IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 25, 2022, accepted April 5, 2022, date of publication April 12, 2022, date of current version April 20, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3166923

ALODAD: An Anchor-Free Lightweight Object Detector for Autonomous Driving

TIANJIAO LIANG, HONG BAO, WEIGUO PAN, AND FENG PAN
Beijing Key Laboratory of Information Service Engineering, Beijing Union University, Beijing 100101, China
College of Robotics, Beijing Union University, Beijing 100101, China

Corresponding author: Weiguo Pan (ldtweiguo@buu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61802019, Grant 61932012, Grant 61871039, Grant 61906017, and Grant 62006020; in part by the Beijing Municipal Education Commission Science and Technology Program under Grant KM201911417003, Grant KM201911417009, and Grant KM201911417001; in part by the Beijing Union University Research and Innovation Projects for Postgraduates under Grant YZ2020K001 and Grant YZ2021K001; and in part by the Premium Funding Project for Academic Human Resources Development, Beijing Union University, under Grant BPHR2020DZ02.

ABSTRACT Vision-based object detection is an essential component of autonomous driving. Because vehicles typically have limited on-board computing resources, a small-sized detection model is required. Simultaneously, high object detection accuracy and real-time inference detection speeds are required to ensure safety while driving. In this paper, an anchor-free lightweight object detector for autonomous driving called ALODAD is proposed. ALODAD incorporates an attention scheme into the lightweight neural network GhostNet and builds an anchor-free detection framework to achieve lower computational costs and provide parameters with high detection accuracy. Specifically, the lightweight backbone neural network integrates a convolutional block attention model that analyzes the valuable features from traffic scene images to generate an accurate bounding box, and then constructs feature pyramids for multi-scale object detection. The proposed method adds an intersection over union (IoU) branch to the decoupled detector to rank the vast number of candidate detections accurately. To increase the data diversity, data augmentation was used during training. Extensive experiments based on benchmarks demonstrate that the proposed method offers improved performance compared to the baseline. The proposed method can achieve an increased detection accuracy while meeting the real-time requirements of autonomous driving. The proposed method was compared with the YOLOv5 and RetinaNet models and 98.7% and 94.5% were obtained for the average precision metrics AP50 and AP75, respectively, on the BCTSDB dataset.

INDEX TERMS Autonomous driving, deep learning, lightweight, object detection.

I. INTRODUCTION

Autonomous driving will change the way we travel in the future and will be vital to the development of national and global economies. Commercial applications of autonomous driving have been realized for specific scenarios to date. However, because the current technology directions are mainly based on lidar, the system cost is high, and large-scale deployment cannot be realized in this way. Vision-based methods have become a research hotspot because of their low cost. Object detection is one of the most important aspects of the development of this technological approach. Object detection methods can help autonomous vehicles (AVs) detect and recognize traffic signs, signal lights, pedestrians, and vehicles in traffic scenes automatically and can then transmit the results to the vehicle's decision-making module to ensure that the vehicle is driven safely and in accordance with traffic rules.

In recent years, many algorithms have demonstrated good object detection performances based on deep learning. However, the detection of objects in real traffic scenes remains a challenge. Some researchers have used complex models to obtain a high traffic object detection performance. However, because the on-board computational resources of vehicles are limited, these complex models cannot be deployed in embedded devices, or they are unable to achieve real-time detection during autonomous driving. Improving the detection accuracy of such a model when deployed on an on-board computing unit remains challenging.

Large and complex models are difficult to apply to AVs because they have insufficient on-board memory and computing power. Scenarios in autonomous driving typically require

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

low latency and fast response speeds. Thus, the aim of this paper is to propose an object detection model that can realize the high detection accuracy required while maintaining small parameter sizes for autonomous driving applications.

To solve the problems described above, a lightweight detection framework based on a single stage was proposed. The contributions of the proposed method are as follows: (1) It improves an existing lightweight backbone network based on GhostNet [1]. The method integrates an attentional mechanism into the GhostNet backbone network, which can improve the network and allow it to focus on the objects to be detected, and uses a feature pyramid network for multi-scale detection. (2) A novel complete intersection over union (CIoU)-aware head based on an anchor-free detector and a new confidence calculation method are designed to enhance the correlation between object classification and localization. (3) A data augmentation approach driven by complex traffic scenarios was used to provide a more diverse dataset for training.

The remainder of this paper is organized as follows. In Section 2, we introduce related works on object detection in recent years. The details of the proposed method are presented in section 3. Section 4 focuses on the implementation of the proposed method and compares it with previous methods. Section 5 summarizes the conclusions of the work completed in this study and suggests future development directions for this work.

II. RELATED WORK

A. OBJECT DETECTION

Traditional object detection (using histograms of oriented gradients plus a support vector machine (HOG [2] + SVM [3]) approach) works by selecting candidate regions from a given image, extracting features from these regions, and then classifying these features using a trained classifier. In recent years, the rapid development of deep convolutional neural networks has led to performance improvements in the object detection field. In general, deep learning-based object detection methods can be divided into two types: single-stage methods, such as you only look once series (e.g., YOLO v1 [4], YOLO v2 [5], and YOLO v3 [6]), the single shot multibox detector (SSD) [7], and RetinaNet [8], and multi-stage methods, such as the two-stage region convolutional neural network (R-CNN) series [9]–[11], and cascade R-CNN [12]. The detection speeds of multi-stage methods make it difficult to achieve real-time detection, whereas one-stage detection algorithms can greatly improve the operating speed based on the premise that high accuracy is ensured.

In recent years, researchers have begun to focus on the application of object-detection methods to AVs. Wang *et al.* [13] proposed an improved faster R-CNN for traffic sign detection. Han *et al.* [14] used a revised faster R-CNN for small traffic sign detection. These studies achieved high detection accuracy, but the detection speeds when used on traffic scenes demonstrated the limitations of these methods. He *et al.* [15] used a one-stage detector called YOLO-MXANet to perform small object detection in traffic scenes to improve detection speed. Based on mask R-CNN, the mask scoring (MS) R-CNN approach [16] uses a mask IoU head to learn the predicted mask quality and then obtain a new network structure that combines the characteristics of the example with the corresponding predictive mask to enable regression to the mask IoU. Jiang *et al.* proposed IoU-Net [17] to improve the interpretability of the regression by proposing an IoU-guided non maximum suppression (NMS) to introduce localization confidence as an ordering index in the NMS, and proposed an optimization-based bounding box refinement to replace the traditional regression-based method. Fan *et al.* [18] used CornerNet [19] with foreground attention to detect traffic objects. Xu *et al.* [20] used the center-based detection algorithm, FCOS [21], to detect objects in mobile scenarios. Some of the detection methods described above have already been applied to AVs.

However, there is still a problem regarding the low correlation between the classification score and localization accuracy. Generally, the final scores of the predicted boxes used in NMS are taken from the classification scores alone, and the localization information is not considered. In Figure 1, C represents the classification score. A high classification score with low IoU bounding boxes (Fn) cannot accurately represent the location information of an object, and it suppresses accurate boxes with high IoU (Tn) values during NMS when only classification scores are used for the final scores. This mismatch problem between classification and localization causes anchors with high IoU values but low classification scores to be filtered during the NMS. In this study, we propose a novel traffic object detection model to solve this problem.

B. FEATURE EXTRACTION

Feature extraction is an essential step for object detection. Traditional feature extraction methods are based on the use of handcrafted features. Yao *et al.* [22] proposed a traffic sign feature extraction method based on HOG features. Pedro *et al.* [23] proposed a deformable part model (DPM) to extract object features for both vehicles and pedestrians. The performance of these traditional methods is limited because they lack the ability to acquire spatial and semantic information. Their slow extraction speeds and low representational abilities cannot meet the requirements of autonomous driving systems. In recent years, deep convolutional neural network (CNN) algorithms have been widely used for feature extraction applications because of their competitive performance.

Additional feature extraction networks based on deep learning have been proposed based on AlexNet [24], which has a relatively small receptive field because of its limited network depth and the number and size of its convolution kernels. The visual geometry group network (VGGNet) [25] simplifies the network design workflow by increasing the network depth and stacking small convolutions to expand the receptive field, reduce the number of network parameters, and stack the same types of network blocks repeatedly. The

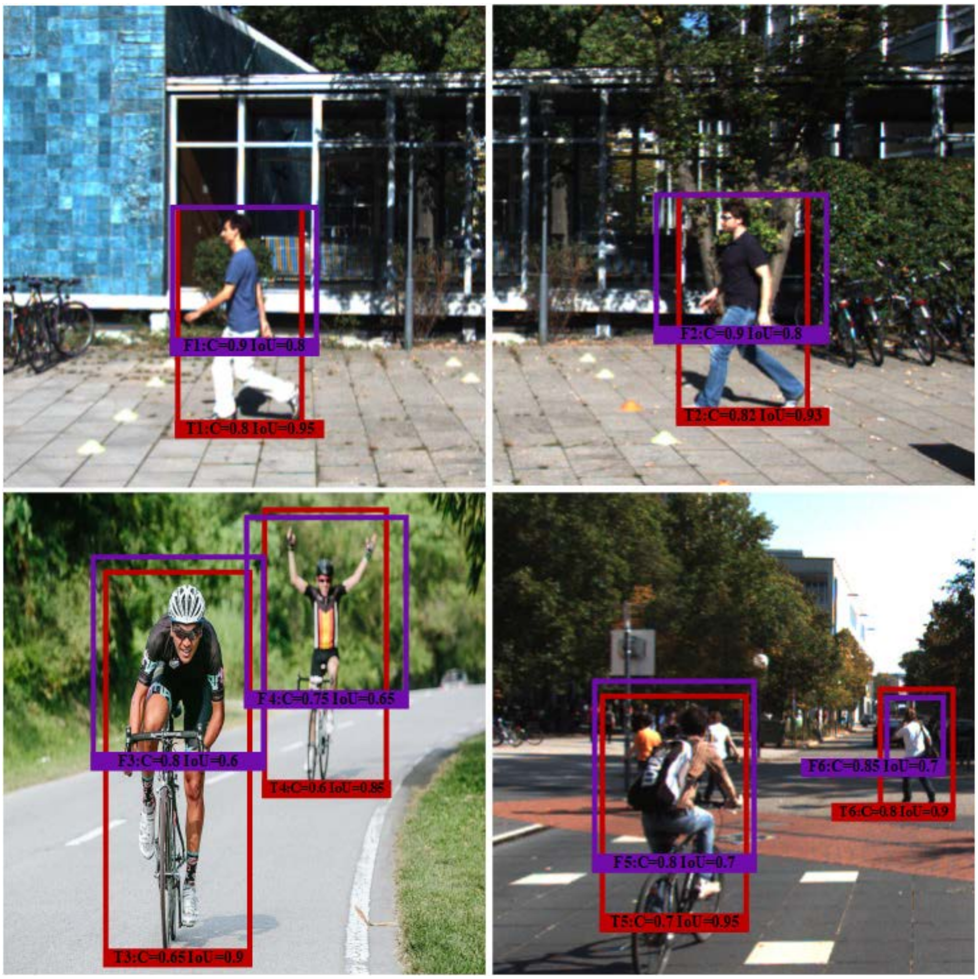


FIGURE 1. Mismatch problems between classification and localization. "C" represents the classification score. "F" means the final inaccurate detect boxes with higher classification score than "T" boxes but suppress the accurate "T" boxes with higher IoU during non-maximum suppression.

network in network [26] structure first uses a global average pooling layer to replace the fully connected layer and then uses a 1 × 1 convolution layer to learn a nonlinear combination of the feature graph channels, which has become the mainstream method for feature fusion. GoogLeNet [27] uses convolution kernels of different sizes to provide enhanced multi-scale detection capabilities. However, the large number of required parameters limits the computational power of this network. Residual networks (ResNets) [28] introduced a residual block to reduce the gradient disappearance problem in deeper neural networks, thereby allowing these networks to acquire deeper features. Jung *et al.* [29] used ResNet to perform vehicle classification and localization in traffic surveillance systems.

The networks described above are not designed for use in mobile or embedded devices. Improvements in calculation accuracy also cause excessive memory consumption and computing power mismatch. The aim of this study is to design a more efficient network by reducing the number of network parameters without compromising network performance.

MobileNet [30] is a convolutional neural network proposed by Google that is small in size and less computationally expensive, and is thus suitable for use in mobile devices. MobileNet uses depth-wise separable convolution and width multiplication to reduce the number of required network parameters. The depth-wise separable convolution method decomposes a standard convolution into depth-wise and point-wise convolutions. The number of floating-point operations (FLOPs) of a standard convolution is given by K^2MHWN, whereas that of the depth separable convolution is given by $\left(\mathrm{K}^2\mathrm{+M}\right)$ HWN.

$$\frac{depthwise + pointwise}{conv} = \frac{\left(K^2 + M\right) HWN}{K^2 MHWN} = \frac{1}{M} + \frac{1}{K^2} \tag{1}$$

In the general network architecture, M (number of output feature channels) $\gg K^2$ (convolution kernel size squared) (e.g., $K = 3$ and $M \geq 32$), and H, W, N are defined as the height, width, and number of channels, respectively. Biswas *et al.* [31] used an SSD and MobileNet to perform automatic traffic density estimation. ShuffleNet [32] mainly uses channel shuffle methods, point-wise group convolutions, and depth-wise convolution to modify the original residual blocks, thus reducing the number of arguments and computations required. Chen *et al.* [33] proposed an efficient neural network to perform point cloud analysis by shuffling the feature channels to capture fine-grained features. Although the models above can achieve better performance when implemented under a lightweight network framework, there is a lot of redundancy between feature maps, which increases the calculation of the feature map, and most of these calculations are redundant. GhostNet was proposed as a new end-to-side neural network architecture intended to use the redundancy between the feature graphs to generate feature graphs at a lower cost, as illustrated in Fig. 2. It use "cheap operation" to alleviate the increased computation due to content redundancy between feature maps of the same layer, which can reduce the computation and improve the detection speed of the model while maintaining the same detection accuracy. Based on the original feature image, the algorithm uses linear transformation to generate ghost feature maps that can extract the required information from the original feature maps with lower computational costs.

The main purpose of these lightweight networks is that they are designed to perform classification tasks and do not have the ability to identify local features. In this study, we focus on ways to improve the representation of local region features in lightweight networks.

III. PROPOSED METHOD

The method proposed in this study is based on an anchor-free approach that can reduce the number of calculations caused by the use of an anchor, with the aim of making the detection method move further toward high real-time accuracy.

The proposed method is illustrated in Fig. 3. The network architecture can be divided into three parts: backbone, feature pyramid network, and prediction head. We integrated the convolutional block attention model (CBAM) [34] into GhostNet to generate the attention map sequentially along the channel-wise and spatial-wise dimensions, which can find the attention region and extract its features more effectively in a lightweight manner in autonomous driving scenarios. The prediction head is built on the feature pyramid network, which consists of two branches: one branch is used for regression, including bounding box localization and IoU prediction processes, and the other is used to perform classification. We separate the classification and regression tasks into two independent sub-networks and then add a synchronized CIoU-aware head to the tail of the regression branch to solve the mismatch problem.

A. LIGHTWEIGHT BACKBONE NETWORK

Convolutional neural networks usually require a large number of parameters and floating-point operations (FLOPs) to achieve high accuracy. GhostNet can reduce the number of computational steps required to generate feature maps at lower computational costs. In addition, GhostNet eliminated some of the same feature maps in subsequent steps without losing any information, thus providing a more efficient way to generate feature maps.

TABLE 1. Overall architecture of our backbone network.

Operator	Out_channel	Stride
Conv2d 3×3	16	2
GhostA bottleneck	16	1
GhostA bottleneck	24	2
GhostA bottleneck	24	1
GhostA bottleneck	40	2
GhostA bottleneck	40	1
GhostA bottleneck	80	2
GhostA bottleneck	80	1
GhostA bottleneck	80	1
GhostA bottleneck	80	1
GhostA bottleneck	112	1
GhostA bottleneck	112	1
GhostA bottleneck	160	2
GhostA bottleneck	160	1
GhostA bottleneck	160	1
GhostA bottleneck	160	1
GhostA bottleneck	160	1
Conv2d 1×1	256	1

GhostNet can solve the computational redundancy problem of traditional convolution operations; however, it ignores the need for effective feature extraction. In this study, we integrated CBAM into GhostNet to enhance the object area in the feature map. CBAM is an attention mechanism that combines spatial $M_s(\boldsymbol{F})$ and channel attention $M_c(\boldsymbol{F})$ information, which is defined as follows:

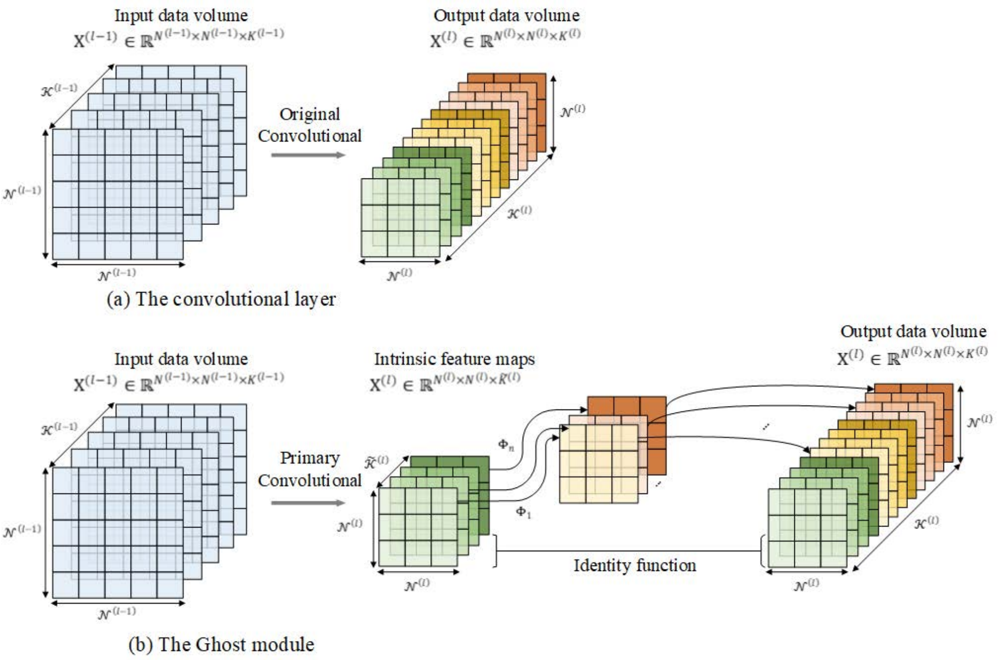
$$M_c(\boldsymbol{F}) = \sigma\left(MLP\left(AvgPool\left(\boldsymbol{F}\right)\right) + MLP\left(MaxPool\left(\boldsymbol{F}\right)\right)\right) \tag{2}$$

where σ denotes the sigmoid function, $\boldsymbol{F}$ denotes a feature map, *MLP* is a multilayer perceptron, and *AvgPool* and *MaxPool* represent average pooling and maximum pooling, respectively.

$$M_s(\boldsymbol{F}) = \sigma\left(f^{7\times7}\left(\left[AvgPool\left(\boldsymbol{F}\right);\ MaxPool\left(\boldsymbol{F}\right)\right]\right)\right) \tag{3}$$

where σ denotes the sigmoid function and $f^{7\times7}$ represents a convolution operation with a filter size of 7 × 7.

The proposed module composed of CBAM is shown in Fig. 4. Although its small model representation ability is weak and the upper limit of the potential performance is reduced, the experimental results show that the Ghost module with CBAM can provide stable performance improvements with only a small number of additional calculations. The architecture of the proposed backbone network is shown in Table 1. Here, Conv2d $n\times n$ represents a standard two-dimensional convolutional layer with an $n\times n$ kernel size. GhostA represents the Ghost attention bottleneck.


Input data volume
$X^{(l-1)} \in \mathbb{R}^{N^{(l-1)} \times N^{(l-1)} \times K^{(l-1)}}$
Original
Convolutional
Output data volume
$X^{(l)} \in \mathbb{R}^{N^{(l)} \times N^{(l)} \times K^{(l)}}$
$\mathcal{K}^{(l-1)}$
$\mathcal{N}^{(l-1)}$
$\mathcal{N}^{(l)}$
$\mathcal{K}^{(l)}$
Primary
Convolutional
Intrinsic feature maps
$X^{(l)} \in \mathbb{R}^{N^{(l)} \times N^{(l)} \times \tilde{K}^{(l)}}$
$\tilde{\mathcal{K}}^{(l)}$
Φ_n
Φ_1
Identity function

(a) The convolutional layer

(b) The Ghost module

FIGURE 2. GhostNet Module. The top part is the standard convolutional layer, and the bottom part is the Ghost module for outputting the same number of feature maps. Φ represents an inexpensive operation.

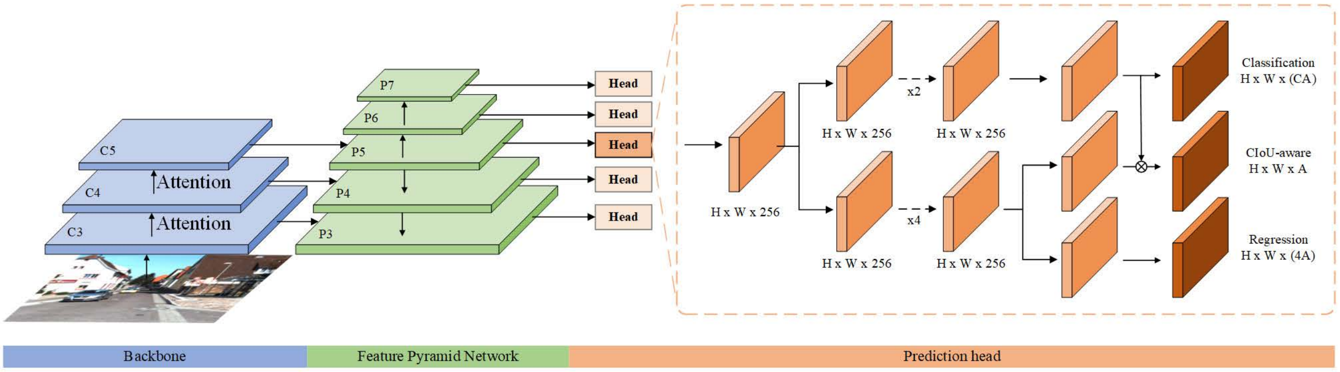


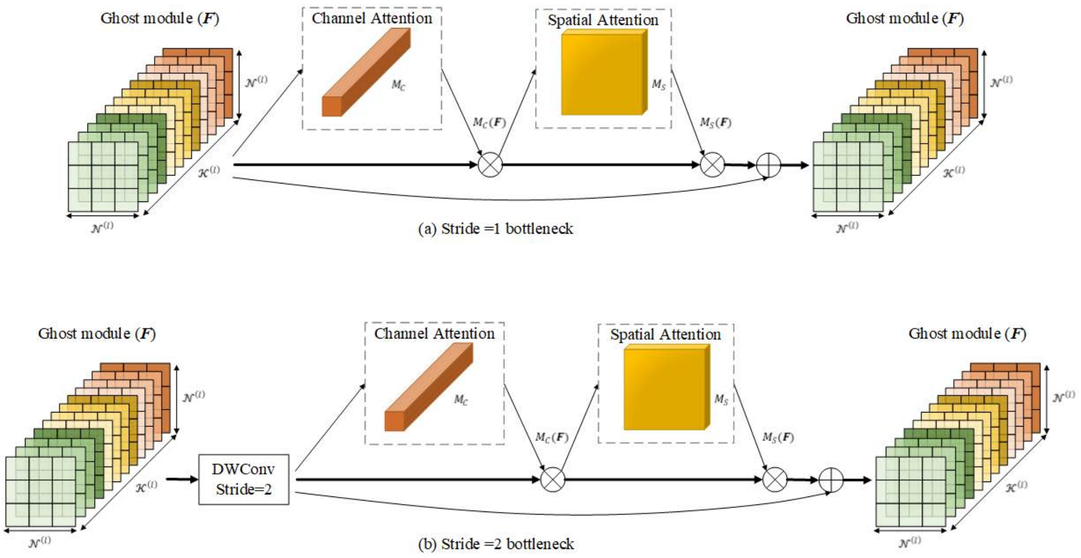
FIGURE 3. Architecture of the proposed detector. P3 to P7 denote the feature maps used for the prediction. H and W are the height and width of feature maps.

B. MULTI-SCALE DETECTION

The overlap between different ground truths may cause ambiguity that is difficult to handle during the training process. This ambiguity leads to a reduction in the detector performance. In this study, we show that the multiscale prediction method can effectively solve this problem. Following the approaches of the feature pyramid network (FPN) [35] and pyramid attention network (PAN) [36], the method in this study uses different levels of feature layers to detect objects of different sizes. We constructed a pyramid with five-scale feature maps$\{P_3, P_4, P_5, P_6, P_7\}$, where the subscripts indicate the pyramid levels. P_3, P_4 and P_5 were extracted using the backbone network layers $\{C_3, C_4, C_5\}$ and by performing a top-down convolution to reduce the degradation that occurred as the depths of the convolutional layers increased. P_6, P_7 were processed using a 3 × 3 convolution with two strides from P_5, P_6, respectively. Multilevel detection shares information between the different feature layers, which can make the detector parameters more efficient and thus improve the detection performance.

C. CIOU-AWARE DECOUPLED HEAD

In object detection, the models that perform classification and regression tasks are relatively independent. A classification step is performed to divide the objects by category, and a regression step is performed to predict the locations of the objects. These two tasks differ in their functionality and complexity. The different tasks should be completed using different branches. However, one-stage detection models, such as the YOLO series, are in a process of continuous



(a) Stride =1 bottleneck

(b) Stride =2 bottleneck

FIGURE 4. Ghost attention bottleneck. The top part is the Ghost attention bottleneck with stride = 1, which integrates the convolutional block attention model into Ghost bottleneck. The bottom part is the Ghost attention bottleneck with stride = 2 added depthwise convolutional layer. $M_S(F)$ and $M_C(F)$ denote the spatial and channel attention, respectively.

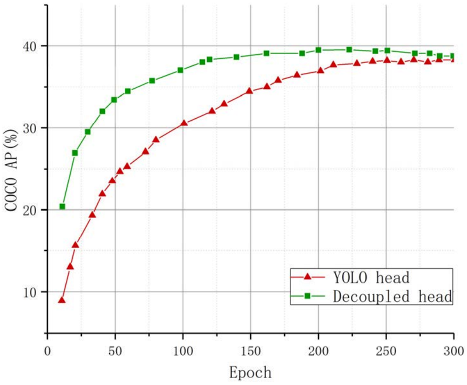


FIGURE 5. Training curves for detectors with YOLOv3 head or decoupled head which verifies that use of the decoupled head for the single-stage model improves the convergence speed.

evolution, which means that their detector heads remain coupled. Inspired by the segmenting objects by locations (SOLO) [37] instance segmentation algorithm, we propose a new detection head for object detection. As shown in Fig. 3, this involves using a decoupled head to replace the coupled head.

The decoupled head contained a convolutional layer with a 1 × 1 kernel to reduce the number of channels, and this layer was followed by two parallel branches. The classification branch contained two convolutional layers, and the regression branch contained four convolutional layers. Experiments have verified that the use of the decoupled head for the single-stage model significantly improves the convergence speed, as illustrated in Fig. 5.

The low correlation observed between the classification score and localization accuracy reduces the dense object detection capability of the detector. This mismatch problem between classification and localization causes only the classification confidence to be used for bounding box sorting in NMS, which means that anchors with high IoU and low classification scores will be filtered.

To solve this mismatch problem, the proposed method adds a synchronized subnetwork at the end of the regression branch. Specifically, the CIoU-aware head adds classification branch feature maps to increase the impact of classification on IoU predictions. This approach was used to assist in calculating the anchor score in the final NMS step. Therefore, the complete prediction head contained two subnetworks and three heads. The final score S_{conf} of the anchor used in the final NMS step was obtained by adding the classification confidence to the IoU predicted by inference.

$$S_{conf} = \alpha p_i + (1 - \alpha)\, CIoU_i \tag{4}$$

The parameter α here is a control coefficient used to balance the classification result and predict the CIoU [38] in the range [0, 1]. S_{conf} considers the impact of both classification and localization on the inference results and reflects both the category and location information of the object, which is a more accurate detection confidence that can meet the object detection task requirements. S_{conf} is used in NMS and can reduce the ranking of object detection with a high classification score and poor localization by altering the influence of the classification and localization on the score value.

In the proposed method, we used CIoU for bounding box regression and a CIoU-aware head. CIoU solves the problem in which it is not possible to directly optimize the parts in which the bounding box and ground truth do not overlap. The distance between the two boxes, overlap rate, scale, and penalty terms are all considered, making the bounding box regression more stable as a result. This can also prevent divergence during the training. The loss function of CIoU adds the impact term βv based on the loss function of the distance-IoU (DIoU) [38], which considers the length-to-width ratio between the predicted and ground-truth boxes.

The CIoU is defined as:

$$CIoU_i = IoU_i - \frac{\rho^2\left(b_i, b_i^{gt}\right)}{c_i^2} - \beta v \tag{5}$$

$$\beta = \frac{v}{(1 - IoU_i) + v} \tag{6}$$

$$v = \frac{4}{\pi^2}\left(\left(arctan\frac{w_i^{gt}}{h_i^{gt}}\right) - arctan\frac{w_i}{h_i}\right)^2 \tag{7}$$

where β is a trade-off parameter and v is a parameter used to measure the consistency of the aspect ratio. Furthermore, $\rho(\cdot)$ is the distance between the central points of the two boxes, and c is the diagonal length of the smallest enclosing box that covers the two boxes.

The loss functions of the proposed model are as follows:

$$\mathcal{L}_{cls} = \frac{1}{N}\sum_{i}^{N} FL\left(p_i, \hat{p}_i\right) \tag{8}$$

$$\mathcal{L}_{loc} = \frac{1}{N_{Pos}}\sum_{i\in Pos}^{N} 1 - IoU_i + \frac{\rho^2\left(b_i, b_i^{gt}\right)}{c_i^2} + \beta v \tag{9}$$

$$\mathcal{L}_{iou} = \frac{1}{N_{Pos}}\sum_{i\in Pos}^{N} BCE\left(CIoU_i, C\hat{I}oU_i\right) \tag{10}$$

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \mathcal{L}_{loc} + \mathcal{L}_{iou} \tag{11}$$

The total loss function of the proposed model comprises the following three parts: The first is the classification loss $\mathcal{L}_{cls}$, which includes focal loss (FL). $\mathcal{L}_{iou}$ is part of the CIoU-aware head, which includes the binary cross-entropy loss, and $\mathcal{L}_{loc}$ is part of the bounding box regression. $\mathcal{L}_{loc}$ and $\mathcal{L}_{iou}$ are only computed for positive examples.

D. DATA AUGMENTATION

Deep convolutional neural networks have been successfully applied in the field of computer vision. This type of network is data-driven and requires a large quantity of training data. As the depth of network architecture increases, an increasing number of parameters must be learned. In the proposed method, traditional global pixel augmentation methods (e.g., random scaling, cropping, translation, shearing, and rotation) were used to enhance data diversity. We also used data augmentation methods that have been proposed in recent years, as illustrated in Fig. 6; for example, MixUp [39] mixes two random samples proportionally, and the classification results are then distributed proportionally; Cutout [40] replaces the sampled regions at random with zero-pixel values, and the ground truth remains unchanged; CutMix [41] fills parts of other images from the training dataset into the sample, and the ground truth is then distributed with a certain proportionality. This can improve the robustness of the model without incurring additional cost.

IV. EXPERIENCE AND RESULTS

A. DATASET AND EVALUATION METRICS.

The common objects in context (COCO) [42] datasets were used to evaluate the generalization ability of the model, and the BCTSDB [43] and KITTI [44] datasets were used to test the model's detection ability in traffic scenarios. BCTSDB is a traffic sign dataset that includes 15,690 images and 25,243 annotations with 14121 training images and 1569 test images, and has labels that are divided into three categories: prohibitory, mandatory, and warning. The KITTI dataset includes three categories, comprising vehicles, pedestrians, and cyclists, and consists of 7481 training images and 7518 test images, with 80,256 labelled objects in total. In this study, all of our experiments followed the COCO format. The training set was randomly selected from the dataset and the remainder was used as the test set. The training set was used to train the model and the test set was used to test the model performance. The final experimental results were obtained by repeating this operation thrice and averaging the results.

The experiment used the average precision (AP) to compare the different models and their respective accuracies, including AP (IoU =.50–.95), AP$_{50}$ (IoU =.50), AP$_{75}$ (IoU =.75), AP$_{\mathrm{L}}$ (large, area $> 96^2$), AP$_{\mathrm{M}}$ (medium, $32\ ^2 <$ area $< 96^2$), and AP$_{\mathrm{S}}$ (small, area $< 32^2$), followed by the COCO evaluation format. Both recall and precision are considered during the calculation of the AP, which takes the average value of the precision rates at each recall point over a range from 0 to 1. Precision is the ratio at which the original object is detected accurately, and recall is the proportion of labeled objects in the image that are detected correctly. AP to AP$_{75}$ considers accuracy from the perspective of IoU, and AP$_{\mathrm{S}}$ to AP$_{\mathrm{L}}$ evaluates model performance from the scale diversity of objects.

When compared with the original convolution, the theoretical speed-up ratio of the Ghost module is given by

$$\begin{aligned} r_s &= \frac{n \cdot h' \cdot w' \cdot c \cdot k \cdot k}{\frac{n}{s} \cdot h' \cdot w' \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot h' \cdot w' \cdot d \cdot d} \\ &= \frac{c \cdot k \cdot k}{\frac{1}{s} \cdot c \cdot k \cdot k + \frac{s-1}{s} \cdot d \cdot d} \approx \frac{s \cdot c}{s + c - 1} \approx s \end{aligned} \tag{12}$$

where $d \times d$ has a similar magnitude to $k \times k$, and $s \ll c$. Here, $k \times k$ represents the convolutional kernel size, h' and w' give the height and width of the output data, respectively, $d \times d$ represents the linear operation kernel size, and N is the channel number of the feature maps. The Ghost module has 1 identity mapping and $m\cdot(s-1) = \frac{n}{s}\cdot(s-1)$ linear operations. In this study, we used $d = 3$ and $s = 2$ in the following experiments for both the effectiveness and efficiency.

B. UNIT VALIDATION EXPERIMENT

The experimental parameters are presented in this section. All experiments in this study used the same computational hardware to demonstrate the performance of the proposed method. The computer was configured using two NVIDIA TITAN V graphics cards, with a total of 24 GB of video random access

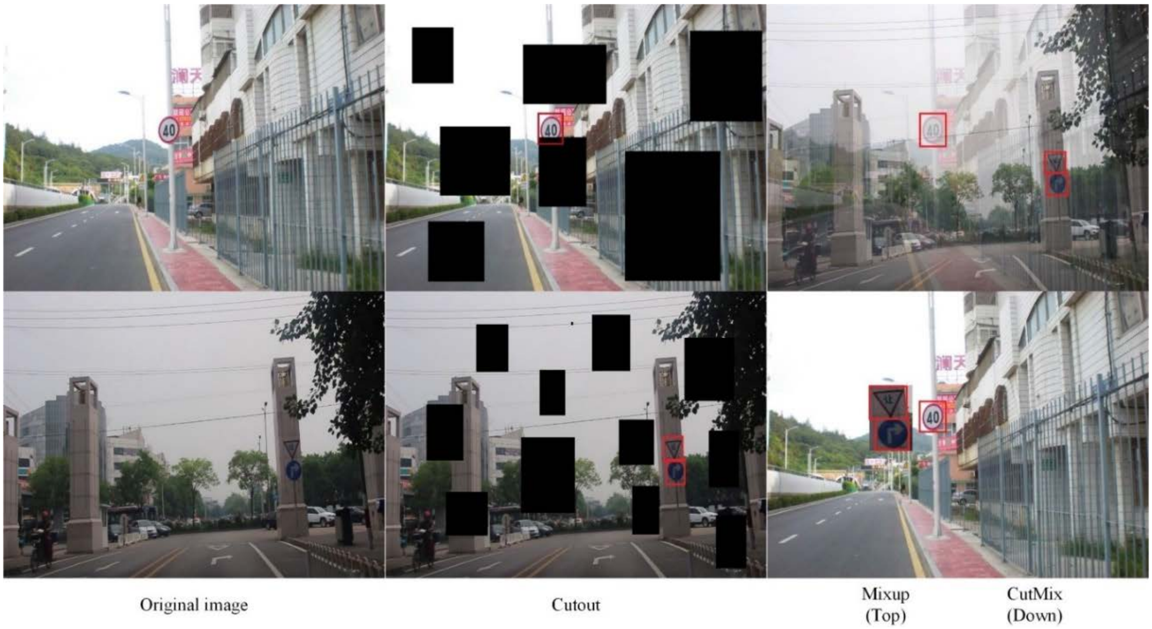


FIGURE 6. Data augmentation for traffic object. On the basis of commonly used data augmentation, Cutout, Mixup and CutMix were used to increase variety of image data.

TABLE 2. Detection results based on the BCTSDB dataset.

Model	Params(M)	FPS	mAP (%)
YOLOv3	61.9	55.8	58.5
YOLOv3-SPP	63.0	55.7	59.5
YOLOv3-GhostNet	23.49	62.5	63.8

memory (VRAM). The network structures were implemented using PyTorch. The default hyper-parameters used in the proposed method and other SOTA object detection methods are the same as those used in MMDetection [45]. The input images were resized to a maximum of 640 × 640 pixels without changing the aspect ratio. The backbone networks of the different methods were pre-trained using the ImageNet dataset. Other settings for all experiments were consistent with MMDetection unless otherwise specified. For the proposed method, the initial learning rate was set at 2.5×10^{-2}, and the warm-up ratio was set at 0.1.

To test the effectiveness of GhostNet, we replaced Darknet in YOLOv3 with GhostNet and compared the results obtained with those from the original YOLOv3 and YOLOv3-SPP. As shown in Table 2, the experimental results proved that GhostNet can provide significant improvements in terms of the number of parameters required, computational complexity, and accuracy.

The results of the comparison of the different models on the COCO-val dataset are shown in Table 3. These results show that the CIoU-aware approach with a decoupled head can improve the correlation between localization and classification, and can thus effectively improve the detection accuracy.

The score for each anchor was calculated using S_{conf}. The hyper-parameter α is used to balance the effects of classification and regression. We evaluated different α values on the COCO–val dataset. Experiments showed that the best performance was obtained at $\alpha = 0.5$. In particular, when $\alpha = 1$, this means that the classification alone is used for the confidence calculation, whereas the influence of the bounding boxes is not considered. The experimental results in Table 4 show that the performance at $\alpha = 1$ is not as high as that when α takes other values, indicating the effectiveness of the CIoU-aware method.

The results in Table 5 show that the proposed Ghost attention bottleneck can achieve a better performance than the other models in the ImageNet dataset. The visualization results for our proposed model (Ghost attention bottleneck) with a baseline (GhostNet) are illustrated in Fig. 7. The first row shows the original images of traffic signs in the BCTSDB dataset. The second row shows the visualization results for the baseline, and the third row shows the visualization results for the proposed model. The figure clearly shows that the Ghost attention bottleneck can cover the object region to be detected and provide a better performance than the baseline model.

Ablation experiments are performed to verify the effectiveness of the proposed module. The CIoU-aware decoupled head, data augmentation, and anchor-free model were gradually added to the YOLOv3-GhostNet 1.1x baseline. The same parameters and training schemes were used in each ablation experiment. The ablation results for the COCO dataset are listed in Table 6. Owing to factors such as density and small objects on a single image in the dataset, most detection algorithms achieve low accuracy on the COCO dataset. In the

TABLE 3. Comparison of the different methods on the COCO-val dataset.

Model	Head	Backbone	AP (%)	AP_{50} (%)	AP_{75} (%)	FPS
YOLOv3	Coupled	DarkNet-53	21.6	44.0	19.2	56
RetinaNet	Decoupled	ResNet-50-FPN	35.9	55.8	38.4	52
CIoU-aware RetinaNet	CIoU-aware Decoupled head	ResNet-50-FPN	37.1	56.3	40.5	52

TABLE 4. Effectiveness of different coefficients on the COCO-val dataset.

α	AP (%)	AP_{50} (%)	AP_{75} (%)	AP_S (%)	AP_M (%)	AP_L (%)
1.0	34.9	52.2	37.4	16.7	38.4	46.8
0.7	35.8	53.1	38.5	18.1	39.9	49.5
0.5	36.0	52.8	39.0	18.4	40.2	50.0
0.3	35.9	51.6	39.2	18.2	40.1	50.2

TABLE 5. Classification results with different attention module when using the light-weight network on ImageNet[24] dataset.

Model	Attention	Parameters (M)	FLOPs(M)	Top-1 Acc. (%)	Top-5 Acc. (%)
MobileNetV1 0.7×	None	2.30	283	65.1	86.3
MobileNetV1 0.7×	SE	2.71	283	67.5	87.5
MobileNetV1 0.7×	CBAM	2.71	289	68.5	88.5
GhostNet 1.0×	None	5.08	141	71.5	89.7
GhostNet 1.0×	SE	5.20	141	73.9	91.4
GhostNet 1.0×	CBAM	5.20	147	75.1	91.9

TABLE 6. Ablation experiments on COCO val dataset.

Methods	Parameters (M)	FLOPs (G)	AP (%)	$AR^{max=10}$ (%)	AR^{medium} (%)
YOLOv3-GhostNet 1.1x	3.74	6.76	23.7	28.9	32.6
+ CIoU aware decoupled head	4.6 (+0.86)	9.66(+2.9)	25.8(+2.1)	32.8(+3.9)	35.5(+2.9)
+anchor-free	4.46 (-0.14)	8.96(-0.7)	26.4(+0.6)	37.3(+4.5)	42.4(+6.9)
+data augmentation	4.46 (+0)	8.96(+0)	28.8(+2.4)	39.1(+1.8)	44.2(+1.8)

table, AR is the average recall because we used the COCO format to evaluate different methods. $AR^{max=10}$ means AR given 10 detections per image, and AR^{medium} means AR for medium objects ($32^2 < \text{area} < 96^2$). As can be seen from the results listed in the table, the component CIoU-aware decoupled head, anchor-free model, and data augmentation improved the AP by 2.1%, 0.6%, and 2.4%; $AR^{max=10}$ by 3.9%, 4.5%, and 1.8%; and AR^{medium} by 2.9%, 6.9%, and 1.8%, respectively.

TABLE 7. Experimental results on COCO val dataset.

Model	AP (%)	Params. (M)	FLOPs (G)	FPS
YoloV3-Tiny	16.6	8.86	5.62	64
YoloV4-Tiny	21.7	6.96	6.06	51
YOLOX-Tiny	32.8	5.06	6.45	59
YOLOv5-s	36.7	7.30	17.1	62
YOLOv3-MobileNetV2	30.1	3.74	6.76	68
Ours.	33.2	6.95	5.97	72

C. OVERALL VERIFICATION EXPERIMENT

To verify the generalization of the model, we compared the performance of the proposed model with those of other models on the COCO-val dataset, as presented in Table 7, which lists the detection results based on the YOLO series. The experiments demonstrated that our model remains competitive on common datasets. Compared with YOLOv3-MobileNetV2, our methods can improve the detection accuracy by 3.1% while increasing the detection speed because of fewer computations. Although our method is 3.5% lower than YOLOv5-s in terms of accuracy (mean AP or mAP), it offers advantages in terms of both the parameters and computation.

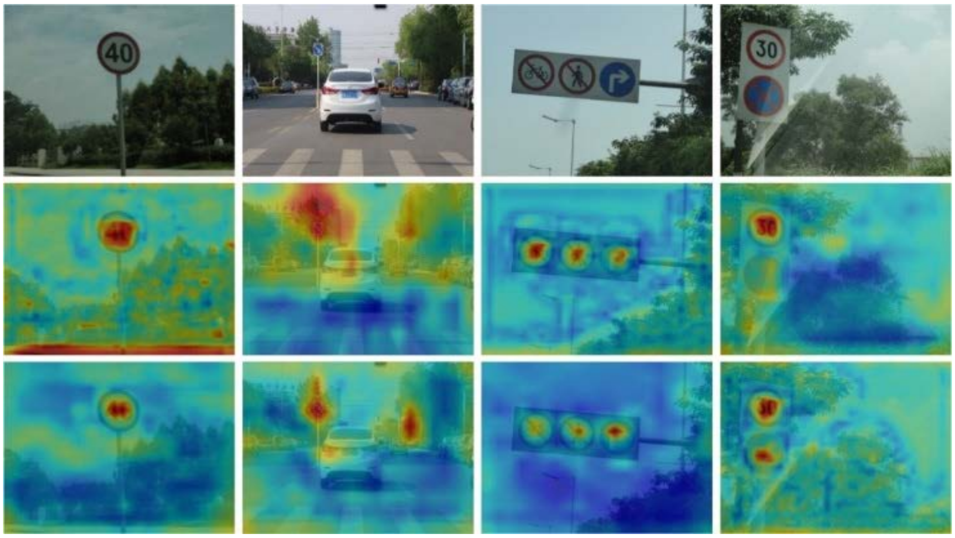


FIGURE 7. Visualization of feature maps. The first row shows the original images. The second row shows the visualization results for GhostNet and the third row shows the visualization results for our proposed model.

TABLE 8. Experimental results on BCTSDB val dataset.

Model	Backbone	Params. (M)	FLOPs (G)	AP (%)	AP$_{50}$ (%)	AP$_{75}$ (%)	AP$_S$ (%)	AP$_M$ (%)	AP$_L$ (%)	FPS
Faster R-CNN[11]	ResNet50	60.52M	121.84	70.2	94.7	86.0	65.3	76.5	84.5	28
Cascade R-CNN[12]	ResNet50	68.93M	118.81	75.8	96.7	92.5	72.9	79.3	89.2	23
RetinaNet[8]	ResNet50	37.74M	95.66	59.7	89.4	71.2	47.2	72.5	83.3	52
YOLOv3[6]	Darknet53	61.95M	78.22	59.5	92.7	70.4	54.2	70.1	83.8	56
FCOS	ResNet50	31.84M	78.67	68.6	95.8	83.9	62.7	75.7	83.9	61
YOLOV5-m	CSPDarknet	21.2M	49.0	79.3	99.1	95.2	-	-	-	-
YOLOX-s[46]	CSPDarknet	9.0M	26.8	77.2	95.7	91.6	75.4	79.6	85.4	53
YOLOv3[6]	Mobilenetv2 x1.0	3.74M	6.76	50.5	90.2	51.4	43.8	63.8	75.8	68
YOLOX-s[46]	Mobilenetv2 x1.0	3.91M	5.87	75.2	96.3	91.4	73.7	77.3	83.7	63
Ours.	Improved GhostNet	6.95M	5.97	78.6	98.7	94.5	75.1	82.6	88.4	72

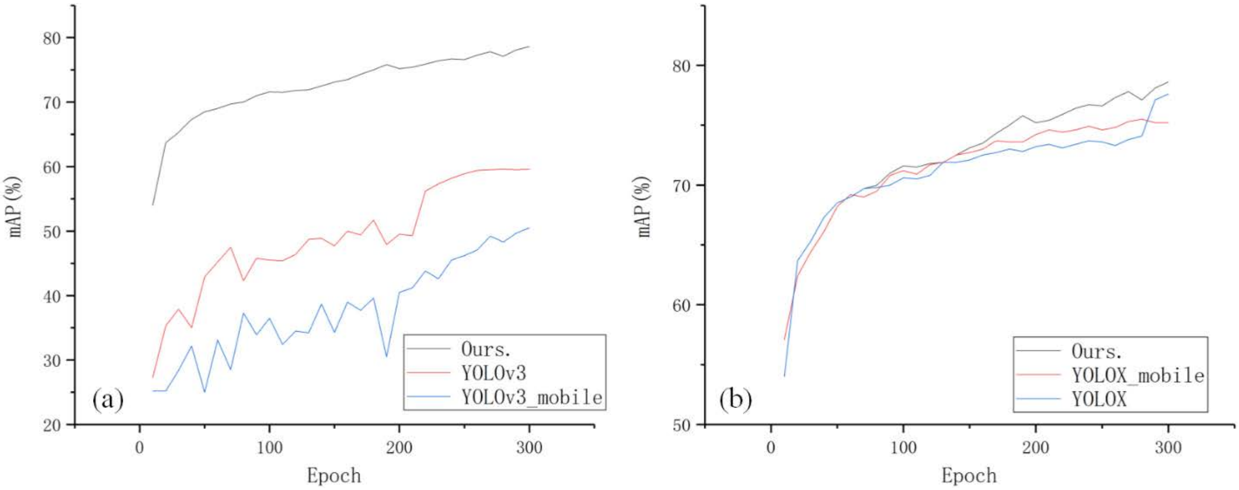


FIGURE 8. mAP curves for the different methods on the BCTSDB dataset. The figure shows that the proposed algorithm has high accuracy and can converge faster.

Specifically, the computational cost of our method is one-third of that of YOLOv5-s, which means that the proposed method is more competitive in lightweight object detection applications.

The results of the comparisons between the performances of the different methods are presented in Table 8, which lists the detection results that include those for the multi-stage algorithms (faster R-CNN, cascade R-CNN) and the single-stage algorithms (RetinaNet, FCOS, YOLOv5, YOLOv3, YOLOX). Our method achieved high detection accuracy with fewer parameters, which yielded more competitive results, with AP, AP50, and AP75 values of 78.6%, 98.7%, and

TABLE 9. Experimental results on KITTI val dataset.

Methods	Car			Pedestrian			Cyclist			Time (ms)
	Easy(%)	Moderate(%)	Hard(%)	Easy(%)	Moderate(%)	Hard(%)	Easy(%)	Moderate(%)	Hard(%)	
Regionlets[47]	84.75	76.45	59.70	73.14	61.15	55.21	70.41	58.72	51.83	-
Faster R-CNN[11]	87.97	79.11	70.62	78.97	65.24	60.09	71.40	61.86	53.97	142
Sensekitti[48]	94.79	93.17	84.38	82.72	68.41	62.72	82.90	73.48	64.03	-
Mono3D[49]	84.52	89.37	79.15	80.30	67.29	62.23	77.19	65.15	57.88	-
MS-CNN[50]	93.87	88.68	76.11	85.71	74.89	68.99	84.88	75.30	65.27	-
SSD[7]	87.34	87.74	77.27	50.38	48.41	43.46	48.25	52.31	52.13	30
YOLOv3[6]	84.37	77.69	75.62	82.58	76.29	73.36	85.14	80.07	77.65	28
ASSD[51]	89.28	89.95	82.11	69.07	62.49	60.18	75.23	76.16	72.83	30
RFBNet[52]	87.31	87.27	84.44	66.16	61.77	58.04	74.89	72.05	71.01	23
Ours.	94.48	91.03	82.05	85.32	78.31	76.29	83.04	81.54	78.76	19

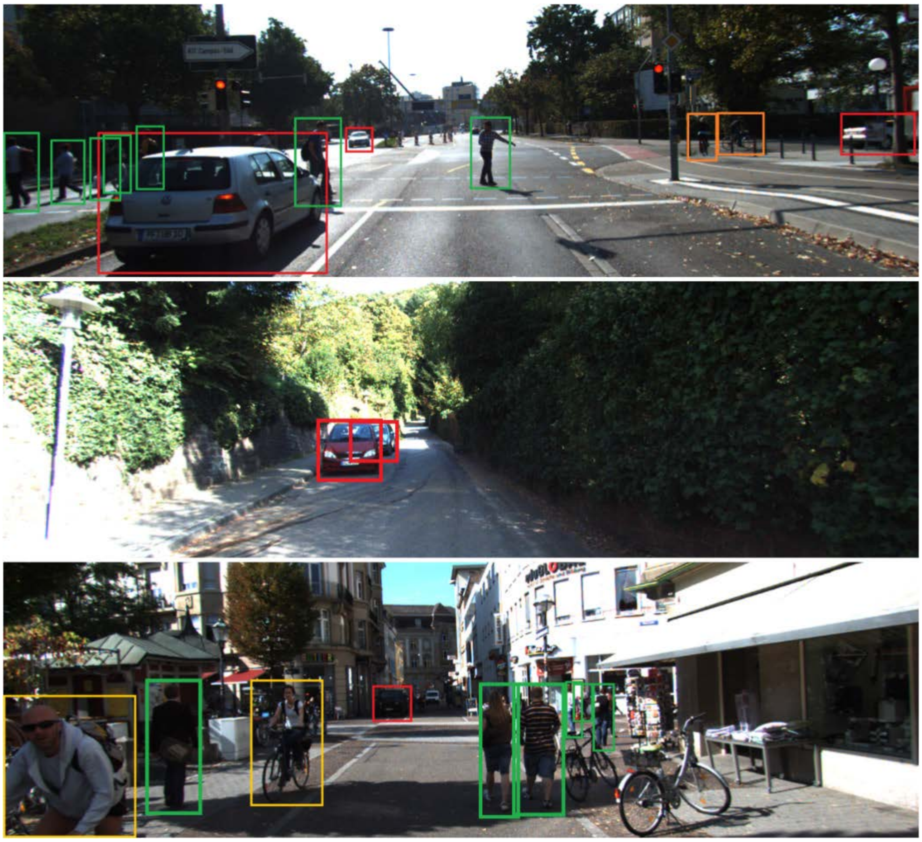


FIGURE 9. Detection Results on KITTI dataset. The above figure shows the proposed algorithm accurately detecting objects in a traffic scene on the KITTI object detection dataset.

94.5%, respectively. A comparison with generic multi-stage networks and single-stage networks shows that our method has the advantages of low computational requirements and a low number of parameters that allow it to overcome the memory limitations in the autonomous driving field. Compared with some lightweight networks, including the original YOLOX-s, YOLOv3, and YOLOX with the MobileNetv2 lightweight backbone network, our method can obtain higher detection accuracy. It is evident from the results that our method can achieve a performance comparable to that of YOLOv5 with a low number of parameters and computations.

The mAP curves for the different methods when applied to the BCTSDB dataset are shown in Fig. 8, demonstrating that the proposed model converges more rapidly and has a higher AP value.

We also evaluate the proposed method using the KITTI dataset. As shown in Table 9, the detection accuracy of the proposed method is significantly improved compared to that

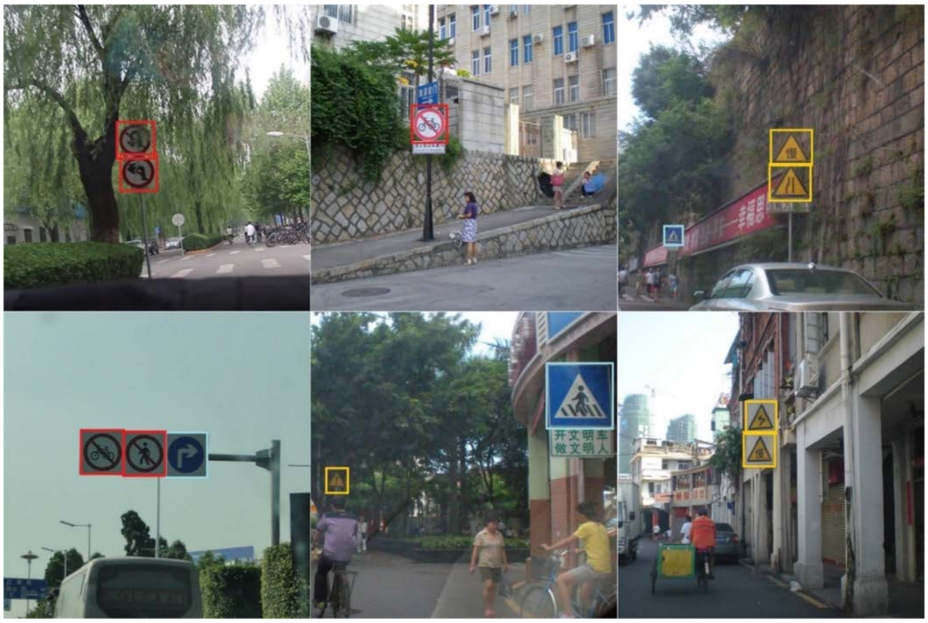


FIGURE 10. Detection Results on BCTSDB dataset. The above figure shows the proposed method can stably and accurately detect traffic signs in different backgrounds.

of the existing algorithms. The size of the image used in this part was the same as that of the original dataset.. Our method also demonstrated the shortest processing time while maintaining high detection accuracy.

Fig. 9 and 10 show the detection results obtained for the KITTI and BCTSDB datasets, respectively. The results clearly show that our method can effectively detect objects in traffic scenes.

D. DISCUSSION

In AVs, real-time performance and accuracy are two important performance indicators. This ensures that the AVs detect objects quickly and accurately and make autonomous driving safe. The main purpose of the proposed method is to improve the model in two aspects. In the unit validation section, we verified the effectiveness of the proposed module GhostNet with attention, CIoU-aware decoupled head, anchor-free, and data augmentation. The overall section compares the proposed method with a lightweight detection method on public datasets. The results show that the number of parameters of our proposed method is slightly improved, and the two important indicators of detection accuracy and real-time performance are improved, which can promote the reliability of vision-based object detection algorithms in autonomous driving systems. Although the proposed method has been improved compared to other existing methods, it also faces many problems. For example, the verification of the current algorithm is based on public datasets, whereas in actual traffic scenarios, the influence of weather, illumination, and other factors reduces the generalization ability of the detection model. It will take years to the fully automated environment, in such a mixed (AVs and human-driven vehicles) traffic scene at present, the relationship between visual perception objects among multiple vehicles is a challenging problem.

V. CONCLUSION

In this study, we propose an anchor-free lightweight object detector for autonomous driving applications. The detector can achieve a high detection accuracy and trade-off with a small-sized model. The approach incorporates an attention scheme in a lightweight neural network called GhostNet and adds an IoU branch to the anchor-free decoupled detector to rank the large number of candidate detections accurately. Data augmentation is used to enhance the robustness of the detection model in real-world scenarios. Extensive experiments on COCO, KITTI, and BCTSDB datasets verified the effectiveness of the proposed algorithm.

Furthermore, the proposed method achieved high detection accuracy when using a small size detection model; when applied to real traffic scenarios, the interference in real complex scenarios is not considered. In future work, ALODAD will be improved by the application of specific data augmentation methods or domain adaptation techniques.

A. ABBREVIATIONS

Abbreviations	Full Name
AR	Average Recall
$AR^{max=10}$	AR given 10 detections per image
AR^{medium}	AR for medium objects (32^2 <area<96^2)
AP	Averaged AP at IoUs from 0.5 to 0.95 with an interval of 0.05

AP_{50}	AP at IoU threshold 0.5
AP_{75}	AP at IoU threshold 0.75
AP_L	AP for objects of large scales (area>96^2)
AP_M	AP for objects of medium scales (32^2 <area<96^2)
AP_S	AP for objects of small scales (area<32^2)
AVs	Autonomous Vehicles
BCTSDB	BUU Chinese Traffic Sign Detection Benchmark
CBAM	Convolutional block attention model
CIoU	Complete IoU
DIoU	Distance-IoU
FLOPs	Floating-point operations per second
FPN	Feature pyramid network
GhostA	Ghost attention bottleneck
HOG	Histogram of Oriented Gradients
IoU	Intersection over union
NMS	Non-Maximum Suppression
SOTA	State-of-the-art
SSD	Single Shot multibox Detector
SVM	Support Vector Machine
VRAM	Video random access memory
YOLO	You Only Look Once

ACKNOWLEDGMENT

Authors' contributions: Conceptualization: Tianjiao Liang and Weiguo Pan; methodology: HONG BAO; software: Tianjiao Liang and Weiguo Pan; validation: Feng Pan; writing–original draft preparation: Tianjiao Liang; writing–review and editing: Tianjiao Liang and Weiguo Pan. All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] 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)*, Seattle, WA, USA, Jun. 2020, pp. 1577–1586, doi: 10.1109/CVPR42600.2020.00165.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Diego, CA, USA, vol. 1, Jun. 2005, pp. 886–893, doi: 10.1109/CVPR.2005.177.

[3] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. Appl.*, vol. 13, no. 4, pp. 18–28, Jul. 1998, doi: 10.1109/5254.708428.

[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)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[5] 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, doi: 10.1109/CVPR.2017.690.

[6] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," Apr. 2018, *arXiv:1804.02767*. Accessed: Nov. 28, 2021.

[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*, vol. 9905, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020, doi: 10.1109/TPAMI.2018.2858826.

[9] 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, doi: 10.1109/CVPR.2014.81.

[10] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[11] 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, doi: 10.1109/TPAMI.2016.2577031.

[12] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 6154–6162, doi: 10.1109/CVPR.2018.00644.

[13] F. Wang, Y. Li, Y. Wei, and H. Dong, "Improved faster RCNN for traffic sign detection," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Rhodes, Greece, Sep. 2020, pp. 1–6, doi: 10.1109/ITSC45102.2020.9294270.

[14] C. Han, G. Gao, and Y. Zhang, "Real-time small traffic sign detection with revised faster-RCNN," *Multimedia Tools Appl.*, vol. 78, no. 10, pp. 13263–13278, May 2019, doi: 10.1007/s11042-018-6428-0.

[15] X. He, R. Cheng, Z. Zheng, and Z. Wang, "Small object detection in traffic scenes based on YOLO-MXANet," *Sensors*, vol. 21, no. 21, p. 7422, Nov. 2021, doi: 10.3390/s21217422.

[16] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring R-CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 6402–6411, doi: 10.1109/CVPR.2019.00657.

[17] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Computer Vision—ECCV*, vol. 11218, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham, Switzerland: Springer, 2018, pp. 816–832, doi: 10.1007/978-3-030-01264-9_48.

[18] S. Fan, F. Zhu, S. Chen, H. Zhang, B. Tian, Y. Lv, and F.-Y. Wang, "FII-CenterNet: An anchor-free detector with foreground attention for traffic object detection," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 121–132, Jan. 2021, doi: 10.1109/TVT.2021.3049805.

[19] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," *Int. J. Comput. Vis.*, vol. 128, pp. 642–656, Mar. 2020, doi: 10.1007/s11263-019-01204-1.

[20] X. Xu, W. Liang, J. Zhao, and H. Gao, "Tiny FCOS: A lightweight anchor-free object detection algorithm for mobile scenarios," *Mobile Netw. Appl.*, vol. 26, no. 6, pp. 2219–2229, Dec. 2021, doi: 10.1007/s11036-021-01845-y.

[21] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: A simple and strong anchor-free object detector," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 1922–1933, Apr. 2022, doi: 10.1109/TPAMI.2020.3032166.

[22] C. Yao, F. Wu, H.-J. Chen, X.-L. Hao, and Y. Shen, "Traffic sign recognition using HOG-SVM and grid search," in *Proc. 12th Int. Conf. Signal Process. (ICSP)*, Hangzhou, China, Oct. 2014, pp. 962–965, doi: 10.1109/ICOSP.2014.7015147.

[23] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010, doi: 10.1109/TPAMI.2009.167.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Apr. 2015, *arXiv:1409.1556*. Accessed: Nov. 26, 2021.

[26] M. Lin, Q. Chen, and S. Yan, "Network in network," Mar. 2013, *arXiv:1312.4400*. Accessed: Dec. 1, 2021.

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[29] H. Jung, M.-K. Choi, J. Jung, J.-H. Lee, S. Kwon, and W. Y. Jung, "ResNet-based vehicle classification and localization in traffic surveillance systems," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Honolulu, HI, USA, Jul. 2017, pp. 934–940, doi: 10.1109/CVPRW.2017.129.

[30] 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," Apr. 2017, *arXiv:1704.04861*. Accessed: Nov. 22, 2021.

[31] D. Biswas, H. Su, C. Wang, A. Stevanovic, and W. Wang, "An automatic traffic density estimation using single shot detection (SSD) and MobileNet-SSD," *Phys. Chem. Earth, A/B/C*, vol. 110, pp. 176–184, Apr. 2019, doi: 10.1016/j.pce.2018.12.001.

[32] 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.*, Salt Lake City, UT, USA, Jun. 2018, pp. 6848–6856, doi: 10.1109/CVPR.2018.00716.

[33] C. Chen, L. Z. Fragonara, and A. Tsourdos, "Go wider: An efficient neural network for point cloud analysis via group convolutions," *Appl. Sci.*, vol. 10, no. 7, p. 2391, Apr. 2020, doi: 10.3390/app10072391.

[34] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Computer Vision—ECCV*, vol. 11211, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham, Switzerland: Springer, 2018, pp. 3–19, doi: 10.1007/978-3-030-01234-2_1.

[35] 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, doi: 10.1109/CVPR.2017.106.

[36] W. Wang, E. Xie, X. Song, Y. Zang, W. Wang, T. Lu, G. Yu, and C. Shen, "Efficient and accurate arbitrary-shaped text detection with pixel aggregation network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 8439–8448, doi: 10.1109/ICCV.2019.00853.

[37] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, "SOLO: Segmenting objects by locations," in *Computer Vision—ECCV*, vol. 12363, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham, Switzerland: Springer, 2020, pp. 649–665, doi: 10.1007/978-3-030-58523-5_38.

[38] 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*, Apr. 2020, vol. 34, no. 7, pp. 12993–13000, doi: 10.1609/aaai.v34i07.6999.

[39] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," Apr. 2017, *arXiv:1710.09412*. Accessed: Dec. 20, 2021.

[40] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," Nov. 2017, *arXiv:1708.04552*. Accessed: Dec. 20, 2021.

[41] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 6022–6031, doi: 10.1109/ICCV.2019.00612.

[42] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Computer Vision—ECCV*, vol. 8693, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 740–755, doi: 10.1007/978-3-319-10602-1_48.

[43] T. Liang, H. Bao, W. Pan, and F. Pan, "Traffic sign detection via improved sparse R-CNN for autonomous vehicles," *J. Adv. Transp.*, vol. 2022, pp. 1–16, Mar. 2022, doi: 10.1155/2022/3825532.

[44] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013, doi: 10.1177/0278364913491297.

[45] K. Chen *et al.*, "MMDetection: Open MMLab detection toolbox and benchmark," Jun. 2019, *arXiv:1906.07155*. Accessed: Nov. 28, 2021.

[46] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," Aug. 2021, *arXiv:2107.08430*. Accessed: Nov. 28, 2021.

[47] X. Wang, M. Yang, S. Zhu, and Y. Lin, "Regionlets for generic object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 10, pp. 2071–2084, Oct. 2015, doi: 10.1109/TPAMI.2015.2389830.

[48] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "CRAFT objects from images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 6043–6051, doi: 10.1109/CVPR.2016.650.

[49] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3D object detection for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2147–2156, doi: 10.1109/CVPR.2016.236.

[50] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Computer Vision—ECCV*, vol. 9908, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 354–370, doi: 10.1007/978-3-319-46493-0_22.

[51] J. Yi, P. Wu, and D. N. Metaxas, "ASSD: Attentive single shot multibox detector," *Comput. Vis. Image Understand.*, vol. 189, Dec. 2019, Art. no. 102827, doi: 10.1016/j.cviu.2019.102827.

[52] S. Liu, D. Huang, and Y. Wang, "Receptive field block net for accurate and fast object detection," in *Computer Vision—ECCV*, vol. 11215, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham, Switzerland: Springer, 2018, pp. 404–419, doi: 10.1007/978-3-030-01252-6_24.

TIANJIAO LIANG was born in Shandong, in 1998. He received the B.S. degree from the University of Jinan, in 2020, China. He is currently pursuing the M.S. degree in software engineering with the Beijing Union University. His research interests include computer vision, deep learning, and object detection.

HONG BAO was born in Beijing, in 1958. He received the B.S. degree in computer science from Beijing Union University, in 1983, and the Ph.D. degree in computer science from Bejing Jiaotong University, in 2012. His research interests include intelligent driving, cognitive computing, networks, and distributed systems.

WEIGUO PAN was born in Handan, Hebei, China. He received the B.S. degree from the North China University of Water Resources and Electric Power, in 2009, and the M.S. degree from Beijing Union University, in 2012, and the Ph.D. degree from the University of Chinese Academy of Sciences, in 2015. His research interests include machine learning, object detection, and intelligent driving.

FENG PAN was born in Nanjing, Jiangsu, China. He received the B.S. degree in electronic science and technology from the Nanjing University of Science and Technology, in 2000, the M.S. degree in computer science and technology from the University of Chinese Academy of Sciences, in 2008, and the Ph.D. degree in control science and engineering from the College of Information Science and Technology, Beijing University of Chemical Technology. His research interests include machine learning and intelligent driving.