE 13(a) on the left, when an operator is working on a lift truck in a complex environment, the YOLOv5s fails to detect the operator wearing a safety helmet, leading to an undetected instance. In contrast, the proposed method successfully detects objects that the YOLOv5s model missed, as shown in FIGURE 13(b), proving the proposed algorithm effectively addresses the challenges of object detection in complex backgrounds.

E. COMPARATIVE EXPERIMENTS

To further validate the detection performance of the proposed method, our method was contrasted with multiple one-stage object detection algorithms, including SSD and YOLOv5s, alongside well-regarded approaches like YOLOv8, YOLOv5s GhostNet and YOLOv5s MobileNetV3.

During the training process, we compared the mean Average Precision (mAP) at an Intersection over Union (IoU) threshold of 0.5 between our method and other approaches for approximately 150 epochs.

FIGURE 16 and TABLE 1 demonstrate that the mAP (0.5) for each method exhibits a significant increase in the initial 30 epochs and stabilizes as the epoch count reaches 120. However, the mAP (0.5) of our method rises faster and finally stabilizes at the maximum value of 0.98, which is higher than other methods.

In the detection process, we also conducted comparative experiments using our method and the above-mentioned algorithms. In the same experimental environment, the weight file with the most optimal training performance is saved. The evaluation metrics used for comparative experiments encompass AP, mAP (0.5), model size, precision, and recall. In addition, Person-AP, Hat-AP and Car-AP respectively represent the AP of operator, helmet and lift truck.

As shown in Table 2, FIGURE 14 and FIGURE 15, our algorithm has been optimized in terms of detection accuracy and model size.

Compared to SSD, the Precision and Recall have improved by 16.8% and 10.8%, the mAP (0.5) has improved by 8.4%, the Person-AP, Hat-AP and Car-AP increase significantly, and the model size decreases by 93.5%.

Compared to the original YOLOv5s, the Precision and Recall have improved by12.2% and 12.0%, the mAP (0.5) has improved by 6.7%, the Person-AP, Hat-AP and Car-AP increase by 8.5%, 9.3% and 8.5%, resulting in a substantial decrease in the model size by 53.3%.

Compared to YOLOv5s GhostNet, the Precision and Recall have improved by 5.1% and 3.2%, the mAP (0.5) has improved by 2.0%, the Person-AP, Hat-AP and Car-AP increase by 3.0%, 1.5% and 2.1%, resulting in a substantial decrease in the model size by 22.6%.

Compared to YOLOv5s MobileNetV3, the Precision and Recall have improved by 8.6% and 6.8%, the mAP (0.5) has improved by 4.6%, the Person-AP, Hat-AP and Car-AP increase by 5.2%, 2.3% and 2.7%.

Compared to YOLOv8, the Precision and Recall have improved by 4.7% and 3.3%, the mAP (0.5) has improved by 2.4%, the Person-AP, Hat-AP and Car-AP increase by 3.2%, 1.7% and 1.6%, the model size decreases by 73.2%.

Taking into account the complexity of each model and the actual outcomes of detection, it can be generally inferred that our proposed method outperforms other approaches in terms of helmetless detection.

In order to compare the detection effects of different methods more visually, we performed multiple sets of comparison experiments for different scenarios.

According to Close-up scene 1 in FIGURE 17, when detecting the helmetless operator at close range, there are certain differences in the performance of each algorithm in identifying helmets and lift trucks. In terms of confidence in detecting safety helmets and lift trucks, both SSD and YOLOv5s exhibit lower confidence, indicating relatively poor accuracy in identifying these two objects. In contrast, YOLOv5s Ghost, YOLOv5s MobileNetV3 and YOLOv8 demonstrate some advancements in the detection of helmets and lift trucks compared to the original SSD and YOLOv5s. As a comparison, our method outperforms the other five algorithms, achieving a confidence level of 0.99.

According to the Vast-Extent scene 2 in FIGURE 17, SSD, YOLOv5s, YOLOv5s Ghost and YOLOv5s MobileNetV3 fail to detect the person at the edges of the image, resulting in missed detection. Only YOLOv8 and our method do not experience any missed detections. However, the confidence score for person detection by YOLOv8 is only 0.31, relatively low. On the other hand, our method achieved a confidence score of 0.77 for person detection at the image edges. Furthermore, when detecting lift truck, safety helmets, and persons in the image, our method demonstrated higher confidence scores

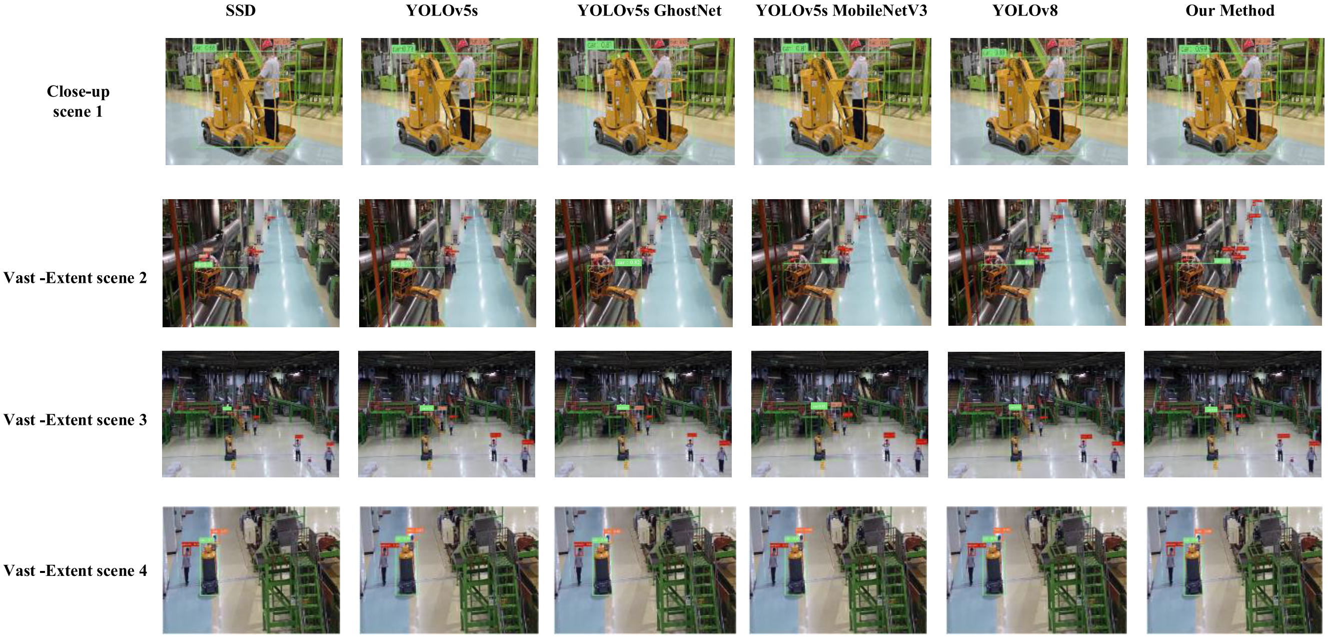


FIGURE 17. The detection comparison in multiple scenarios.

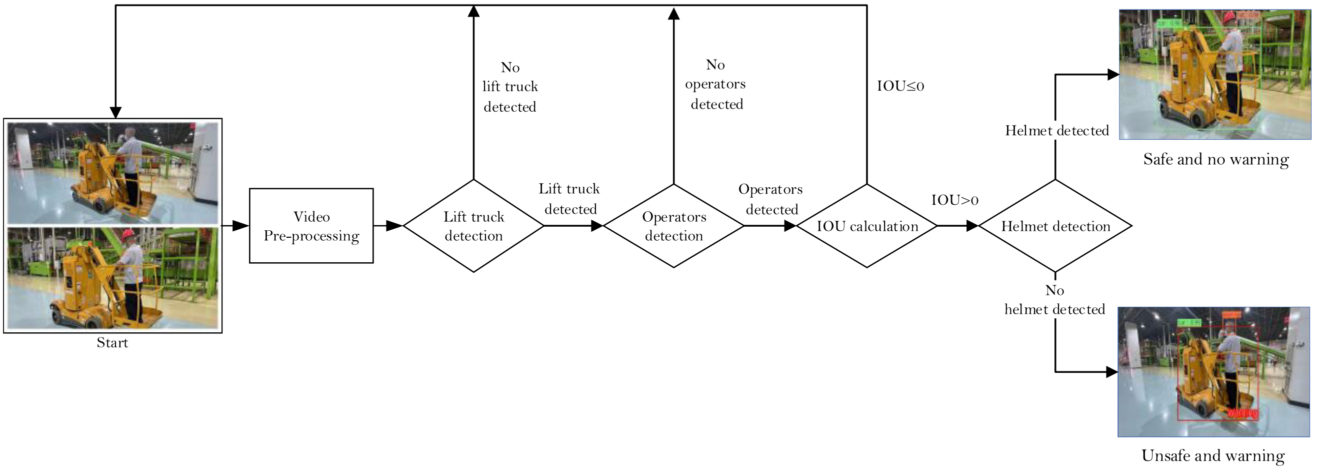


FIGURE 18. The detecting and warning logic.

compared to the other five algorithms. Clearly, our method exhibits superior performance in regard to detecting persons at a distance compared to the other five algorithms.

According to the Vast-Extent scene 3 and Vast-Extent scene 4 in FIGURE 17, our method shows significant improvement in detecting the operator, helmet, and lift truck from the perspective comparison to the other methods.

In conclusion, as shown in FIGURE 17, in different scenarios, our method can achieve the best detection performance in both close range and long-range scenarios. It effectively reduces instances of missed detections and minimizes errors, thereby producing detection outcomes that are more precise and dependable.

F. REAL-TIME HELMETLESS DETECTION SYSTEM

1) REAL-TIME HELMETLESS DETECTION LOGIC

As depicted in FIGURE 18, in the initial phase of the detection system, real-time video undergoes a series of meticulous pre-processing steps. These encompass cleansing, noise filtering, frame rate adjustment, image enhancement, as well as alignment and stabilization of the video. Subsequently, the detection process initiates by identifying the presence of a lift truck. If no lift truck is detected, the current cycle terminates. Otherwise, if a lift truck is present, the system further scrutinizes whether there is an operator on the lift truck. If no operator is detected, the current cycle also terminates. Conversely, if an operator is present, the system proceeds to

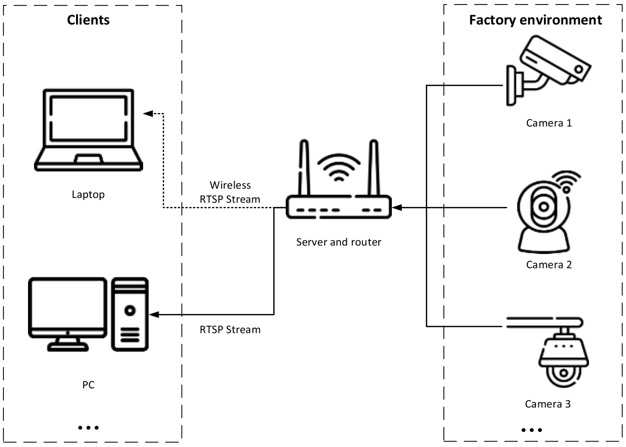


FIGURE 19. The diagram of the real-time detection system.

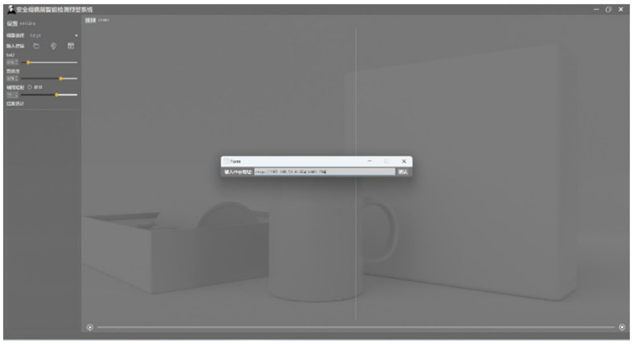
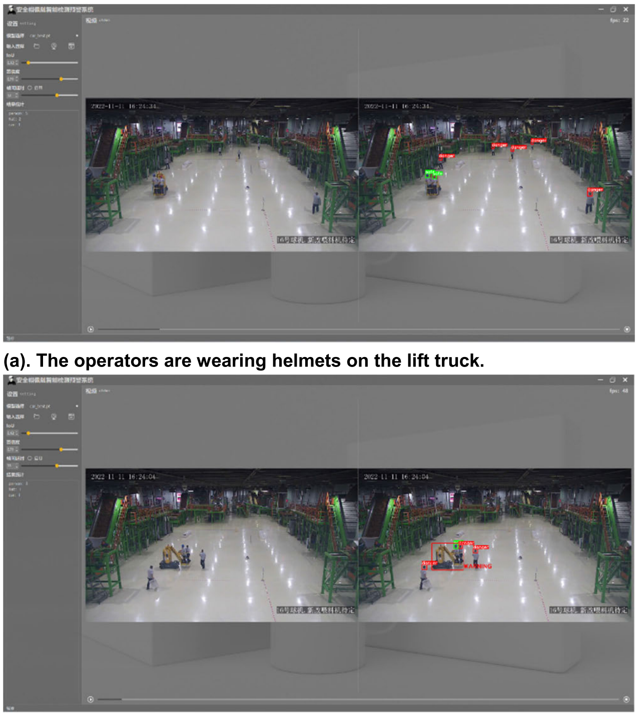


FIGURE 20. Loading RTSP stream in the detection system.



(a). The operators are wearing helmets on the lift truck.

(b). The operators are helmetless on the lift truck.

FIGURE 21. Helmetless detection system.

analyze the bounding boxes of both the lift truck and the operator. The termination condition is triggered when the intersection over union (IoU) of the bounding boxes between the lift truck and the operator is less than or equal to zero, indicating no overlap between them. Conversely, if the IoU is greater than zero, it signifies an overlap between the lift truck and the operator, prompting the system to verify if the operator is wearing a safety helmet. If the operator is wearing a safety helmet, no alarm will be activated. Otherwise, if the operator is not wearing a safety helmet, an alert will be promptly issued, ensuring the safety protocols are adhered to.

2) IMPLEMENTATION OF REAL-TIME DETECTION

The real-time monitoring system is utilized to detect whether the personnel operating the lift truck are wearing safety helmets. It is based on a server architecture and provides real-time viewing and statistical analysis functions. As show in FIGURE 19, the system primarily consists of three components: cameras, streaming media server and clients.

The cameras are responsible for capturing real-time video streams, while the streaming media servers receive, process, store, and transmit the video streams. The clients are used for real-time viewing and conducting statistical analysis. To achieve real-time recognition and viewing effects, the cameras need to support common streaming protocols, such as real-time streaming protocol (RTSP).

RTSP is an application layer control protocol based on the client/server model, which is used to control the transmission of real-time streaming media data. The cameras push the real-time captured video streams to the streaming media servers or clients using RTSP protocol to realize real-time monitoring and viewing functions.

As show in FIGURE 20, upon receiving the video streams, the streaming media servers first handle the reception and parsing of the video streams. Then, the videos are encoded and transcoded to meet the requirements of different transmissions and devices. Subsequently, the server stores and transmits the real-time video streams, establishing connections with clients through wireless RTSP stream or RTSP stream. The server acts as an intermediary between the cameras and clients.

At the server side, our improved YOLOv5s model is deployed for the identification of whether personnel operating the lift truck are wearing safety helmets. This model runs on the server side and changes the input source to a streaming address. By packaging the recognized images into video streams and pushing them to the playback address, clients can view the recognized videos in real-time and manage the display of video streams through interface control functions (such as play, pause, fast forward, etc.). The client also provides statistical analysis functions, such as counting the number of people wearing safety helmets and analyzing violations, aiming to facilitate more effective management and decision-making for supervisory personnel.

As depicted in FIGURE 21, once the camera captures real-time video streams, it transmits the streams to a streaming media server via RTSP. The streaming media server is responsible for receiving and parsing these real-time video streams. At the server end, the proposed helmetless detection model is deployed. This model receives the real-time video stream data transmitted from the camera and processes the video frames.

The recognized images are encapsulated into video streams and pushed to the client in real-time through the interfaces provided by the streaming media server. The client can utilize the corresponding interface to control the functionality and view the real-time video stream of lift truck operators wearing safety helmets. The detection results are illustrated in FIGURE 21, where FIGURE 21(a) represents the display of "safe" when the operator wears a safety helmet, and "danger" is displayed regardless of whether the operator is on the lift truck or not. Notably, when an operator on the lift truck is not wearing a safety helmet, in addition to displaying "danger," a warning prompt of "WARNING" will also be shown, as demonstrated in FIGURE 21(b).

IV. CONCLUSION

To tackle the challenge of detecting small targets concealed by safety helmets worn by operating personnel on lift trucks, the enhanced YOLOv5s improves the effectiveness and precision of detection. The primary contributions of this research paper encompass: (1) Introducing a novel attention mechanism EfficientViT to replace the backbone of the original YOLOv5s, thus enhancing the accuracy in detecting small helmets. (2) Replacing the original YOLOv5s head's C3 module with the C^2F module to improve predictability for challenging events. (3) Applying the Alpha-IoU Loss function to further enhance YOLOv5's ability in detecting small targets. (4) A real-time helmetless detection system was built with a set of well-designed detecting logic.

We have devised three datasets for model training and performance evaluation. The experimental outcomes suggest significant improvements compared to the original YOLO v5s. Specifically, the mAP (0.5) has been enhanced by 6.7%, and the mAP (0.5:0.95) has improved by 6.5%. Furthermore, the model size has been substantially reduced by 51.16%. In addition, our method exhibits advantages over other models mentioned in the text in terms of mAP and model dimensions.

However, the methods proposed in this article do have certain limitations:

(1) Video quality and angles: The performance of the model may be constrained by real-time monitoring video quality and angles, such as pixel blurriness, poor lighting conditions, or camera angle issues. These factors could potentially affect the accuracy and robustness of helmet-less detection.

(2) Pose and occlusion: In real-time monitoring videos, the accuracy of detecting individuals without helmets may be influenced to some extent by variations in the head poses of lift truck drivers and possible occlusions.

In terms of future development prospects, our focus for further investigations will be on enhancing performance under low video quality. This entails accelerating inference speed while simultaneously improving detection accuracy.

REFERENCES

[1] A. Hayat and F. Morgado-Dias, "Deep learning-based automatic safety helmet detection system for construction safety," *Appl. Sci.*, vol. 12, no. 16, p. 8268, 2022.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[3] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6517–6525.

[4] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[5] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[6] Ultralytics. *YOLOv5*. [Online]. Available: https://github.com/ultralytics/YOLOv5

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Computer Vision—ECCV*. 2016.

[8] J. Jeong, H. Park, and N. Kwak, "Enhancement of SSD by concatenating feature maps for object detection," 2017, *arXiv:1705.09587*.

[9] Z. Li and F. Zhou, "FSSD: Feature fusion single shot multibox detector," 2017, *arXiv:1712.00960*.

[10] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD : Deconvolutional single shot detector," 2017, *arXiv:1701.06659*.

[11] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10778–10787.

[12] 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.*, Columbus, OH, USA, Jun. 2014, pp. 580–587.

[13] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[16] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 936–944.

[17] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," 2019, *arXiv:1911.11907*.

[18] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "MobileNetV3: Searching for MobileNetV3,"2019, *arXiv:1905.02244*.

[19] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[20] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.

[21] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with YOLOv8," 2023, *arXiv:2305.09972*.

[22] L. Jun, W. C. Dang, and P. Lihu, "Safety helmet detection based on LO," *Comput. Syst. Appl.*, vol. 28, no. 9, pp. 174–179, Sep. 2019.

[23] K. Han and X. Zeng, "Deep learning-based workers safety helmet wearing detection on construction sites using multi-scale features," *IEEE Access*, vol. 10, pp. 718–729, 2022.

[24] F. Wu, G. Jin, M. Gao, H. E. Zhiwei, and Y. Yang, "Helmet detection based on improved YOLOv3 deep model," in *Proc. IEEE 16th Int. Conf. Netw., Sens. Control (ICNSC)*, May 2019, pp. 363–368.

[25] W. Tai, Z. Wang, W. Li, J. Cheng, and X. Hong, "DAAM-YOLOv5: A helmet detection algorithm combined with dynamic anchor box and attention mechanism," *Electronics*, vol. 12, no. 9, p. 2094, 2094.

[26] A. Benjumea, I. Teeti, F. Cuzzolin, and A. Bradley, "YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles," 2021, *arXiv:2112.11798*.

[27] Z. Jin, P. Qu, C. Sun, M. Luo, Y. Gui, J. Zhang, and H. Liu, "DWCA-YOLOv5: An improve single shot detector for safety helmet detection," *J. Sensors*, vol. 2021, Oct. 2021, Art. no. 4746516.

[28] H. Cai, J. Li, M. Hu, C. Gan, and S. Han, "EfficientViT: Lightweight multi-scale attention for on-device semantic segmentation," 2023, *arXiv:2205.14756*.

[29] Y. Sun, G. Chen, T. Zhou, Y. Zhang, and N. Liu, "Context-aware cross-level fusion network for camouflaged object detection," 2021, *arXiv:2105.12555*.

[30] D. Moore and T. Rid, "Cryptopolitik and the Darknet," *Survival*, vol. 58, no. 1, pp. 7–38, 2016.

[31] J. Wang, Z. Liu, Y. Cheng, Y. Shen, J. Shen, X. Huang, C. Yan, H. Zhang, X. Li, and S. Yan, "CSPDarkNet: A light modular darknet for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, 2020.

[32] B. Koonce and B. Koonce, "EfficientNe convolutional neural networks with swift for tensorflow: Image recognition and dataset categorization," Tech. Rep., 2021.

[33] 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," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 658–666.

[34] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, 2006, pp. 850–855.

[35] J. He, S. Erfani, X. Ma, J. Bailey, Y. Chi, and X.-S. Hua, "α-IoU: A family of power intersection over union losses for bounding box regression," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 20230–20242.

[36] F. Geser, G. K. Wenning, K. Seppi, M. Stampfer-Kountchev, C. Scherfler, M. Sawires, C. Frick, J. P. Ndayisaba, H. Ulmer, M. T. Pellecchia, and P. Barone, "Progression of multiple system atrophy (MSA): A prospective natural history study by the European MSA study group," *Movement Disorders*, vol. 21, no. 2, pp. 179–186, 2006.

[37] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

YUNCHANG ZHENG received the B.S. degree in electronic science and technology and the M.S. degree in signal and information processing from the University of Electronic Science and Technology of China, in 2013 and 2016, respectively. Since 2017, he has been a Lecturer with the College of Electrical Engineering, Hebei University of Architecture, Zhangjiakou, China. His research interests include image processing, deep learning, and artificial intelligence.

MENGFAN WANG is currently pursuing the degree in electrical engineering and automation with the Hebei University of Architecture, Zhangjiakou, China. His research interests include machine learning and artificial intelligence.

YICHAO LIU has been an Assistant Engineer with China Tobacco Zhangjiakou Cigarette Factory Company Ltd., since 2016. His research interests include security management and image processing.

CUNYANG LI is currently pursuing the degree in measurement and control technology and instruments with the Hebei University of Architecture, Zhangjiakou, China. His research interests include deep learning and artificial intelligence.

QING CHANG received the B.S. degree in measurement and control technology and instruments from Yanshan University, in 2004, and the M.S. degree in computer technology from the Hebei University of Technology, in 2011. From 2010 to 2018, she was with the Department of Scientific Research Management. Since 2004, she has been a Teacher with the Department of Electrical Engineering, Hebei University of Architecture, where she is currently an Associate Professor. Her research interests include virtual instruments, signal processing, and industrial process control.