IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 14 July 2023, accepted 30 July 2023, date of publication 7 August 2023, date of current version 17 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3302909

RESEARCH ARTICLE

GBForkDet: A Lightweight Object Detector for Forklift Safety Driving

LINHUA YE[1,2] AND SONGHANG CHEN[3], (Senior Member, IEEE)

[1]Quanzhou Institute of Equipment Manufacturing, Haixi Institutes, Chinese Academy of Sciences, Quanzhou 362200, China
[2]College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350117, China
[3]Fujian Institute of Research on the Structure of Matter, Chinese Academy of Sciences, Quanzhou, Fujian 350108, China

Corresponding author: Songhang Chen (songhang.chen@fjirsm.ac.cn)

This work was supported in part by the Fujian Science and Technology Innovation Laboratory for Optoelectronic Information of China under Grant 2021ZZ121, and in part by the Central Guidance on Local Science and Technology Development Fund of Fujian Province under Grant 2021L3047.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Quanzhou Institute of Equipment Manufacturing Haixi Institutes.

ABSTRACT The importance of object detection in intelligent logistics applications is increasingly recognized. However, current detector models suffer from challenges such as high computational cost and low detection accuracy, which limit their deployment on edge devices with limited computational power in logistics scenarios. To address these issues, this paper proposes a novel lightweight detector model (GBForkDet) based on YOLOv8 for forklift safety driving. Firstly, the Ghost module is integrated into YOLOv8 to optimize the Backbone feature extraction process, reducing the computational cost of the model. Then, a Bi-directional Omni-Dimensional Dynamic Neck (BiODNeck) is designed to fuse feature information in complex logistics scenarios. GBForkDet significantly improves the capture of contextual logistics background information by reconstructing the Neck of YOLOv8 with BiODNeck. This is attributed to cross-layer weighted feature fusion and a complementary focus on learning convolutional kernels in any convolutional layer along all four dimensions of the kernel space. Furthermore, the introduction of the Normalized Wasserstein distance (NWD) as an enhanced loss function improves the detection of small distant objects in logistics scenarios. Experimental results show that GBForkDet achieves a mAP of 92.7% and 95.3% on the established Forklift-3k and KITTI datasets while reducing the model parameters by 17.9% and the computational cost by 22.5% compared to the baseline YOLOv8s model. Under the Jetson Nano edge platform and 640×640 input size, the GBForkDet model achieves a remarkable inference time of 108.2 ms using TensorRT acceleration.

INDEX TERMS Object detection, intelligent logistics, YOLOv8, feature fusion.

I. INTRODUCTION

With the rapid development and increasing complexity of the logistics field, forklifts play a vital role in warehousing and logistics scenarios. However, forklifts are exposed to numerous safety hazards that risk personal safety during operations. The leading causes of these hazards include limited visibility of drivers due to excessive stacking of goods, rear blind spots caused by the height of the forklift body, and potential dangers and violations due to driver fatigue, lack of concentration, and unsafe operations. Given these safety concerns, there is an urgent need for safety warnings in forklift operations. Traditional safety warning methods are based on laser sensors and ultrasonic sensors. Laser sensors offer the advantages of high accuracy and a longer range for distance measurement, enabling accurate detection and measurement of obstacle positions. Laser sensors are typically installed at the front or sides of the forklift and provide critical inputs

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

for warning decisions. However, laser sensors have limited adaptability in complex environments, such as encountering transparent or highly reflective objects, which can interfere with or diminish their detection capabilities. Additionally, laser sensors have narrow horizontal and vertical fields of view, resulting in the omission or misjudgment of certain obstacles. Particularly, blind spots exist at the rear and sides of the forklift, which can increase safety risks. Ultrasonic sensors represent another commonly used traditional safety warning method. They utilize the echo of sound waves to measure the distance to objects, allowing real-time detection of obstacles in front of the forklift. Ultrasonic sensors are suitable for scenarios involving low-speed movement and close-range obstacle avoidance. However, they have relatively limited accuracy and range for distance measurement. Ultrasonic sensors fail to meet the requirements for scenarios requiring higher precision and long-distance measurement.

As deep learning has rapidly developed, machine vision-based safety alerts have emerged as an essential application of object detection models in complex logistics scenarios. Adequate safety warnings can be achieved by equipping forklifts with low-cost depth cameras and utilizing RGB-D images to predict target positions and distances. Machine vision-based safety warning methods can effectively overcome the limitations of traditional approaches and provide assistance for forklift operations. Object detection is a critical task in the field of deep learning and is primarily categorized into two methods: the one-stage approach and the two-stage approach. The two-stage approach divides the target detection task into two stages: proposal generation and target classification. Firstly, a series of candidate boxes containing potential target objects is generated using the region proposal method. Subsequently, these candidate boxes are classified and accurately localized using a classifier. Classical two-stage methods include R-CNN (Region-based Convolutional Neural Networks) [1], Fast R-CNN [2], and Faster R-CNN [3]. Although these methods achieve high detection accuracy, their computational speed is relatively slow, requiring multiple computations. On the other hand, the one-stage approach directly performs object classification and bounding box regression without utilizing region proposal methods. Representative one-stage methods include the YOLO series [4], [5], [6], [7], SSD (Single Shot MultiBox Detector) [8], and RetinaNet [9]. These methods can directly generate target category scores and bounding box coordinates, thus exhibiting better real-time performance. However, detectors based on the one-stage approach generally exhibit inferior detection accuracy compared to those based on the two-stage approach.

Despite the excellent performance of one-stage object detection methods on devices with ample computing resources, they face high computational costs and low detection accuracy on the resource-constrained edge and mobile devices. Particularly, real-time capability is crucial in safety driving scenarios within complex logistics. Therefore, achieving a balance between detection accuracy and speed has become a significant challenge in intelligent logistics. Another challenge is the lack of suitable logistics scene datasets, which presents difficulties in research work. And this is due to the complexity and variability of logistics environments, involving different types of goods, stacking methods, lighting conditions, and other factors, making data collection challenging. Furthermore, introducing a Feature Pyramid Network (FPN) [10] has been widely applied in computer vision tasks, effectively addressing the challenge of multi-scale feature fusion. However, the FPN structure fails to extract semantic information from targets in complex logistics backgrounds effectively. To address this issue, Bi-Directional Feature Pyramid Network (BiFPN) [11] proposes a simple and efficient weighted bidirectional feature pyramid network. This network incorporates learnable weights to determine the importance of different input feature layers, enabling better adaptation to various complexities of logistics scenarios.

This paper proposes GBForkDet to address the challenges of high computational costs and low detection accuracy in machine vision-based safety prewarning on the edge and mobile devices. The main contributions of this paper are as follows:

- Introducing the Ghost module to optimize the calculation process in Backbone, replacing the C2f module with the C3Ghost module.
- Designing BiODNeck to enhance the model's ability to capture feature information from complex logistics backgrounds by reconstructing the Neck of YOLOv8.
- Optimizing the loss function using the normalized Gaussian Wasserstein distance and mitigating GBForkDet's sensitivity to small targets during detection.
- A series of experiments are conducted on the established Forklift-3k dataset to validate the effectiveness of the GBForkDet model. Furthermore, an inference speed of 108.2 ms is achieved using TensorRT acceleration on the Jetson Nano platform.

II. RELATED WORKS

A. CHALLENGES IN OBJECT DETECTION

Over the past few years, deep learning has been widely applied in various fields, including autonomous driving [12], [13], [14], [15]. Deep learning-based object detection methods have outperformed traditional approaches [16], [17], [18]. Deformable convolution [19] has been introduced in autonomous driving to address real-time requirements. Additionally, a fog driving detection network has been designed to tackle the issue of foggy weather conditions in autonomous driving scenarios [20]. While machine vision-based object detection has found extensive application in autonomous driving, there are notable differences and challenges compared to forklift driving in logistics scenarios. Forklift driving in logistics settings involves challenges such as narrow roads, cluttered goods, and relatively low speeds, which pose specific challenges to researchers in this domain.

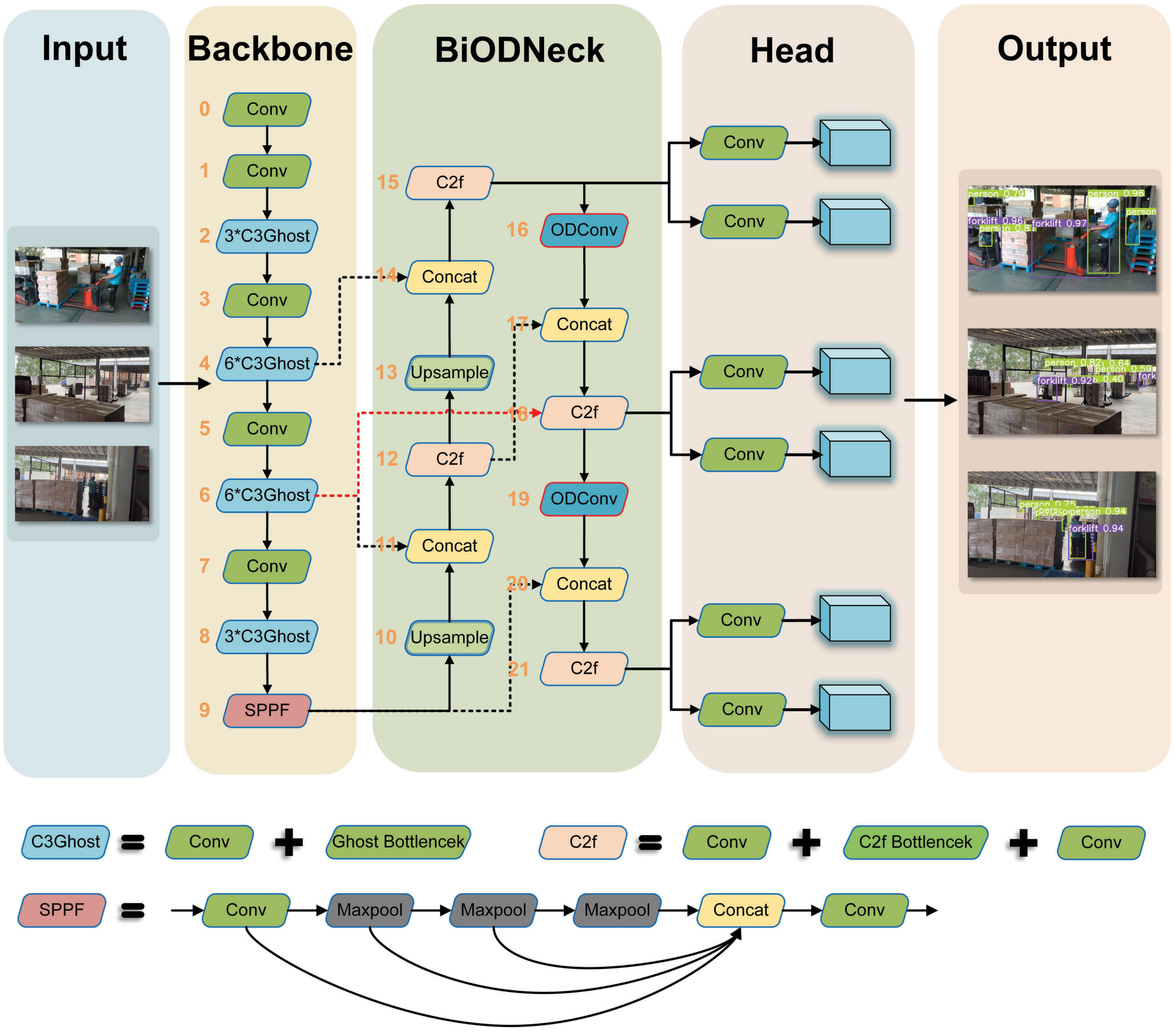


FIGURE 1. Overall architecture of the GBForkDet model. a) The Ghost module optimizes the Backbone. b) Each head uses a decoupled head. c) The number of each layer of the module is marked with a number on the left side.

B. YOLOv8

The YOLO series has consistently been the most popular detection framework in industrial applications, as it strikes a good balance between speed and accuracy. YOLOv8 [21] is one of the most advanced one-stage object detection models currently available. It inherits numerous advantages from YOLOv5 and consists of four components: Input, Backbone, Neck, and Head. YOLOv8 encompasses five models, namely YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. The YOLOv8s model offers the best trade-off between detection performance and model size. The proposed GBForkDet model in this study is based on YOLOv8s. The Backbone network extracts features from the input images, while the neck component fuses feature information from the Backbone. The prediction head uses the feature information from the Neck to make predictions. The YOLO series models combine classification and regression tasks for joint optimization. However, this approach leads to mutual influence between classification and regression errors, making it challenging to obtain optimal detection results. To address this issue, YOLOv8 adopts a new decoupled head structure, optimizing the losses of different tasks separately. Furthermore, YOLOv8 introduces an Anchor-Free method to replace the traditional Anchor-Based approach. Anchor-Based object detection models require predefined anchor boxes to model objects of different scales and aspect ratios. In contrast, the Anchor-Free method employed by YOLOv8 is better suited to handle objects of varying scales and aspect ratios while reducing computational costs.

C. ADVANCES IN LIGHTWEIGHT MODELS

With the popularity of edge devices and mobile devices, lightweight models have become increasingly important in the field of machine vision. The current major challenge is to design a lightweight neural architecture that can achieve both fast inference and high performance [22], [23], [24] to meet the requirements of detection tasks on these devices. SqueezeNet [25] is an early lightweight model that reduces the number of parameters by using 1 × 1 convolutional kernels and channel compression techniques. It provides comparable accuracy with fewer parameters, making it suitable for environments with limited computational resources. MobileNetV1 [26] is a classic lightweight model that uses depthwise separable convolutions and other operations to reduce the number of parameters and computational complexity. MobileNetV1 significantly reduces the model's size while maintaining relatively high accuracy by introducing techniques such as depthwise separable and pointwise convolutions. MobileNetV2 [27] is an improvement over MobileNetV1, further introducing an inverted residual structure and linear bottlenecks to enhance the model's accuracy and computational efficiency. The inverted residual structure effectively increases the model's non-linearity, while the linear bottlenecks help reduce the computational cost. ShuffleNetV1 [28] is another lightweight model that reduces the number of parameters and computational complexity by introducing channel shuffling operations. Channel shuffling effectively reduces the inter-channel correlation in the model, resulting in lower computational complexity and improved efficiency. Building upon ShuffleNetV1, ShuffleNetV2 [29] introduces pointwise group convolutions and channel shuffling operations, further improving the model's accuracy and computational efficiency. Pointwise group convolutions and channel shuffling operations enhance the model's non-linearity and reduce computational complexity. To further enhance the performance of lightweight models, researchers have proposed GhostNet [30]. GhostNet achieves lightweight model representation with lower computational costs and higher accuracy by introducing Ghost modules and optimizing network structures. The Ghost module reduces the number of parameters and computational complexity by adding ghost channels before convolutional layers, providing efficient feature representation through feature reuse and information communication. The work on these lightweight models provides inspiration for designing an efficient detection model for logistics scenarios in this paper.

III. PROPOSED METHODS

A. OVERALL ARCHITECTURE

Fig. 1 illustrates the overall architecture of the GBForkDet model. The architecture comprises four main modules: Input Module, Backbone, BiODNeck, and Head. In the Input Module, the image size is set to 640×640, and sequential online data augmentation operations are performed, including HSV colour space enhancement, Mosaic, and geometric flipping. The Backbone module replaces the C2f module with the C3Ghost module, composed of Ghost Bottlenecks. The Neck module is restructured with BiODNeck to improve feature fusion in the model. The black dotted lines represent the original cross-layer connection in baseline YOLOv8, while the red dotted line illustrates the newly introduced cross-layer connection in GBForkDet. During the prediction stage, the decoupling heads separate the feature information from the Neck module into classification and regression tasks, allowing different predictions for each. The training loss of GBForkDet is based on YOLOv8 and optimized using the Normalized Gaussian Wasserstein distance (NWD). The loss function of GBForkDet consists of three components: classification loss, box regression loss, and distribution focal loss. The classification loss is utilized to evaluate the model's ability to accurately recognize objects in the image. The box regression loss measures the discrepancy between the predicted bounding box coordinates and the ground truth bounding box annotations. The distribution focal loss employs cross-entropy to expedite the network's attention on the distribution of neighbouring regions around the target location.

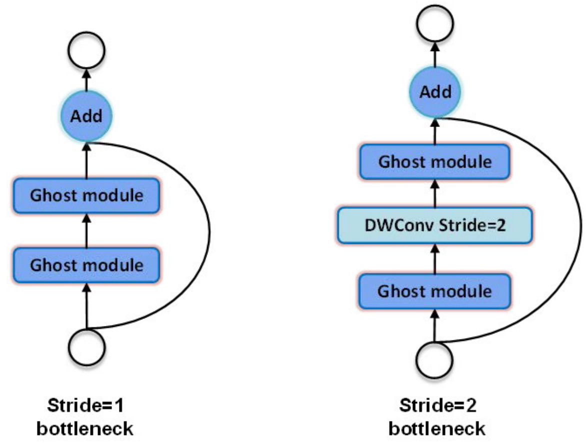


FIGURE 2. **Ghost Bottleneck.**

B. GHOST MODULE

In the Backbone network, efficient network structures such as VGG [31], DenseNet [32], and CSPDarknet53 [33] have been widely used. However, these networks incur significant computational costs. In contrast, GhostNet effectively reduces redundancy in feature information by adopting economically efficient operations. Fig. 2 illustrates the Ghost Bottleneck structure, which consists of two sets of Ghost modules with multiple Conv layers and shortcut connections. The Backbone network of the GBForkDet model replaces the C2f module in YOLOv8 with the C3Ghost module composed of Ghost Bottlenecks. The Ghost Module divides the input feature maps into main and auxiliary paths, significantly reducing the parameter count. The Ghost Bottleneck combines the Ghost Module with traditional residual connections, further reducing the number of parameters and computational complexity. Additionally, GhostNet employs lightweight bottleneck structures, including reducing the number of convolutional layers and channels, and using smaller kernel sizes. These strategies allow GhostNet to significantly reduce the parameters while maintaining model performance, making it highly suitable for resource-constrained devices and scenarios.

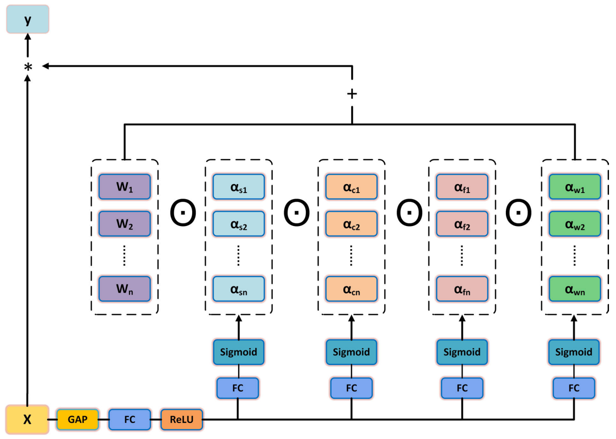


FIGURE 3. The computation process of ODConv.

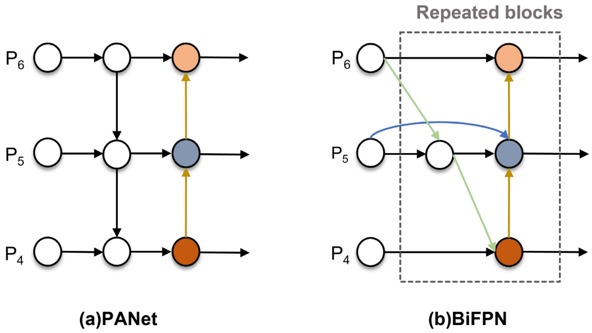


FIGURE 4. (a) is the structure of PANet; (b) is the structure of BiFPN.

C. FEATURE FUSION

1) OMNI-DIMENSIONAL DYNAMIC CONVOLUTION

The limitations of traditional static convolution in terms of receptive field size, multi-scale adaptability, parameter redundancy, adaptability, and local perception capability have constrained models' expressive power and adaptability. Therefore, researchers have proposed methods such as dynamic convolution [34], [35] to overcome the limitations of static convolution and enhance model performance. Dynamic convolution determines each convolutional kernel's weight based on the convolutional layer's input and combines it with an attention module to obtain adaptive dynamic convolutional kernels. The output y of dynamic convolution is as follows:

$$y = x * (\alpha_1\ W_1 + \alpha_2\ W_2 + \cdots\cdots + \alpha_n\ W_n\ ) \qquad (1)$$

where x and y denote the input features and output features respectively. α_i (i = 1,2,...,n) is the attention vector, and N is the number of convolutional kernels. Each convolutional kernel W_i (i = 1,2,...,n) has the same size as the standard convolutional kernel.

Omni-Dimensional Dynamic Convolution (ODConv) [36] is a more elegant dynamic convolutional structure. It utilizes a multi-dimensional attention mechanism and parallel strategies to learn complementary attention for convolutional kernels across all four dimensions in the kernel space. These four different attention focuses are the input channel number of the convolutional kernel's receptive field, the kernel's output channel number, and the number of kernels. These four attention focuses are mutually complementary and are multiplied with the convolutional kernels in the order of position, channel, filter, and kernel, enabling convolutional operations to capture context information different from all spatial positions, input channels, filters, and kernels of the input x, significantly enhancing contextual information capture. The output of ODConv is represented as follows:

$$\begin{aligned} y = x * \big(&\alpha_{w1} \odot \alpha_{f1} \odot \alpha_{c1} \odot \alpha_{s1} \odot W_1 + \cdots\cdots \\ &+ \alpha_{wn} \odot \alpha_{fn} \odot \alpha_{cn} \odot \alpha_{sn} \odot W_n\big) \end{aligned} \qquad (2)$$

where y and x denote output features and input features respectively. α_{wi}, α_{fi}, α_{ci}, α_{si} (i = 1,2,...,n) represent four different attention mechanisms: the scalar, spatial dimension, input channel dimension, and output channel dimension of the convolutional kernel W_i (i = 1,2,...,n). The symbol $\odot$ denotes multiplication along different dimensions in the kernel space. Fig. 3 illustrates the computation process of ODConv.

2) BI-DIRECTIONAL FEATURE PYRAMID NETWORK

YOLOv8 incorporates the Path Aggregation Network (PANet) [11] to efficiently integrate features of different scales, as shown in Fig. 4 (a). While PANet proves effective in fusing different feature layers, its underlying mechanism relies on simply adding these features. However, the training process produces features of different scales due to the different sizes of detected objects in different images. The simple summation of these feature maps within PANet leads to an unequal contribution of features of different scales to the fused representation. To address this challenge, the Bi-directional Feature Pyramid Network (BiFPN) extends PANet by introducing a bidirectional flow of information from higher-level to lower-level features, as shown in Fig. 4 (b).

BiFPN introduces a more sophisticated bi-directional feature fusion approach, along with cross-scale connections and weighted information fusion, to achieve comprehensive feature integration with relatively fewer additional parameters. In addition to the bottom-up feature propagation path, BiFPN incorporates an additional edge connecting the output and input, facilitating the flow and fusion of feature information across scales. BiFPN achieves cross-scale feature fusion through the following fusion formula:

$$O = \sum_i \frac{w_i}{\epsilon + \sum_j w_j} * I_i \qquad (3)$$

3) BI-DIRECTIONAL OMNI-DIMENSIONAL DYNAMIC NECK

Fig. 5 illustrates the structures of the Neck of YOLOv8 and the proposed BiODNeck in this paper. Inspired by BiFPN, we have restructured the Neck of YOLOv8 to maintain efficient feature extraction capabilities in lightweight models. In this study, ODConv is integrated into the proposed

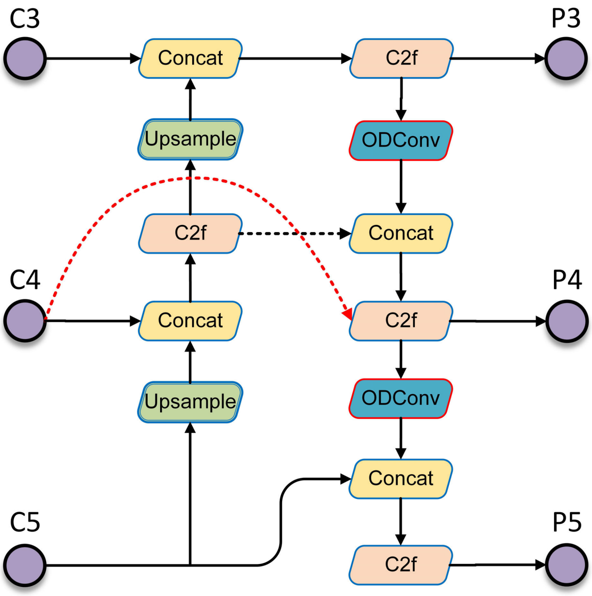


FIGURE 5. The structure of the proposed BiODNeck.

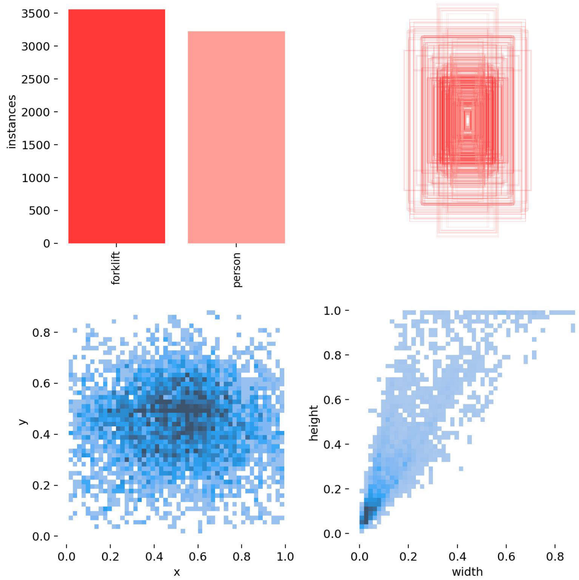


FIGURE 6. Visualization of the information in the Forklift-3k dataset.

BiODNeck in a plug-and-play manner, allowing adaptive tuning of convolutional kernels to extract complex background features from logistics images selectively. Specifically, the design of BiODNeck includes the following aspects: Firstly, the concatenation operation is replaced with element-wise addition to fuse feature information using a weighted fusion method. Secondly, cross-layer connections are introduced to enhance shallow semantic information at the Head. In addition, ODConv restructures key convolutional modules in the original Neck and significantly improves the model's feature extraction and learning capabilities, substantially improving logistics target recognition accuracy.

D. NORMALIZED WASSERSTEIN DISTANCE

Due to the low contrast and blurry characteristics of distant small objects, combined with their limited pixel coverage in images, they are often overlooked, posing a challenging problem in logistics scenarios. Traditional methods such as CIoU, DIoU [37], and EIoU [38], which are based on Intersection over Union (IoU) [39], are sensitive to positional deviations of tiny objects. This paper optimizes the GBForkDet model by introducing Wasserstein distance loss to improve sensitivity to small targets. Specifically, we incorporate the Normalized Wasserstein distance (NWD) [40] into the existing IoU loss function. This loss function measures the distance between the distributions of real and generated samples, resulting in a novel loss formulation. By incorporating the NWD, the training process considers both the coordinate offsets and the target size information of the bounding boxes, thus placing more emphasis on the detection of tiny objects. The NWD has the advantage of being insensitive to objects of different scales, making it suitable for measuring the similarity between tiny objects. Its integration enables the GBForkDet model to address better the detection challenges posed by distant small targets in logistics scenarios. By optimizing the loss function, we effectively enhance the model's sensitivity to small objects, effectively addressing the challenges present in logistics scenarios.

For two two-dimensional Gaussian distributions $\mu_1 = N(m_1, \Sigma_1)$ and $\mu_2 = N(m_2, \Sigma_2)$, their second-order Wasserstein distances can be defined as follows:

$$W_2^2(\mu_1, \mu_2) = \|m_1 - m_2\|_2^2 + Tr\left(\sum_1 + \sum_2 - 2\left(\sum_2^{1/2}\sum_1\sum_2^{1/2}\right)^{1/2}\right) \tag{4}$$

where $\sum_1$ and $\sum_2$ denote the covariance matrices of Gaussian distributions μ_1 and μ_2, respectively. Simplifying the expression leads to:

$$W_2^2(\mu_1, \mu_2) = \|m_1 - m_2\|_2^2 + \left\|\sum_1^{1/2} - \sum_2^{1/2}\right\|_F^2 \tag{5}$$

For the Gaussian distributions N_a and N_b which are obtained from the bounding boxes $A = (cx_a, cy_a, w_a, h_a)$ and $A = (cx_b, cy_b, w_b, h_b)$, the equation mentioned above can be further simplified as follows:

$$W_2^2(N_a, N_b) = \left\|\left(\left[cx_a, cy_a, \frac{w_a}{2}, \frac{h_a}{2}\right]^T, \left[cx_a, cy_a, \frac{w_b}{2}, \frac{h_b}{2}\right]^T\right)\right\|_2^2 \tag{6}$$

where $W_2^2(N_a, N_b)$ serves as a distance measure, it cannot be used directly as a similarity measure due to the requirement

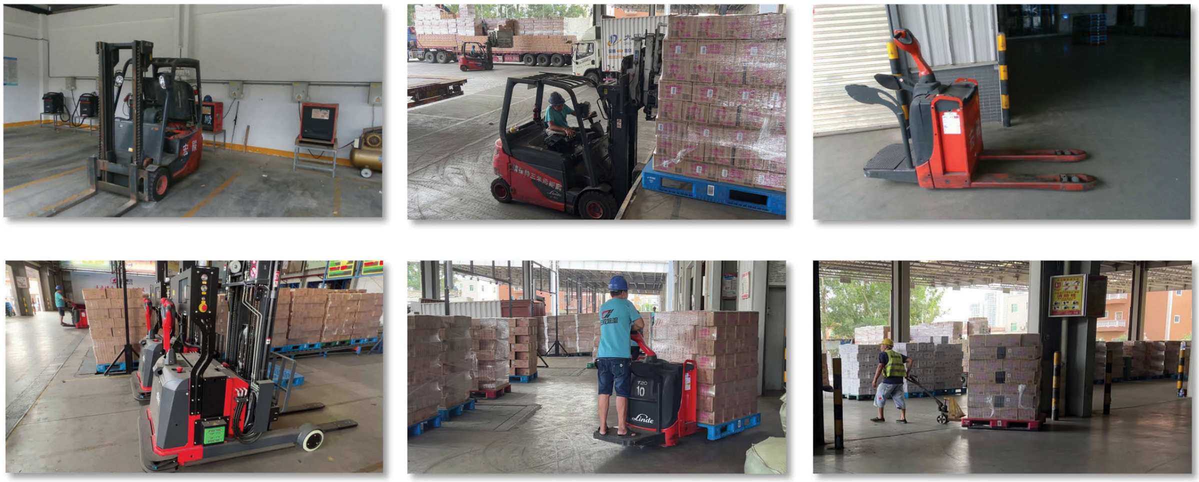


FIGURE 7. The logistics image samples in the Forklift-3k dataset.

FIGURE 8. The image samples in the KITTI dataset.

of IoU to yield values between 0 and 1. To overcome this limitation, we apply exponential normalization to $W_2^2\left(N_a, N_b\right)$, resulting in the derivation of a novel metric termed the normalized Wasserstein distance. The computation is outlined as follows:

$$NWD\left(N_a, N_b\right) = \exp\left(-\frac{W_2^2\left(N_a, N_b\right)}{C}\right) \tag{7}$$

where C is a dataset-dependent constant that ensures stability within a certain range. In our experiments, we set C to the average absolute size of the targets in the dataset to optimize performance. The NWD metric is designed as loss function by:

$$L_{NWD} = 1 - NWD\left(N_a, N_b\right) \tag{8}$$

IV. EXPERIMENTS

A. DATASETS

1) FORKLIFT-3k DATASET

In the field of logistics object detection, there is a relative scarcity of publicly available dataset. Therefore, we conducted data collection and construction to create a logistics object detection dataset named Forklift-3k, aimed at supporting research on safe driving for forklifts. This dataset primarily consists of two target categories: forklifts (4,572 samples) and persons (4,190 samples), totalling 3,342 images. Fig. 6 shows a detailed visualization of the information in the Forklift-3k dataset, including the distribution of target classes, annotation sizes, centre point coordinates and aspect ratios of the bounding boxes. The images were captured using an Intel RealSense D435i depth camera and various mobile devices, with a uniform size of 1920×1080 pixels. Fig. 7 showcases several common types of forklifts in logistics scenarios.

2) KITTI DATASET

The KITTI [41] dataset is currently one of the largest internationally recognized benchmark datasets in the field of autonomous driving. It comprises various challenging scenarios, including small distant targets and uneven lighting, which are similar to the complexities encountered in complex logistics environments. Fig. 8 shows some sample images from the KITTI dataset. This paper uses the KITTI dataset to evaluate the robustness of the GBForkDet model. To better match real-world engineering scenarios, the original classes of Car, Van, Truck, and Tram are merged into Car, while Pedestrian and Person sitting are merged into Person. In addition, the classes Misc and DontCare are removed from the dataset. As a result, the dataset consists of a total of 7,481 images, each annotated with one of three different categories: Car (33,261 samples), Person (4,709 samples), and Cyclist (1,627 samples).

B. EVALUATION INDICATORS

This paper employs mean Average Precision (mAP@0.5) as the primary evaluation metric to assess the model's detection performance. A higher mAP value indicates better detection performance. mAP@0.5 represents each category's average precision (AP) when the Intersection over Union (IOU) between the predicted bounding boxes and ground truth bounding boxes exceeds 0.5. For detection tasks, it is customary to calculate True Positive (TP), False Positive (FP), and False Negative (FN) for each category. TP represents correctly classified positive samples, while FP (FN) represents misclassified positive (negative) samples. Precision refers to the ratio of correctly identified images, and its formula is as follows:

$$P = \frac{TP}{TP + FP} \tag{9}$$

Recall represents the ratio of correctly identified positive samples and is calculated as follows:

$$R = \frac{TP}{TP + FN} \tag{10}$$

Average Precision (AP) signifies the area under the Precision-Recall curve. mAP represents the average AP for each category in a multi-class detector, and its formula is given by:

$$mAP = \frac{1}{n}\sum_{i=1}^{n}\int_{0}^{1} P(R)\, dx \tag{11}$$

C. EXPERIMENTAL RESULTS AND ANALYSIS

The experiments in this study are conducted on a Windows 11 operating system, utilizing an i9-12900k CPU and a NVIDIA GEFORCE RTX 4090 (24GB VRAM) processor with 64GB of RAM. Several experimental hyperparameters were set as follows: the number of training epochs is set to 200, and the Mosaic data augmentation operation is closed in the last 10 epochs. The batch size was set to 80, and the optimizer used was SGD with a learning rate of 0.0005, momentum of 0.937, and weight decay of 0.0005. The training of the dataset is performed in a programming environment based on PyTorch 2.0, Python 3.9, and CUDA 11.8. To ensure consistency, all models are trained and tested using the NVIDIA GEFORCE RTX 4090 GPU. To make a fair comparison of the model's performance, all models are trained from scratch under the same experimental conditions.

Fig. 9 illustrates the training process of the baseline YOLOv8s and GBForkDet models on the Forklift-3k dataset. It can be observed that the loss curve on the training set exhibits abnormal fluctuations in the last ten epochs, which is due to the disabling of online data augmentation in the final ten training rounds. From the training process, it is evident that GBForkDet achieves faster convergence compared to YOLOv8s. Furthermore, the training results reveal that GBForkDet exhibits lower box loss, indicating its greater focus on object localization and surrounding contextual features.

To provide a visual representation of the roles of C3Ghost and ODConv in the GBForkDet model, this paper selects representative samples from the Forklift-3k dataset and visualizes the feature maps during the detection process. Fig. 10 and Fig. 11 showcase the visualizations obtained. C3Ghost is responsible for extracting low-level feature semantic information, while ODConv focuses on extracting high-level feature semantic information. These visualizations effectively demonstrate the complementary nature of C3Ghost and ODConv in capturing different levels of semantic information, which contributes to the improved detection performance of the GBForkDet model. The visualizations not only provide insights into the inner workings of the model but also validate the effectiveness of the proposed feature extraction modules in enhancing object detection capabilities.

Fig. 12 shows the response of features to different classes of targets in the same input image in YOLOv8s and GBForkDet. GBForkDet has a more accurate scope of attention, enabling it to detect targets better. This result clearly shows that GBForkDet effectively captures key feature information of targets in complex logistics scenarios.

D. EXPERIMENTAL RESULTS AND ANALYSIS

Table 1 and Table 2 present detailed performance results of different models on the Forklift-3k and KITTI datasets. The evaluated models include well-known detection models such as YOLOv4, YOLOv5-Lite, and YOLOv7-tiny. GBForkDet achieves higher detection accuracy in the person category while performing on par with the baseline model YOLOv8s in the Forklift category. As a lightweight model, GBForkDet exhibits significant advantages. Compared to YOLOv8s, GBForkDet reduces model parameters and FLOPs by approximately 17.9% and 22.5%, respectively. In addition, GBForkDet achieves a remarkable detection speed, with an inference time of just 7.4ms on the RTX4090 GPU.

This paper presents a comparative analysis of the visual results among the classical models YOLOv5-Lite, YOLOv8s, and the proposed GBForkDet. Fig. 13 showcases the visualization of the detection results of these models in logistics scenes, shedding light on the challenging detection issues encountered in such scenarios, including complex backgrounds, uneven lighting, extensive occlusions, and distant small objects. The labelled results are shown in the original images in the figure. YOLOv5-Lite exhibited subpar performance with missed detection issues, particularly for small objects. YOLOv8s showed lower detection accuracy for distant small objects due to a lack of robust feature extraction capabilities in complex logistic backgrounds. In contrast, the GBForkDet model leveraged the introduction

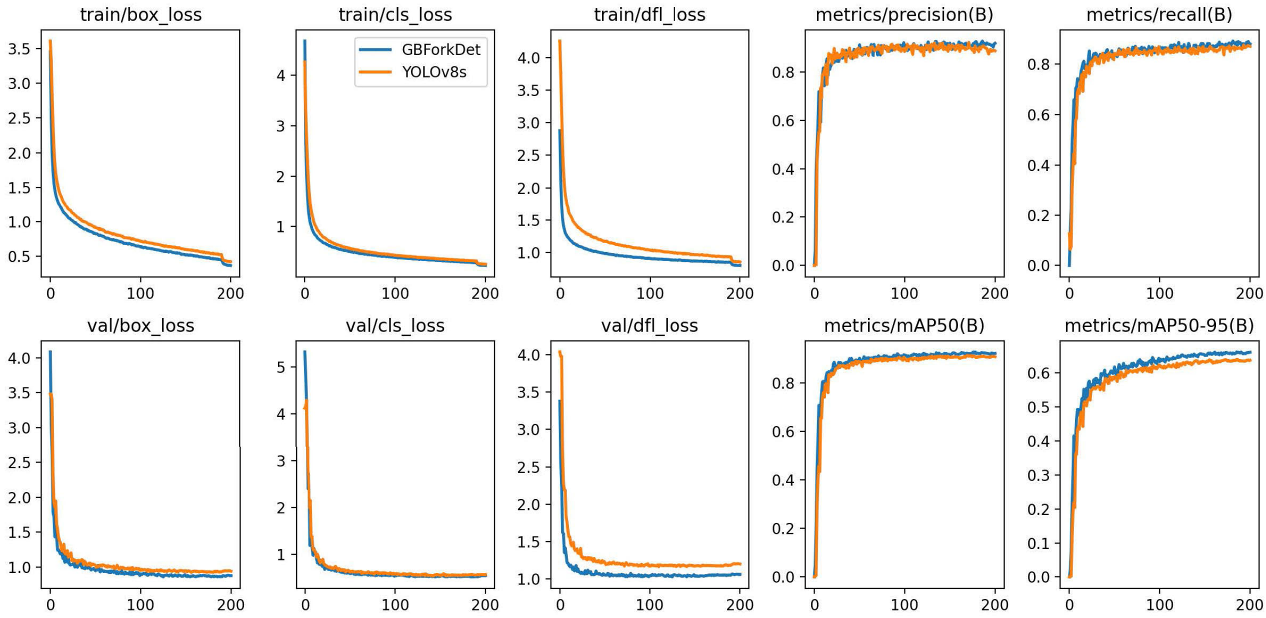


FIGURE 9. Training process diagram for YOLOv8s and GBForkDet (our).

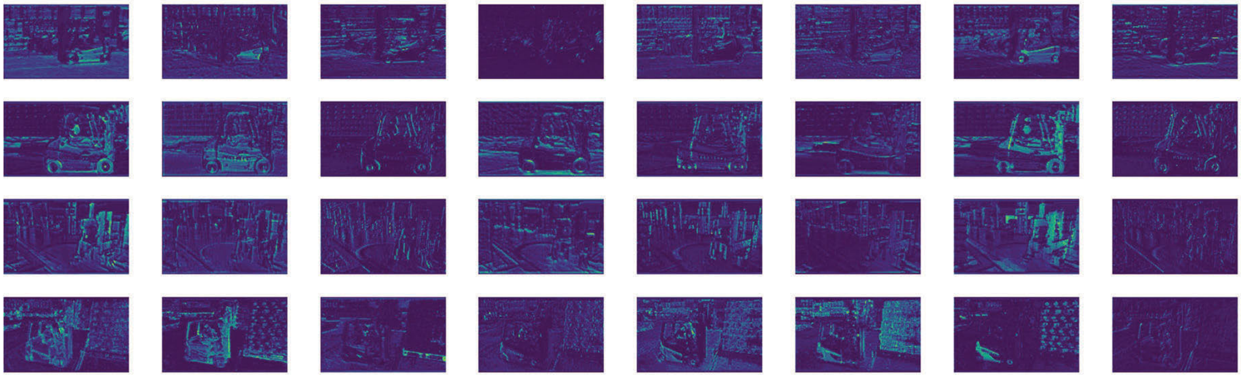


FIGURE 10. Visualization of the feature map of C3Ghost module.

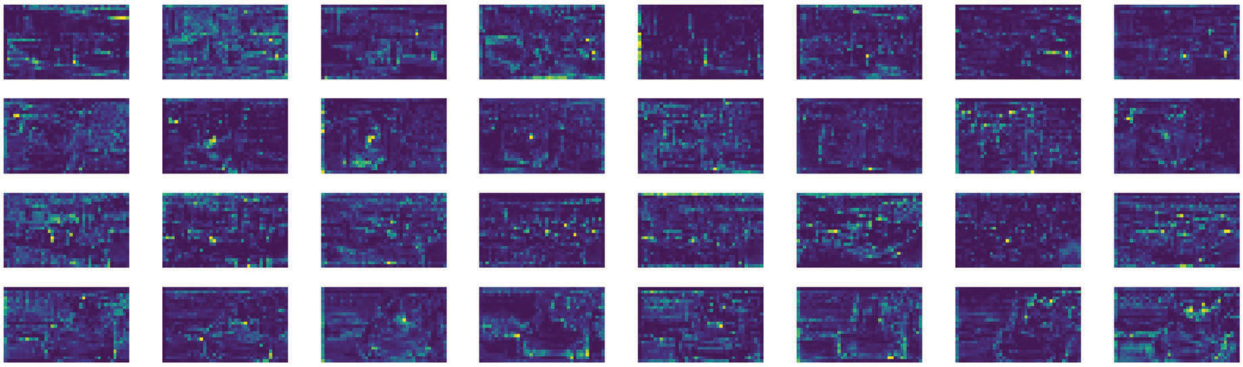
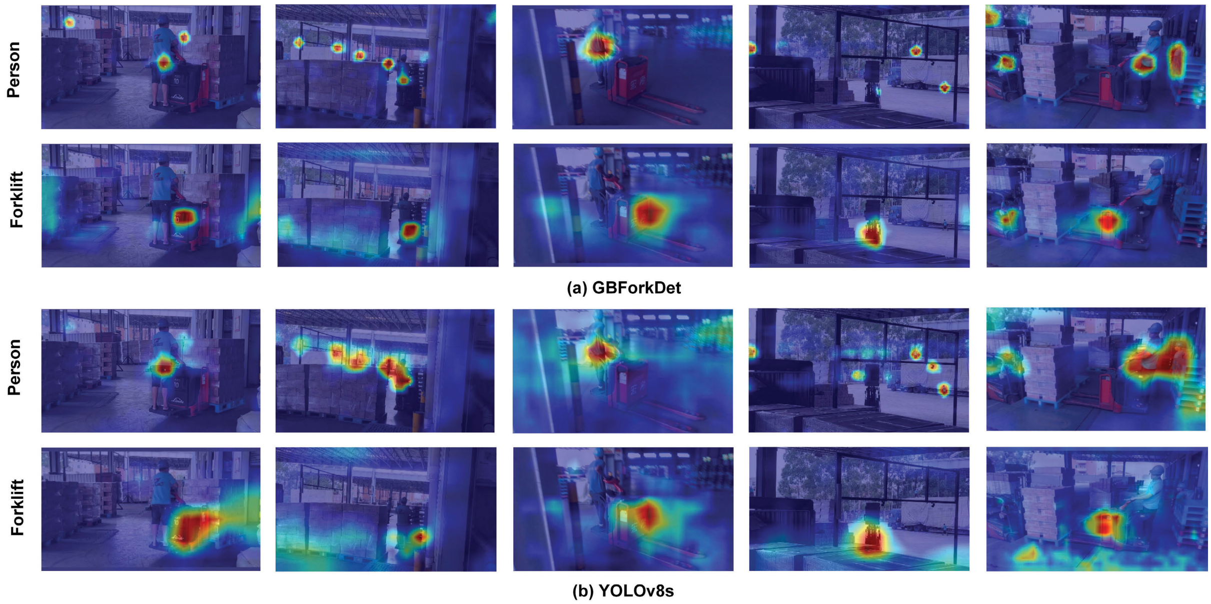


FIGURE 11. Visualization of the feature map of ODConv module.

of the BiODNeck module to enhance its detection performance. This enhancement can be attributed to BiODNeck's increased focus on semantic information in the model's context, enabling effective detection of occluded objects in complex logistics scenes. Consequently, the GBForkDet model achieved more accurate detection of targets in logistics



(a) GBForkDet

(b) YOLOv8s

FIGURE 12. The Grad-CAM graph of the proposed GBForkDet model.

TABLE 1. Performance results of different lightweight models on the Forklift-3k dataset.

Model	forklift(%)	person(%)	mAP(%)	Parameters(M)	FLOPs(G)	time(ms)
YOLOv3	89.6	77.8	83.7	9.31	23.4	11.4
YOLOv3-spp	90.2	80.1	85.1	9.57	23.6	12.6
YOLOv3-tiny	90.3	72.3	81.3	8.67	13.0	3.6
YOLOv4	90.0	78.5	84.5	9.12	20.8	13.2
YOLOv5s	93.4	88.0	90.7	7.02	16.0	10.6
YOLOv5-Lite	92.0	82.1	87.0	4.38	8.8	9.1
YOLOv7-tiny	92.4	87.2	89.8	6.02	13.2	8.5
YOLOv8s	94.1	87.9	91.0	11.16	28.8	9.8
GBForkDet(ours)	94.4	91.0	92.7	9.16	22.0	7.4

TABLE 2. Performance results of different lightweight models on the KITTI dataset.

Model	car(%)	cyclist(%)	person(%)	mAP(%)	Parameters(M)	FLOPs(G)	time(ms)
YOLOv3	97.7	93.6	84.1	91.8	9.31	23.4	11.7
YOLOv3-spp	97.9	95.0	83.9	92.3	9.57	23.6	12.9
YOLOv3-tiny	94.3	85.3	73.2	84.3	8.67	13.0	3.7
YOLOv4	97.9	94.9	84.9	92.6	9.12	20.8	13.6
YOLOv5s	98.3	94.9	86.1	93.1	7.02	16.0	10.8
YOLOv5-Lite	97.1	92.4	80.1	89.9	4.38	8.8	9.4
YOLOv7-tiny	97.6	93.7	78.5	90.0	6.02	13.2	8.9
YOLOv8s	98.6	94.6	87.8	93.7	11.16	28.8	10.1
GBForkDet (ours)	98.7	97.5	89.7	95.3	9.16	22.0	7.6

scenes, surpassing the original model's performance under challenging conditions such as long distances, occlusions, or uneven lighting. The proposed GBForkDet method effectively determines the position of the targets, providing visual evidence of its significant impact on logistics object detection.

E. ABLATION EXPERIMENT

To demonstrate the effectiveness of the proposed improvement strategies, this study conducted ablation studies on the Forklift-3k dataset and designed six sets of comparative experiments in the same environment. The experimental results are shown in Table 3, indicating that the improvement strategies significantly enhance the detection performance of logistics objects in complex logistics backgrounds. The Ghost module can reduce parameters and computational costs in convolutional neural networks. It achieves this by employing two key strategies: feature channel grouping and reconstruction, and the application of depth-wise separable convolutions. In this study, the Ghost module is applied to the Backbone network, replacing the C2f module. This results

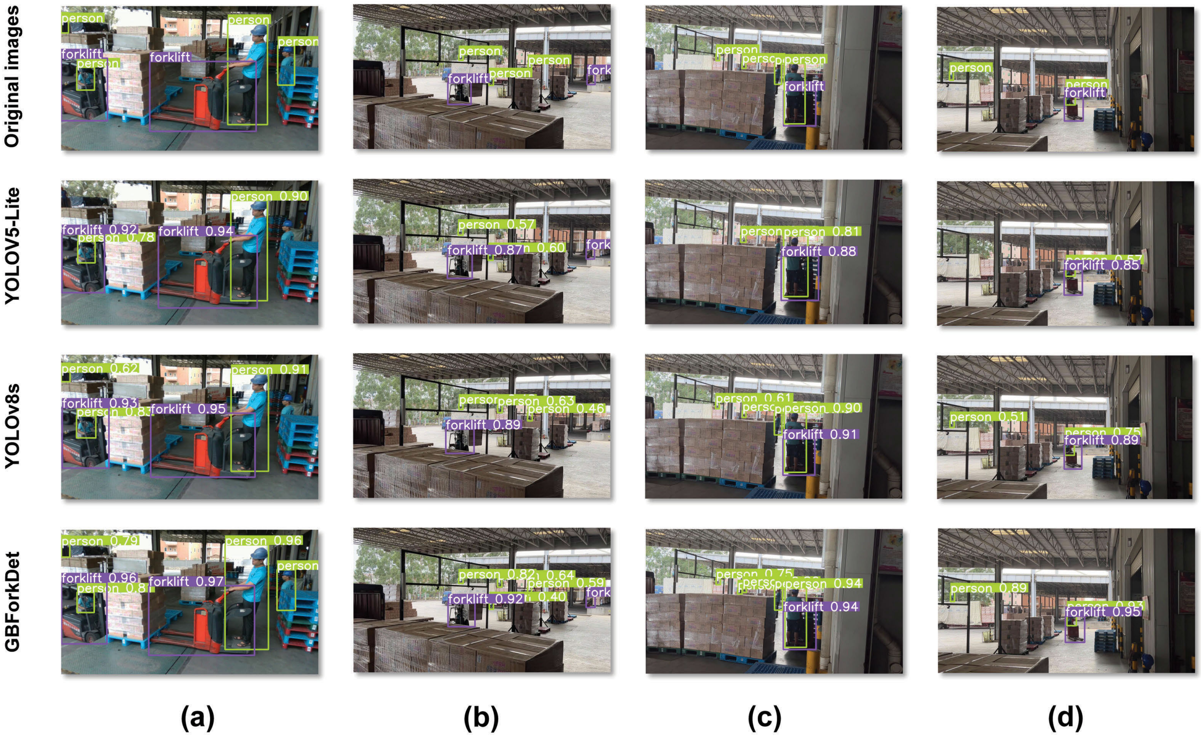


FIGURE 13. Comparison of detection results on the Forklift-3k dataset.

TABLE 3. Ablation experiments for each module based on the Forklift-3k dataset.

Baseline	Ghost	BiODNeck	NWD	mAP(%)	Parameters(M)	FLOPs(G)	time(ms)
√				91.0	11.16	28.4	9.8
√	√			91.2	9.05	22.6	6.9
√		√		91.9	11.25	28.1	10.1
√			√	91.6	11.16	28.4	9.8
√	√	√		92.0	9.16	22.0	7.3
√	√	√	√	92.7	9.16	22.0	7.4

TABLE 4. Performance results for each layer of modules in the GBForkDet model.

Layer	Module	Parameters(M)	FLOPs(G)	time(ms)	Layer	Module	Parameters(M)	FLOPs(G)	time(ms)
0	Conv	0.001	0.18	23.4	12	C2f	0.591	1.89	25.1
1	Conv	0.018	0.94	16.6	13	Upsample	0.000	0.00	1.4
2	C3Ghost	0.009	0.48	26.0	14	Concat	0.000	0.00	2.2
3	Conv	0.073	0.94	13.8	15	C2f	0.148	1.89	49.1
4	C3Ghost	0.040	0.50	30.7	16	Conv	0.147	0.47	5.9
5	Conv	0.295	0.94	11.5	17	Concat	0.000	0.00	0.6
6	C3Ghost	0.154	0.49	12.2	18	C2f	0.493	1.57	15.7
7	Conv	1.180	0.94	13.4	19	Conv	0.590	0.47	7.1
8	C3Ghost	0.564	0.45	10.8	20	Concat	0.000	0.00	0.3
9	SPPF	0.656	0.52	11.4	21	C2f	1.969	1.57	20.1
10	Upsample	0.000	0.00	0.7	22	Detect	2.147	8.29	242.6
11	Concat	0.000	0.00	1.2					

in a reduction of 2.11M parameters and a 20.5% decrease in GFLOPs. By introducing BiODNeck, the mAP was improved by 0.9% with almost no increase in model parameters and computational complexity. Lastly, by optimizing the loss function of GBForkDet using NWD to enhance the detection of small objects, a 0.6% increase in mAP was achieved. Compared to the baseline model, the final GBForkDet model achieved a 1.4% improvement in mAP and reduced FLOPs from 28.4 to 22.0, validating the effectiveness of GBForkDet.

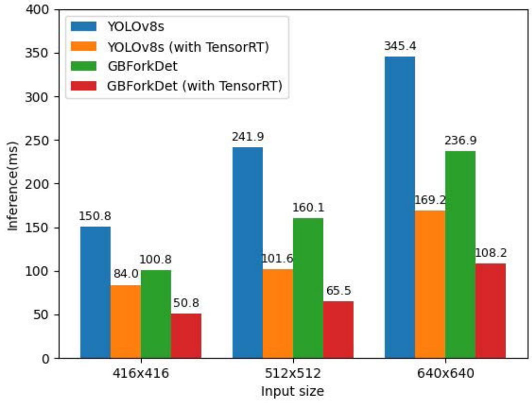


FIGURE 14. Comparison of inference speed of YOLOv8s and GBForkDet before and after TensorRT acceleration at different input sizes.

F. EMBEDDED PLATFORM DEPLOYMENTS

This paper uses the YOLOv8s and GBForkDet models and conducts inference acceleration experiments using TensorRT on the Jetson Nano platform. The inference speed is evaluated for 300 images with different input sizes, and the experimental results are shown in Fig. 14. GBForkDet has a significant detection speed advantage over YOLOv8s on the Jetson Nano platform, which has limited computational resources. The application of TensorRT acceleration yields a significant improvement in the detection speed of the models. When the image size is set to 640×640, the GBForkDet model with FP16 precision achieves a remarkable detection time of only 108.2 ms per image.

Table 4 presents the performance results of the GBForkDet model on the Jetson Nano platform for each layer of modules. This paper provides a detailed analysis of the model parameters, FLOPs, and inference time for each module, visually demonstrating the performance of each module in edge devices.

V. CONCLUSION

This paper proposes a novel detection model, GBForkDet, developed explicitly for ensuring forklift safety driving in complex logistics environments. The model incorporates a lightweight GhostNet to optimize the Backbone network. A novel BiODNeck is also designed to reconstruct the Neck component in YOLOv8. Furthermore, the loss function is enhanced through the NWD, effectively mitigating the model's sensitivity toward detecting small targets.

Utilizing TensorRT on the jetson nano edge platform, GBForkDet achieves substantial improvements in inference speed. The proposed model demonstrates an excellent trade-off between inference speed and detection accuracy, meeting the specific industrial demands of logistics scenarios. Future research efforts will be dedicated to exploring advanced model compression techniques. In this study, we obtained ethical and informed consent from the participating companies for the use of their data. To ensure the privacy of the participants, this paper did not display their full faces.

REFERENCES

[1] 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.*, Jun. 2014, pp. 580–587.

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

[3] 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.*, vol. 28, 2015, pp. 1–9.

[4] 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.

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

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

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

[8] 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*. Amsterdam, The Netherlands: Springer, 2016, pp. 21–37.

[9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[10] 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)*, Jul. 2017, pp. 936–944.

[11] 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.

[12] J. Zhao, S. Hao, C. Dai, H. Zhang, L. Zhao, Z. Ji, and I. Ganchev, "Improved vision-based vehicle detection and classification by optimized YOLOv4," *IEEE Access*, vol. 10, pp. 8590–8603, 2022.

[13] X. Dai, "HybridNet: A fast vehicle detection system for autonomous driving," *Signal Process., Image Commun.*, vol. 70, pp. 79–88, Feb. 2019.

[14] Z. Jin, Q. Zhang, C. Gou, Q. Lu, and X. Li, "Transformer-based vehicle detection for surveillance images," *J. Electron. Imag.*, vol. 31, no. 5, Jun. 2022, Art. no. 051602.

[15] S. Shirmohammadi and A. Ferrero, "Camera as the instrument: The rising trend of vision based measurement," *IEEE Instrum. Meas. Mag.*, vol. 17, no. 3, pp. 41–47, Jun. 2014.

[16] H. Huttunen, F. S. Yancheshmeh, and K. Chen, "Car type recognition with deep neural networks," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2016, pp. 1115–1120.

[17] Y. Gao, S. Guo, K. Huang, J. Chen, Q. Gong, Y. Zou, T. Bai, and G. Overett, "Scale optimization for full-image-CNN vehicle detection," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 785–791.

[18] Y. Tang, C. Zhang, R. Gu, P. Li, and B. Yang, "Vehicle detection and recognition for intelligent traffic surveillance system," *Multimedia Tools Appl.*, vol. 76, no. 4, pp. 5817–5832, Feb. 2017.

[19] Y. Cai, T. Luan, H. Gao, H. Wang, L. Chen, Y. Li, M. A. Sotelo, and Z. Li, "YOLOv4–5D: An effective and efficient object detector for autonomous driving," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021.

[20] H. Wang, Y. Xu, Y. He, Y. Cai, L. Chen, Y. Li, M. A. Sotelo, and Z. Li, "YOLOv5-fog: A multiobjective visual detection algorithm for fog driving scenes based on improved YOLOv5," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022.

[21] J. Terven and D. Cordova-Esparza, "A comprehensive review of YOLO: From YOLOv1 and beyond," 2023, *arXiv:2304.00501*.

[22] T. Zeng, S. Li, Q. Song, F. Zhong, and X. Wei, "Lightweight tomato real-time detection method based on improved YOLO and mobile deployment," *Comput. Electron. Agricult.*, vol. 205, Feb. 2023, Art. no. 107625.

[23] J. Chen, S. Deng, P. Wang, X. Huang, and Y. Liu, "Lightweight helmet detection algorithm using an improved YOLOv4," *Sensors*, vol. 23, no. 3, p. 1256, Jan. 2023.

[24] W. Zhou, Y. Zhu, J. Lei, R. Yang, and L. Yu, "LSNet: Lightweight spatial boosting network for detecting salient objects in RGB-thermal images," *IEEE Trans. Image Process.*, vol. 32, pp. 1329–1340, 2023.

[25] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, ''SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size,'' 2016, *arXiv:1602.07360*.

[26] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, ''MobileNets: Efficient convolutional neural networks for mobile vision applications,'' 2017, *arXiv:1704.04861*.

[27] 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.

[28] X. Zhang, X. Zhou, M. Lin, and J. Sun, ''ShuffleNet: An extremely efficient convolutional neural network for mobile devices,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.

[29] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, ''ShuffleNet V2: Practical guidelines for efficient CNN architecture design,'' in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 116–131.

[30] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, ''GhostNet: More features from cheap operations,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1577–1586.

[31] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' 2014, *arXiv:1409.1556*.

[32] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, ''Densely connected convolutional networks,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[33] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, ''CSPNet: A new backbone that can enhance learning capability of CNN,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 1571–1580.

[34] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, ''CondConv: Conditionally parameterized convolutions for efficient inference,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[35] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, ''Dynamic convolution: Attention over convolution kernels,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11027–11036.

[36] C. Li, A. Zhou, and A. Yao, ''Omni-dimensional dynamic convolution,'' 2022, *arXiv:2209.07947*.

[37] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, ''Distance-IoU loss: Faster and better learning for bounding box regression,'' in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 12993–13000.

[38] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, ''Focal and efficient IOU loss for accurate bounding box regression,'' *Neurocomputing*, vol. 506, pp. 146–157, Sep. 2022.

[39] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, ''UnitBox: An advanced object detection network,'' in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 516–520.

[40] J. Wang, C. Xu, W. Yang, and L. Yu, ''A normalized Gaussian Wasserstein distance for tiny object detection,'' 2021, *arXiv:2110.13389*.

[41] A. Geiger, P. Lenz, and R. Urtasun, ''Are we ready for autonomous driving? The KITTI vision benchmark suite,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

LINHUA YE received the B.S. degree in computer science and technology from the Guangdong University of Finance, Guangzhou, China, in 2021. He is currently pursuing the M.S. degree in electronic information with Fujian Normal University, Fuzhou, China.

His current research interests include deep learning, computer vision, and object detection.

SONGHANG CHEN (Senior Member, IEEE) received the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2014.

From 2014 to 2016, he was an Assistant Researcher with the State Key Laboratory for Management and Control of Complex Systems, Beijing. He is currently an Associate Professor and a member of the Computational Intelligence and Industrial Big Data Laboratory, Haixi Institutes, Chinese Academy of Sciences, Jinjiang, China. His current research interests include multiobjective optimization, high-performance computing, big data, and the Internet of Things.