electronics

Article

BiGA-YOLO: A Lightweight Object Detection Network Based on YOLOv5 for Autonomous Driving

Jun Liu [1], Qiqin Cai [1,2], Fumin Zou [1], Yintian Zhu [2,*], Lyuchao Liao [1] and Feng Guo [1,3]

[1] Fujian Key Laboratory for Automotive Electronics and Electric Drive, Fujian University of Technology, Fuzhou 350118, China; 2211301018@smail.fjut.edu.cn (J.L.); 20011080002@stu.hqu.edu.cn (Q.C.); fmzou@fjut.edu.cn (F.Z.); achao@fjut.edu.cn (L.L.); n180310004@fzu.edu.cn (F.G.)
[2] School of Mechanical Engineering and Automation, Huaqiao University, Xiamen 361021, China
[3] College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China
* Correspondence: 2201905112@smail.fjut.edu.cn

Abstract: Object detection in autonomous driving scenarios has become a popular task in recent years. Due to the high-speed movement of vehicles and the complex changes in the surrounding environment, objects of different scales need to be detected, which places high demands on the performance of the network model. Additionally, different driving devices have varying performance capabilities, and a lightweight model is needed to ensure the stable operation of devices with limited computing power. To address these challenges, we propose a lightweight network called BiGA-YOLO based on YOLOv5. We design the Ghost-Hardswish Conv module to simplify the convolution operations and incorporate spatial coordinate information into feature maps using Coordinate Attention. We also replace the PANet structure with the BiFPN structure to enhance the expression ability of features through different weights during the process of fusing multi-scale feature maps. Finally, we conducted extensive experiments on the KITTI dataset, and our BiGA-YOLO achieved a mAP@0.5 of 92.2% and a mAP@0.5:0.95 of 68.3%. Compared to the baseline model YOLOv5, our proposed model achieved improvements of 1.9% and 4.7% in mAP@0.5 and mAP@0.5:0.95, respectively, while reducing the model size by 15.7% and the computational cost by 16%. The detection speed was also increased by 6.3 FPS. Through analysis and discussion of the experimental results, we demonstrate that our proposed model is superior, achieving a balance between detection accuracy, model size, and detection speed.

Keywords: object detection; lightweight network; attention mechanism; ghost module; CA; BiFPN; YOLOv5

Citation: Liu, J.; Cai, Q.; Zou, F.; Zhu, Y.; Liao, L.; Guo, F. BiGA-YOLO: A Lightweight Object Detection Network Based on YOLOv5 for Autonomous Driving. *Electronics* **2023**, *12*, 2745. https://doi.org/10.3390/electronics12122745

Academic Editor: Hüseyin Kusetogullari

Received: 22 May 2023
Revised: 15 June 2023
Accepted: 16 June 2023
Published: 20 June 2023

1. Introduction

Autonomous driving technology is one of the most prominent fields of interest in recent years [1]. It provides people with safer, more convenient, and more efficient means of transportation. In autonomous driving technology, not only is traffic condition prediction technology [2–4] a component, but a more directly related and crucial technique is object detection. Object detection is a crucial task that helps vehicles identify and track surrounding objects, thereby achieving autonomous driving. Object detection [5] is a critical topic in computer vision that seeks to detect specific objects in images or videos. Numerous object identification techniques have been proposed by researchers over the last few decades, including feature-based methods, classifier-based methods, and regression-based methods. However, these methods have limitations such as low accuracy, slow speed, and poor performance in detecting small objects. Deep learning technology advancements in recent years have resulted in new breakthroughs in object detection.

Among these, the existing image-based object detection algorithms are typically categorized into two groups. The first group consists of two-stage object detection algorithms with candidate box generation, for example, RCNN (Region-based Convolutional Neural

Network) [6]. The algorithm first extracts candidate boxes from the image and then obtains detection results based on the candidate regions. Its characteristics are high recognition accuracy and precision, but low recognition efficiency, and it requires large computing resources, making it unsuitable for low-performance embedded devices. However, later proposed object detection algorithms, such as Fast-RCNN [7] and Faster RCNN [8], also suffer from the disadvantages of low accuracy and robustness. The second category is one-stage object detection methods, along with their representative networks, such as the SSD (Single Shot MultiBox Detector) series of detection methods [9–11], the YOLO series of detection methods [12–14], and the RetinaNet [15].

The SSD method outperforms the R-CNN algorithm in terms of detection speed, but it has limitations in detection accuracy. YOLO is a one-stage detection network that greatly surpasses other CNN (Convolutional Neural Network) models in terms of detection speed while maintaining accuracy, making it suitable for practical road scene detection methods. However, in actual deployment, it is still difficult to achieve the resource processing efficiency required for autonomous driving tasks when running the original YOLO network on embedded devices. Additionally, the original YOLO still has issues with low accuracy and robustness in road object detection.

The YOLO detection algorithm has been continuously improved in subsequent research. For improving the detection accuracy, the YOLOv2 algorithm [16] mainly uses operations such as batch normalization, high-resolution classifier classification, direct object box position detection, and multi-scale training to improve the detection accuracy of the model. YOLOv3 employs a novel Darknet-53 residual network in conjunction with a FPN (feature pyramid network) [17] for multi-scale fusion prediction based on YOLOv2. YOLOv4 fuses the backbone network with the CSPNet (Cross-Stage Hierarchical Network) algorithm [18] to ensure detection accuracy while reducing network computation. It also incorporates a feature pyramid network into the spatial pyramid pooling layer to address the issue of shallow feature loss.

YOLOv5, a popular object recognition approach, is similar to YOLOv4, except the neck network uses the FPN and the PAN (pixel aggregation network) structure [19]. YOLOv5 is extensively employed for object detection tasks across various domains, with numerous studies exploring its foundations. For instance, Zhu et al. proposed TPH-YOLOV5 [20] as an enhanced model for object recognition in UAV (unmanned aerial vehicle)-captured sceneries. This model added a prediction head to recognize objects of varying sizes, and the initial prediction head was replaced with a TPH (transformer prediction head). The introduction of the transformer module enhanced the ability to capture global information and context, resulting in an approximately 7% improvement compared to YOLOv5. Similarly, the Multi-scale YOLOv5 [21] approach enhances detection capabilities by incorporating additional detection heads, introducing the novel SPD-CONV module, and utilizing a standardized attention module, thereby improving the detection of minuscule objects in traffic environments. YOLO-FIRI [22] addresses the issue of low recognition rates in infrared images due to distance and low resolution by compressing channels and optimizing parameters, achieving outstanding results in infrared object detection. Benjumea et al.'s YOLO-Z [23] network demonstrates a 6.9% increase in mAP for detecting smaller objects in autonomous racing scenarios compared to YOLOv5s, at the cost of a 3 ms increase in inference time. Inam et al. proposed an intelligent infrastructure management framework based on a two-stage deep learning approach [24], employing the YOLOv5 model to detect cracks in bridge images, and comparing the performance of YOLOv5s, YOLOv5m, and YOLOv5L. Mahaur et al. introduced an enhanced object detection model called iS-YOLOv5 [25] for detecting small objects, such as traffic signs and traffic signal lights, in real-time driving scenarios.

However, there are still some issues with these methods. Although most methods improve detection accuracy in some cases, they have problems with large FLOPs, large model files, and slow computation speed. Some lightweight models can effectively reduce model parameters but cannot achieve a balance between accuracy and speed.

Therefore, this paper proposes an improved YOLOv5 model aimed at improving the accuracy and speed of autonomous driving object detection while lightening the model architecture. Our model achieves a better performance by introducing new convolutional module and attention mechanisms, as well as multi-scale feature fusion methods. In the experiment, we tested and evaluated our algorithm using the KITTI dataset and compared it with other object detection algorithms. The experimental results show that our algorithm performs well in both accuracy and speed. The main contributions of this paper can be summarized as follows:

- We introduce the Ghost module [26] and design the GHConv (Ghost-HardSwish Conv) module as the main convolution method to simplify the model, and integrate the CA (Coordinate Attention) [27] to capture the spatial dependency of image information in order to more effectively extract features of targets in the image.
- We integrate the BiFPN (Bi-directional Feature Pyramid Network) [28] architecture, which learns the features of different resolutions with different weights and horizontally connects them with residual structures to fuse multi-scale feature information, to improve the PANet architecture in YOLOv5.
- We propose the BiGA (Bidirectional-Ghost conv-Attention)-YOLO network by integrating the above-mentioned improvement methods, and conduct experiments on the KITTI dataset, achieving a mAP@0.5 of 92.2% and a mAP@0.5:0.95 of 68.3%. Furthermore, the model is simplified by 15.7% and detection speed is raised by 6.3 FPS, which achieves a balance among model size, detection accuracy, and speed.

The rest of this work is organized as follows. We present the related groundwork in Section 2. Section 3 introduces the overview of YOLOv5 and our BiGA-YOLO. Section 4 introduces the experiments and analyzes the results. Finally, we give the conclusions of this paper and prospective future research work. For the reader's convenience, we have included some of the main abbreviations, and their corresponding full forms, used in this paper in Table 1.

Table 1. The main abbreviations and their corresponding full names.

Abbreviation	Fullform
CNN	Convolutional Neural Network
RCNN	Region-based Convolutional Neural Network
SSD	Single Shot MultiBox Detector
FPN	Feature Pyramid Network
PAN	Pixel Aggregation Network
CA	Coordinate Attention
CBAM	Convolutional Block Attention Module
BiFPN	Bi-directional Feature Pyramid Network
mAP	Mean Average Precision
FLOPs	Floating Point Operations

2. Related Works

2.1. CNN-Based Object Detection

Deep learning has become a commonly used technique for object detection due to the rapid growth of CNN. They are classified as one-stage detectors or two-stage detectors based on whether or not there are region proposals. Faster-RCNN and Cascade RCNN [29] are two-stage detectors that produce region suggestions from the input image and then feed them into the network for classification and regression. YOLO detectors abandon region proposals to meet real-time detection requirements, showing advantages in speed and accuracy, and providing effective methods for object detection applications in various fields. To eliminate the limitations of anchors and pursue a larger and more flexible solution space, anchor-free detectors have been proposed, including CenterNet [30], FCOS (Fully Convolutional One-Stage Object Detection) [31] and Foveabox [32]. Recently, the ARSL (Adversarial Residual Semi-Supervised Learning) algorithm [33], proposed for the

ambiguity of semi-supervised object detection, has provided a new direction for the research of one-stage object detection.

2.2. Attention Mechanism

Attention mechanisms play an important role in autonomous driving object detection research. The core idea of this mechanism is to assign different weights to input features, allowing the model to focus on the most influential parts for predicting results. Attention mechanisms [34] were first proposed by Bahdanau et al. in 2014 to solve the problem of long-distance dependencies in neural machine translation tasks. Since then, attention mechanisms have achieved significant success in the computer vision field, especially in object detection and image classification tasks.

In the realm of computer vision, there are now several typical attention techniques. The first is spatial attention, which modulates the attention of each point in the feature map to help the model focus more on specific areas, such as the STN (Spatial Transformer Network) [35]. The second type of attention is channel attention, such as the SENet (Squeeze-and-Excitation Network) and ECA (Efficient Channel Attention) [36], which allocate resources on each convolutional channel and adjust the single dimension of the z-axis. The third is spatial and channel integration, such as the CBAM (Convolutional Block Attention Module) [37], and CA, which is used in this paper. CA can capture cross-channel information while incorporating direction and position-aware information, which helps the model to more accurately locate and recognize the objects of interest.

2.3. Model Lightweight

In the context of autonomous driving, real-time performance and computational resource limitations are critical factors. Therefore, lightweight models have significant importance in autonomous driving object detection. Lightweight models aim to reduce the number of model parameters and computational complexity, thereby reducing computational resource requirements and improving inference speed.

To achieve a lightweight YOLO algorithm, various strategies can be used. One method is to use lightweight CNN architectures, such as MobileNet [38], ShuffleNet [39], or EfficientNet [40]. MobileNet adopts depthwise separable convolution to minimize the amount of parameters and computations. ShuffleNet achieves its light weight by group convolution and channel shuffle. EfficientNet was proposed by Tan and Le in 2019, which adapts the network's depth, width, and resolution to achieve an efficient light weight.

Another method is to use network pruning techniques to decrease the model size by removing unimportant neurons or connections. The representative work of this method is Deep Compression [41], as proposed by Han et al. in 2015. In addition, model quantization is also an effective lightweight strategy, which reduces computational resource requirements by lowering the precision of weights and activation values. The representative work of this method is XNOR-Net [42], proposed by Rastegari et al. in 2016.

By combining attention mechanisms and lightweight model strategies, the YOLO algorithm can be effectively improved, making it higher performing in real-time autonomous driving object detection tasks.

3. Method

3.1. Overview and Key Components of YOLOv5

Like popular single-stage detection models, YOLOv5 has strong feature extraction capabilities, fast detection speed, and high accuracy. The YOLOv5 series provides four model scales: YOLOv5-S, YOLOv5-M, YOLOv5-L, and YOLOv5-X, where S stands for small, M for medium, L for large, and X for extra large. The network structure of these models is unchanged, but the modules and convolution kernels are scaled proportionally, which changes the complexity and size of each model. In this paper, we studied the basic network architecture of YOLOv5s because it is more balanced with regard to model

size and performance, making it suitable for use in object detection scenarios such as autonomous driving.

Figure 1 illustrates the basic architecture of YOLOv5s, which is organized into four major parts: input, backbone, neck, and output. In the following sections, we will briefly introduce these parts and their key modules.

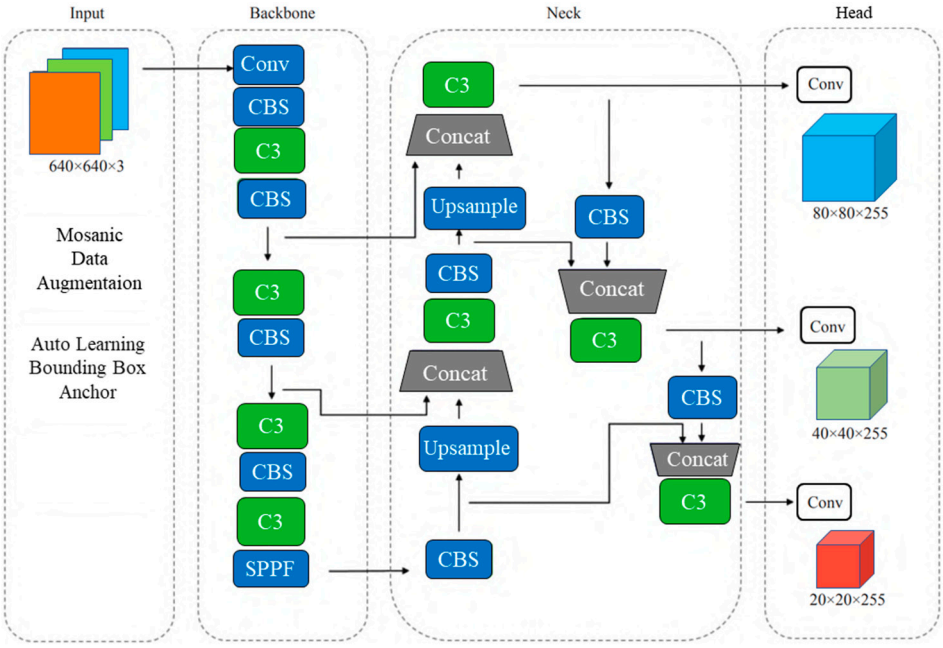

Figure 1. The architecture of the YOLOv5s.

3.1.1. Input

In YOLOv5, the input performs adaptive image padding and Mosaic data augmentation on the image data. The input end also integrates adaptive anchor box calculation, which allows the model to obtain the initial anchor box size by clustering the label boxes of the dataset, without the need for manual anchor box parameter settings before training on different datasets. Mosaic is a data augmentation method based on Cutmix [43]. In Cutmix, two images are combined, while Mosaic combines four training images into one for training. In the previous YOLO series, prior box scales were extracted through clustering, while YOLOv5 embeds the adaptive anchor box calculation function into the code, which adaptively calculates the best anchor point box based on the dataset during each training.

3.1.2. Backbone

The backbone is the basic feature extraction network of YOLOv5, responsible for extracting meaningful features from the input image. YOLOv5 uses CSPDarknet53 as its backbone, which is an improved network based on Darknet53 that introduces the concept of CSPNet. CSPNet divides the feature map into two parts and fuses them at different stages, which helps to improve the feature expression ability and reduce computational costs. It mainly consists of CBS modules and C3 groups, where the CBS module consists of a convolutional layer, a batch norm layer, and a SiLU activation function, while C3 is composed of three CBS and multiple bottlenecks. The SPPF (Spatial Pyramid Pooling Fusion) module principle is basically the same as spatial pyramid pooling. The architecture diagrams of these modules are shown in Figure 2.

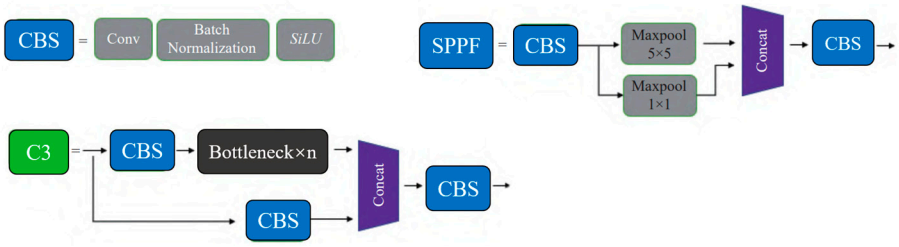

Figure 2. The architecture of components in YOLOv5.

3.1.3. Neck

The neck section of YOLOv5 is the feature pyramid network responsible for fusing multi-scale features from the backbone. YOLOv5's neck adopts the PANet structure, which utilizes a bottom-up path aggregation to enhance the flow of information. PANet adaptively fuses feature maps of various levels using an attention mechanism, thus improving the performance of object detection.

PANet consists of multiple upsampling and downsampling modules, as well as lateral connections. The upsampling module magnifies the feature maps to higher resolutions using bilinear interpolation or transposed convolution, while the downsampling module reduces the feature maps to lower resolutions using max pooling or convolution. The lateral connections transmit information between feature maps of different scales to achieve feature fusion.

3.1.4. Head

The head of YOLOv5 is responsible for generating the object detection results based on the fused feature maps. YOLOv5's head consists of multiple output layers, with each layer responsible for detecting objects of different scales. YOLOv5 utilizes an anchor-based approach, where a set of anchor boxes with predefined scales and aspect ratios are defined beforehand. During training, the network learns to predict the target class, position offsets, and scale changes for each anchor box.

The head includes multiple convolutional layers and activation functions. The convolutional layers are used to generate the prediction results, including the target class, position offsets, and scale changes. Activation functions, such as Sigmoid and Leaky ReLU, are used for non-linear transformations, helping to improve the network's expression ability. During inference, YOLOv5 uses NMS (Non-Maximum Suppression) to eliminate overlapping detection boxes, resulting in the final object detection results.

In summary, YOLOv5 employs the Mosaic data augmentation, the CSPDark-net53 network, the PANet feature pyramid network, and the anchor-based object detection strategy. These key modules and methods collectively ensure YOLOv5's excellent performance as a popular object detector.

3.2. BiGA-YOLO

With reduced computational complexity, we propose a lightweight detection network model based on YOLOv5s, named BiGA-YOLO, to improve the accuracy and speed of object detection. The naming of BiGA-YOLO is primarily based on the abbreviations of the innovative modules and features improved in our network. "Bi" represents bidirectional information flow, indicating the enhanced expressive power of our network through the BiFPN module. "G" represents Ghost conv, and "A" represents our CA attention mechanism. The suffix "YOLO" indicates that our base network is part of the YOLO series. Figure 3 depicts the structure of the improved BiGA-YOLO, which uses GHConv to optimize the overall computational performance of the network and reduce the computational cost. To achieve a loss in accuracy, we introduce the CA attention group to form the CAblock, which is used to extract image information from the previous convolution and allocate different channel weights. The purpose of the CA module is to enhance the network's feature extraction ability by highlighting key information of the detection object,

thereby improving the accuracy of detecting targets in various scenarios. Finally, we use BIFPN to weight and fuse the corresponding features of the image feature maps with different resolutions. Our BiGA-YOLO network achieves a good balance in object detection map, speed, and model lightweight.

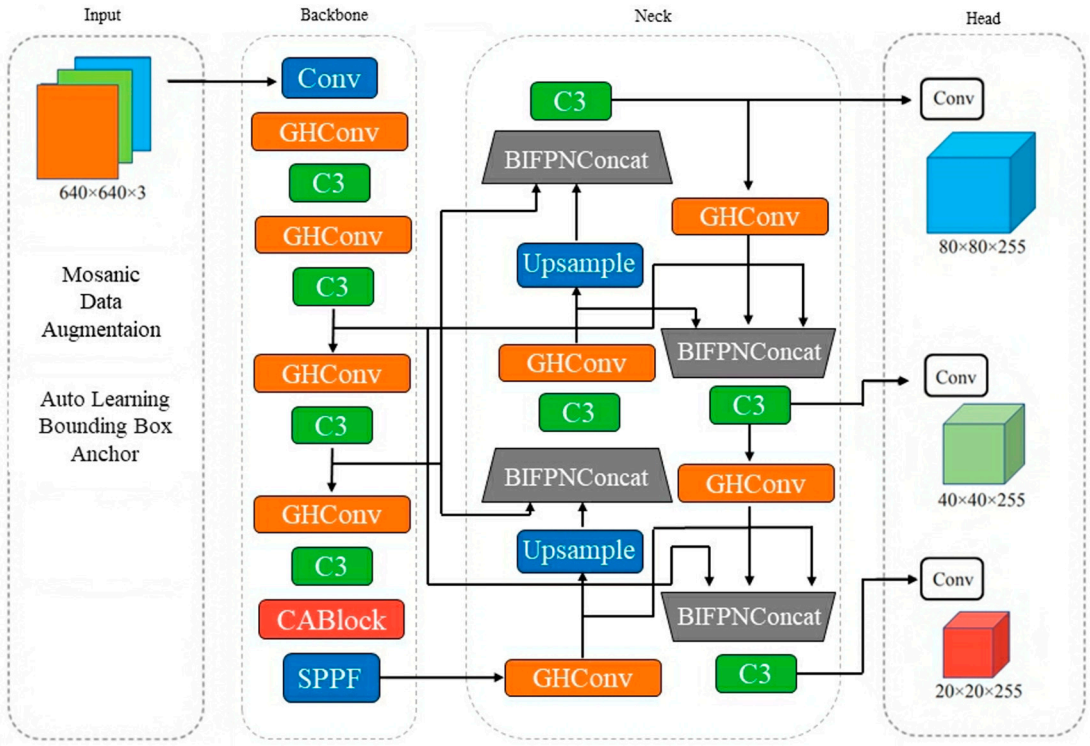

Figure 3. The architecture of BiGA-YOLO network.

3.2.1. GHConv Module

The author of GhostNet, Kai Han, found that some of the feature maps generated after convolution are similar. Therefore, he suggested that we can reduce the computational cost of feature map generation by using inexpensive linear transformations. The two sets of feature maps are then concatenated together to form the desired feature map. This structure uses standard convolution and point convolution for channel reduction.

GhostNet is a lightweight convolutional network that can reduce network computational costs while scarcely reducing detection accuracy. The network designs a Ghost module, which can increase the number of feature maps while reducing parameters. From the Figure 4, we can see that the core idea of this module is to divide the original convolution operation into two stages. The first stage is a conventional convolution calculation, where the number of convolution kernels needs to be controlled to avoid an increase in the model's parameter size. The second stage is a cheap feature map operation. In this stage, it performs another linear convolution operation on the intermediate feature maps generated in the first stage, thereby generating a large number of feature maps.

In the calculating process, assuming the size of the input feature map is $h \times w \times c$, where c represents the number of channels, and h and w represent the height and width, respectively, the calculation formula of n generated feature maps for any convolutional layer is shown in Equation (1).

$$Y = X * f + b \tag{1}$$

where $*$ denotes convolution operation, b represents bias, Y represents a feature map of size $h' \times w'$, f is the convolution filter, and $k \times k$ is the kernel size of the convolution filter f. The

number of FLOPs in the calculation process is $n \cdot h' \cdot w' \cdot c \cdot k \cdot k$. The theoretical acceleration ratio of the Ghost model compared to conventional convolution can be calculated as:

$$\begin{aligned} r_s &= \frac{n \cdot h' \cdot w' \cdot c \cdot k \cdot k}{\frac{n}{s} \cdot h' \cdot w' \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot h' \cdot w' \cdot d \cdot d} \\ &= \frac{c \cdot k \cdot k}{\frac{1}{s} \cdot c \cdot k \cdot k + \frac{s-1}{s} \cdot d \cdot d} \approx \frac{s \cdot c}{s + c - 1} \approx s \end{aligned} \tag{2}$$

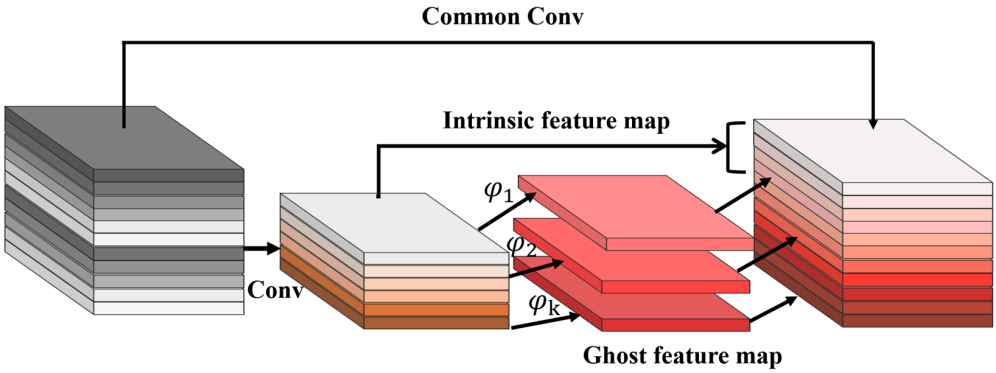

Figure 4. Ordinary convolution and Ghost module.

The GHConv module designed in this paper employs the HardSwish activation function in place of the original ReLU activation function in Ghost module, which effectively alleviates the gradient vanishing problem that exists in the convolutional layer caused by the ReLU activation function and avoids the phenomenon of gradient saturation. As shown in Figure 5, the negative half-axis of the ReLU activation function is 0, which results in the inability to activate the negative half-axis, making it impossible for neurons to learn effective features. On the other hand, the HardSwish function retains the characteristics of the Swish function without an upper bound and with a lower bound, and replaces the exponential operation of the Swish function with an approximate function, achieving similar activation effects as the Swish activation function at a lower computational cost. From the perspective of computational resources, it is more suitable for deployment on embedded devices and real-time object detection scenarios, making it more suitable as the activation function of the Ghost module in this paper. As shown in Figure 3, we integrate the GHConv module into various parts of the backbone and neck, and use it as a convolutional layer in the BiGA-YOLO architecture.

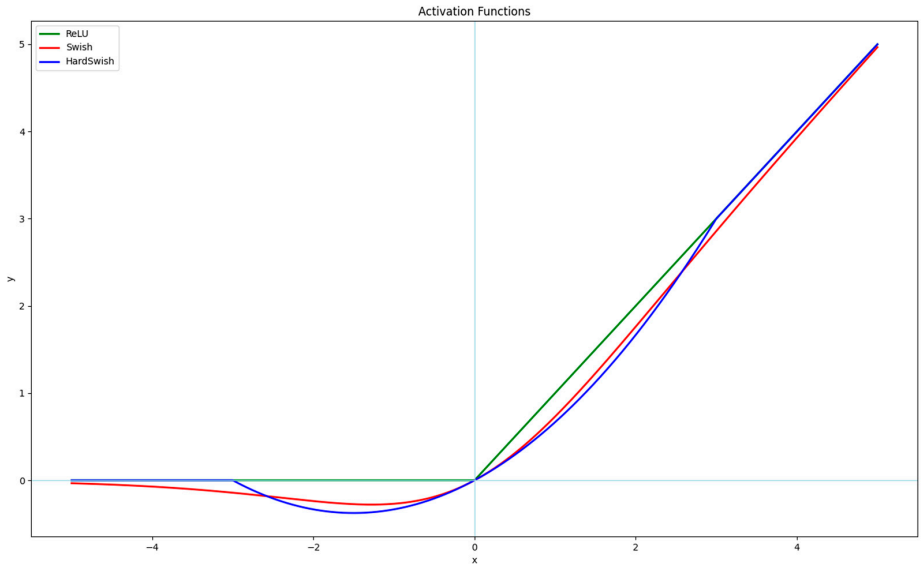

Figure 5. The activation function curve.

3.2.2. CA Module

The CA mechanism was proposed for the reason that the channel attention mechanism ignored important positional information, while CA retained positional information to better capture features in the image. The structure of the CA module is shown in Figure 6. CA decomposes channel attention into two 1D feature encodings and aggregates features along two spatial directions. This way, remote dependencies can be captured along one spatial direction while accurate positional information can be retained along the other spatial direction. The generated feature maps are encoded into a pair of direction-aware and position-sensitive attention maps, which complementarily enhance the representation of attention objects in the input feature map. Therefore, the CA attention mechanism not only considers channel information but also considers direction-related positional information, which can adaptively learn the importance of each pixel in the image. Moreover, the computational cost of the CA module is less than that of traditional attention mechanisms. Therefore, we use the CA Block as part of the improved network. As shown in Figure 3, our CABlock is placed before the SPPF module in the backbone of the BiGA-YOLO network. By preserving direction-related positional information through CABlock, the representation ability of object features is enhanced.

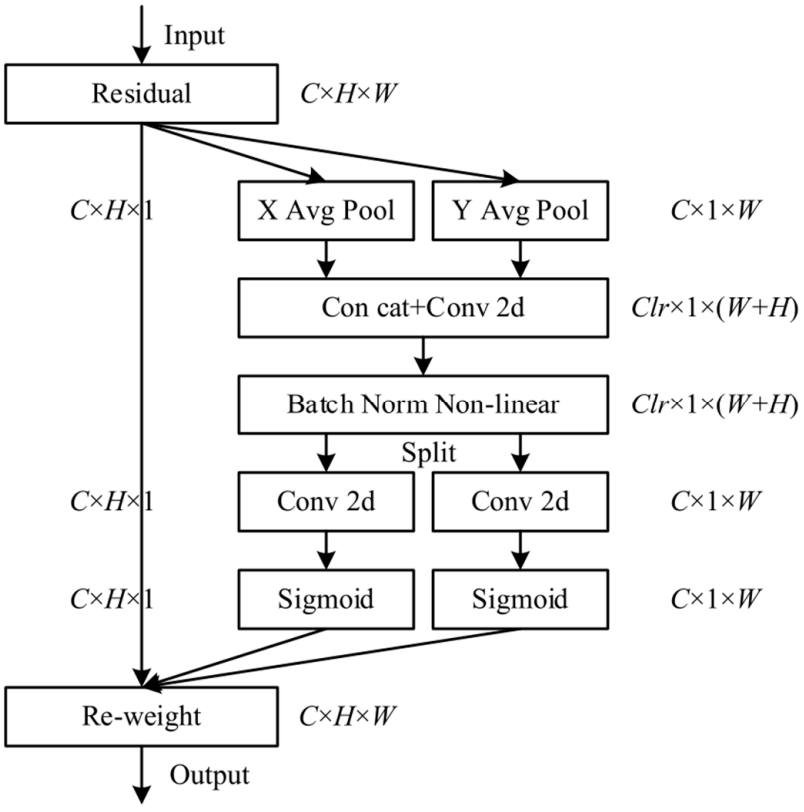

Figure 6. CA schematic.

3.2.3. BiFPN

Extracting and fusing effective information from multi-scale features is crucial for object detection, as complex and diverse detection targets exhibit significantly different features across different scales. Based on deep learning, CNNs can extract low-level and high-level features from high-resolution images, and then we need to fuse these multi-scale features in the network. Although PANet in YOLOv5 can also achieve different feature fusion through upsampling and downsampling, it has a large computational cost. In contrast, the BiFPN used in EfficientDet is a typical complex bidirectional feature fusion FPN structure. The comparison of the two structures is shown in Figure 7. BiFPN removes the two intermediate nodes of the highest and lowest-level feature layers that enter the FPN

structure on the basis of traditional bidirectional feature fusion FPN, and adds a residual edge map connecting the input features and each feature layer in the middle of the output feature map.

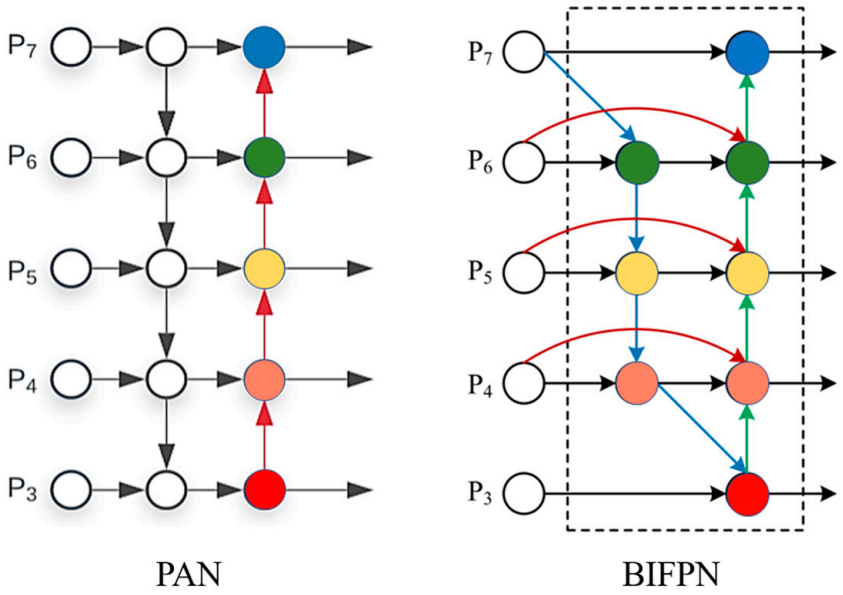

Figure 7. The comparison of PANet and BiFPN.

Since the size of the detected targets in different images is different, features of different resolutions will be generated during training. PANet essentially and simply adds different features, which will lead to unequal weights for different-sized features of the same type in the fused output features. Since the contribution weights of different input features to the output feature map should be different at each node during feature fusion, BiFPN introduces a weight to adjust the contribution of different inputs to the output feature map during training, enhancing the expression ability of features. During training, at each feature fusion node, the input feature map will obtain the weight that achieves the best performance of the object detection algorithm. Overall, BiFPN achieves simple and fast multi-scale feature fusion, simplifying the structure of FPN to some extent. In this paper, we integrate the BiFPN architecture into BiGA-YOLO, as shown in Figure 3. The integrated BiFPN architecture is located in the neck part of BiGA-YOLO and fuses upsampled feature maps with feature maps obtained through downsampling via a residual structure. Additionally, the expression ability of each feature part is adjusted by weights to obtain the best feature representation for object detection after training.

4. Experiments

In this section, we will introduce the data preprocessing, experimental settings, and analysis of experimental results to verify the effectiveness and superiority of the proposed model in this paper.

4.1. Experimental Introduction

4.1.1. Data Description

We employed the KITTI [44] traffic object dataset as the source for network testing. The dataset was jointly created by the Karlsruhe Institute of Technology in Germany and the Toyota Technical Institute in the United States. It contains real traffic images captured by in-vehicle cameras and sensors in various complex traffic scenarios, such as city roads, highways, and campuses. The dataset is suitable for use in computer vision tasks such as object detection, optical flow, and 3D tracking. The dataset labels are subdivided into several categories, including Car, Van, Truck, Pedestrian, Person_sitting, Cyclist, Tram,

Misc, and Dontcare. This paper selects the 2D object detection dataset, which includes 7481 images. As moving targets in traffic environments are mostly vehicles and pedestrians, and to reduce the detection difficulty of lightweight models, vehicle types and pedestrian poses are not further categorized. The original dataset labels are reorganized, where "Van", "Truck", and "Tram" labels are merged, and "Person_sitting" labels are merged into the "Car" and "Pedestrian" classes. The "Misc" and "Dontcare" categories are ignored, and the final selected labels for detection are "Pedestrian", "Cyclist", and "Car". The KITTI dataset with annotated images is randomly divided into a training set (5985 images), a validation set (753 images), and a test set (753 images) in an 8:1:1 ratio.

4.1.2. Implementation and Settings

The experiment was conducted on a server equipped with NVIDIA A10, with the experimental environment consisting of CentOS 7.6, Python 3.8, Pytorch 1.10.0, and CUDA 11.2. In the training phase, a stochastic gradient descent optimizer was used for training. The learning rate was initialized to 1×10^{-2} using cosine lr schedule, the momentum factor was set to 0.90, the weight decay was set to 0.0005, and the training epoch was set to 300.

The resolution ratio of the experimental images was uniformly set to (640, 640). To increase the diversity of samples and improve the performance of the network, Mosaic image data augmentation (scaling, image flipping, mosaic, and mixing) was used for training.

4.1.3. Evaluation Metrics

To evaluate the performance of the proposed model, mAP@0.5, mAP@0.5:0.95, model weights, FLOPs, and FPS were used as performance metrics. The average precision of each category is obtained by calculating the area under the precision curve and recall curve; precision and recall are defined by Equations (3) and (4), respectively:

$$P = \frac{TP}{TP + FP} \tag{3}$$

$$R = \frac{TP}{FN + FP} \tag{4}$$

where TP denotes the number of true positive examples, FP denotes the number of false positive examples, FN denotes the number of false negative examples, and TN denotes the number of true negative examples.

The calculation formulas for mAP@0.5 and mAP@0.5:0.95 are as follows:

$$AP = \int_0^1 P(R)dR \tag{5}$$

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{6}$$

where mAP@0.5 represents the average AP across all categories at an IoU of 0.5, and mAP@0.5:0.95 represents the average mAP at different IoU thresholds ranging from 0.5 to 0.95 with a step size of 0.05. mAP@0.5:0.95 can better reflects the model's generalization and robustness. FPS is used to evaluate the speed of object detection, i.e., the number of images that can be processed per second.

4.2. Results

The experimental models in this paper include YOLOv5,Fast-RCNN, SSD, YOLOv5-Mobilev3, YOLOv5-Shufflenetv2, Multi-scale YOLOv5s, and our BiGA-YOLO. For the Multi-scale YOLOv5s network, we followed the architecture proposed in the original paper. MobileNetv3 [45] is a lightweight and efficient network designed for mobile and embedded devices that uses depth-wise separable convolutions instead of regular convolutions,

significantly reducing FLOPs compared to regular convolutions. ShuffleNetv2 [46] is the latest version of the ShuffleNet series, also a lightweight network structure. This structure proposes two mechanisms: pointwise group convolution and channel shuffle, which effectively reduce computational costs while ensuring detection accuracy. Two models were obtained by replacing the backbone of YOLOv5 with the backbone networks of the above two models, and the comparison of these models better demonstrates the effectiveness of our experimental models. The experimental results on the KITTI dataset are presented in Table 2.

Table 2. Performance results on KITTI.

Model	mAP@0.5 (%)	mAP@0.5:0.95 (%)	FLOPS (G)	Weights (M)	FPS (Hz)
YOLOv5s	90.3	63.6	16.5	14.0	108.6
Fast-RCNN	76.9	44.9	223.4	160.1	70.5
SSD	74.3	42.2	93.57	131.0	101.3
YOLOv5s-MobileNetv3	87.6	57.8	6.3	7.2	106.3
YOLOv5s-Shufflenetv2	88.4	59.9	8.0	7.7	110.2
Multi-scale YOLOv5s	91.1	63.9	32.1	31.8	98.7
Ours	92.2	68.3	13.8	11.8	114.9

From Table 2, it can be observed that, although the computational complexity and model size of YOLOv5 is significantly reduced by 61% and 51%, respectively, by replacing the backbone network with lightweight architectures such as MobileNetv3 and ShuffleNetv2, there is also a noticeable decrease in its model accuracy, particularly with mAP@0.5:0.95 decreasing by 5.8% and 3.7%, respectively. However, our improved model shows a reduction in computational complexity and model size by 16% and 15.7% compared to YOLOv5s, respectively, while achieving improved model accuracy and detection speed. Specifically, it shows an increase of 1.9% in mAP@0.5 and 4.7% in mAP@0.5:0.95, with a corresponding improvement in FPS by 5.8%. Table 3 shows the mAP@0.5 values of each model for different object categories in the KITTI dataset. Among them, our BiGA-YOLO achieves a remarkable 98.1% mAP in the Car category and a substantial lead in the Pedestrian category, with a slightly lower performance in the Cyclist category. By analyzing the overall results, we can identify the strengths and weaknesses of each model.

Table 3. The mAP results for various object categories on KITTI.

Model	Car	Pedestrian	Cyclist
YOLOv5s	97.2	83.6	90.1
Fast-RCNN	84.8	70.5	75.4
SSD	80.3	68.5	74.1
YOLOv5s-MobileNetv3	96.7	80.7	85.4
YOLOv5s-Shufflenetv2	96.1	82.3	86.8
Multi-scale YOLOv5s	96.5	87.3	89.5
Ours	98.1	91.5	87.0

These results demonstrate that our model achieves superior performance while being lightweight, outperforming both YOLOv5s and the aforementioned lightweight architectures. Thus, our results confirm the superiority of our proposed model. More visualizations of the training process data for BiGA-YOLO are shown in Figure 8, where we can observe that the loss curve tends to converge at around 300 epochs.

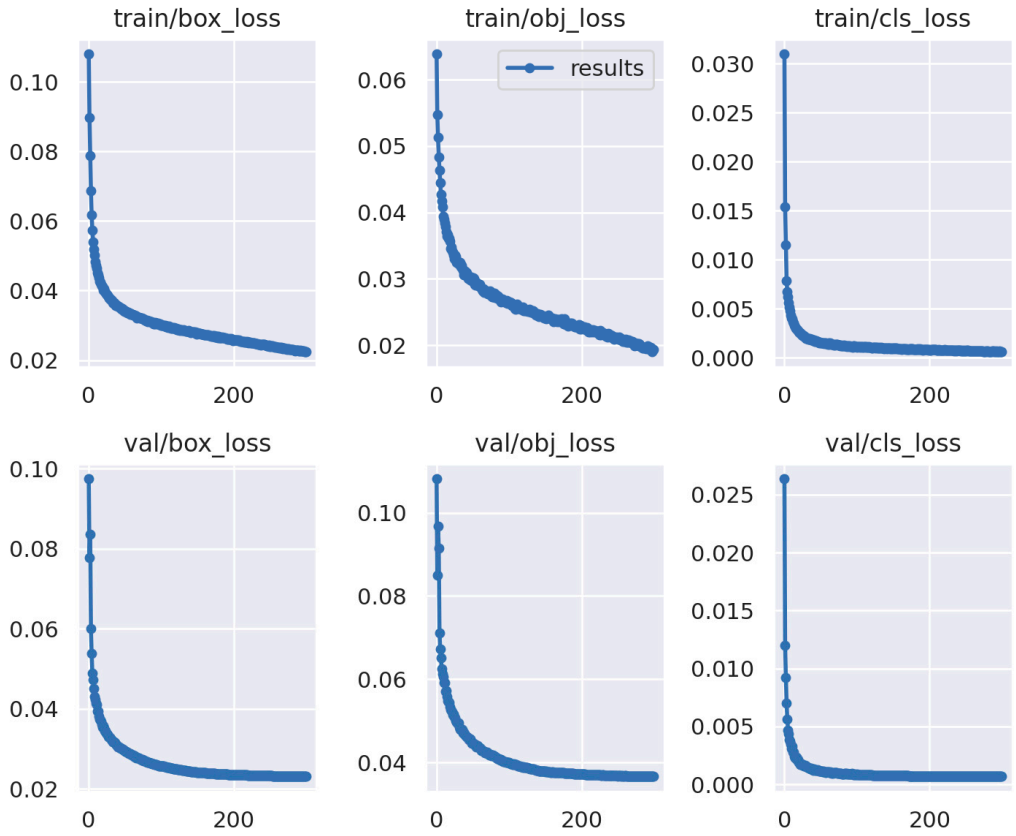

Figure 8. The training process data of BiGA-YOLO.

4.3. Comparisons

To further verify the effectiveness of the proposed model, we compared the visual effects of the network using Grad-CAM. Grad-CAM can represent the degree of attention of the network to the input image information through a heatmap. After inputting the two networks into Grad-CAM for testing, the heatmaps of the networks' attention to target recognition are shown in Figure 9. It can be observed that the BiFPN structure of our BiGA-YOLO has a higher heat map for the detection target position area compared to the PANet structure of the original YOLOv5, and has a lower heat map for irrelevant environmental information in non-target areas. This indicates that BiFPN can better extract the feature information of the detection target to some extent while reducing the attention to irrelevant information in the environment, which is reliable.

We compared the visual effects of the experimental results with the detection results of the YOLOv5s model. As shown in Figure 10, the first row shows the visual effect of the detection result of YOLOv5s, and the second row shows the visual effect of our proposed BiGA-YOLO. It can be clearly seen that our proposed BiGA-YOLO has significantly improved the confidence level of the detection boxes compared to YOLOv5s. Specifically, our proposed model has significantly improved the confidence level of the detection boxes for the three categories of targets, including cars, pedestrians, and cyclists.

Furthermore, we visualized the detection results of the model for the three targets of cars, pedestrians, and cyclists in multiple different road scenarios, as shown in Figure 11. To fully demonstrate the effectiveness of the model, we selected scenes with more complex road conditions for visualization, and each scene contains multiple categories of detection targets. From the visual effect, it can be seen that, in the detection of the three categories of targets, the confidence level of the detection boxes generated by our proposed BiGA-YOLO can reach over 0.9, which proves the effectiveness and reliability of BiGA-YOLO.

(**a**)

(**b**)

Figure 9. The heatmap of two models. (**a**) The heatmap of the YOLOv5; (**b**) The heatmap of BiGA-YOLO.

Figure 10. Comparison of the detection visualization results between YOLOv5 and BiGA-YOLO.

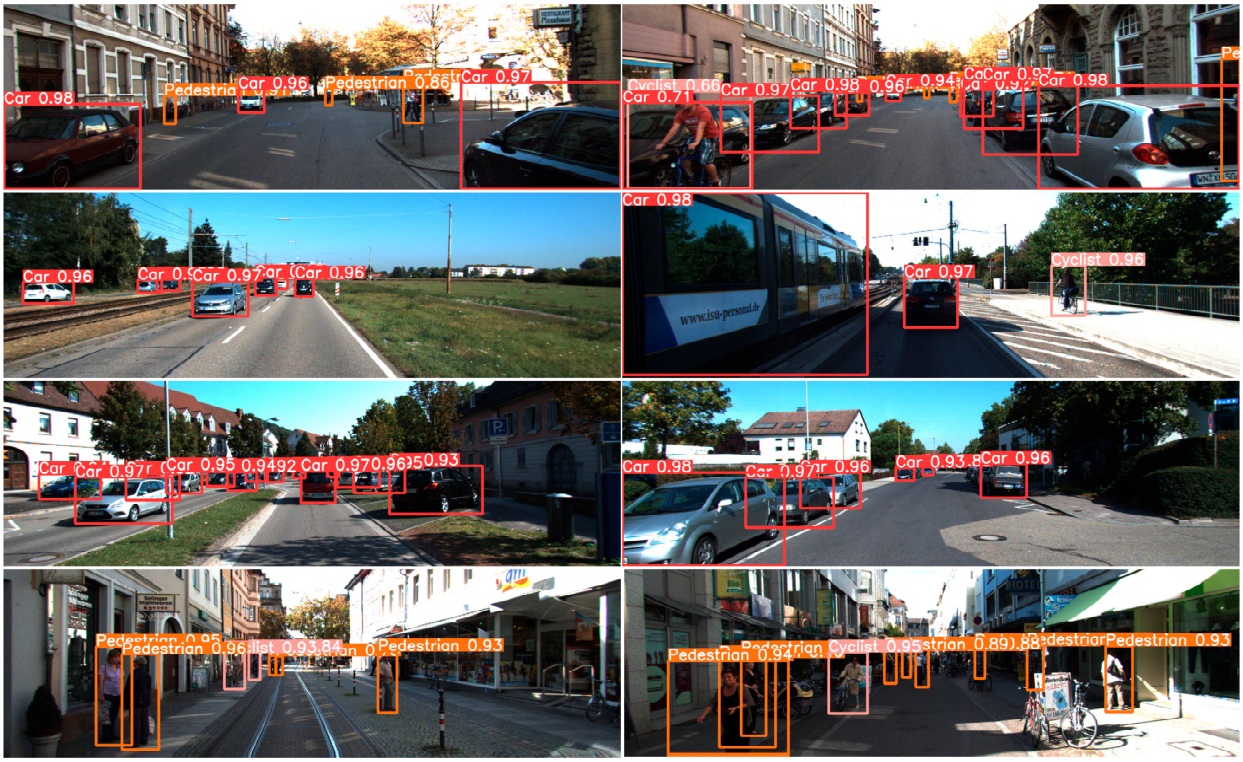

Figure 11. The detection visualization results in different scenarios of BiGA-YOLO.

4.4. Ablation Study

To analyze the differences in the combinations of components in the BiGA-YOLO model, we conducted ablation experiments, and the results are shown in Table 4. We gradually added the GHConv, CA, and BIFPN components. It can be observed that, although using the Ghost module alone and using the improved YOLOv5 with GHConv and CA architectures show a reduction in computational complexity and model size, there is also a slight decrease in model detection accuracy, especially with mAP@0.5:0.95 decreasing by 3.2% and 2.1%, respectively. Finally, the results of our improved model outperform all other models in every aspect, demonstrating the superiority of the proposed architecture that integrates these three modules. This architecture achieves a certain balance between improving detection accuracy and maintaining lightweight models in two aspects.

Table 4. Comparison of ablation experiment results.

Model	mAP@0.5 (%)	mAP@0.5:0.95 (%)	FLOPS (G)	Weights (M)	FPS (Hz)
YOLOv5s	90.3	63.6	16.5	14.0	108.6
YOLOv5s + GHConv	89.6	60.4	12.3	11.3	110.4
YOLOv5s + GHConv + CA	89.4	61.5	12.6	11.4	110.1
Ours	92.2	68.3	13.8	11.8	114.9

4.5. Discussion

From Tables 2 and 3, it can be seen that compared to other YOLOv5s variants, such as YOLOv5s-MobileNetv3, YOLOv5s-ShuffleNetv2, and Multi-scale YOLOv5s, our BiGA-YOLO demonstrates superiority in detection accuracy, speed, and model complexity. Multi-scale YOLOv5s slightly outperforms the baseline YOLOv5s by 0.8% in mAP@0.5, but falls behind in computational complexity and detection speed. Although it improves detection accuracy, it comes at the cost of substantial computational overhead and model size. In comparison to our proposed network, our architecture offers advantages in detection accuracy and speed while significantly simplifying computational complexity and model

size. These results indicate that our proposed network architecture maintains a certain level of superiority among other YOLOv5 variants in existing research, providing a valuable reference for future studies on YOLOv5 in the object detection field.

5. Conclusions

In this paper, an improved model based on YOLOv5 called BiGA-YOLO is proposed, which integrates GHConv, CA, and BiFPN. The model can effectively improve the accuracy and speed of object detection while maintaining a lightweight architecture and a certain degree of robustness. Our GHConv module can obtain feature maps at a lower computational cost, CA can better consider spatial information, and BiFPN can better fuse multi-scale object features through learnable weights. BiGA-YOLO improves the detection accuracy of the overall model architecture, simplifies the model, reduces the requirements for platform storage and computing resources, and makes it more suitable for object detection in embedded devices and autonomous driving scenarios. At the same time, our proposed method has certain limitations. It may not perform well in indoor object detection and, like other detectors, its performance may decline in environments with weak lighting or adverse weather conditions. In the future, our work can be extended to achieve good object detection performance under complex road conditions and different lighting conditions.

Author Contributions: Conceptualization, J.L., Q.C. and F.Z.; methodology, J.L.; software, J.L., Q.C. and F.G.; validation, J.L., Q.C. and F.Z.; formal analysis, J.L. and F.Z.; investigation, Q.C. and Y.Z.; resources, F.Z.; data curation, J.L., L.L. and Y.Z.; writing—original draft preparation, J.L. and Q.C.; writing—review and editing, J.L., Q.C., F.Z. and L.L.; visualization, J.L., Q.C. and F.G.; supervision, F.Z. and L.L.; project administration, F.Z. and L.L.; funding acquisition, F.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is funded by the National Natural Science Foundation of China (41971340), the 2020 Fujian Province "the Belt and Road" Technology Innovation Platform (2020D002), the Provincial Candidates for the Hundred, Thousand and Ten Thousand Talent of Fujian (GY-Z19113), the Municipal Science and Technology project (GY-Z22006, GY-Z220230), the Open fund project (KF-X1902, KF-19-22001), the Patent Grant Project (GY-Z20074), and the Crosswise project (GY-H-21021, GY-H-20077).

Data Availability Statement: Restrictions apply to the availability of these data. Data were obtained from public datasets KITTI.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yurtsever, E.; Lambert, J.; Carballo, A.; Takeda, K. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access* **2020**, *8*, 58443–58469. [CrossRef]
2. Liao, L.; Li, B.; Zou, F.; Huang, D. MFGCN: A Multimodal Fusion Graph Convolutional Network for Online Car-hailing Demand Prediction. *IEEE Intell. Syst.* **2023**, *38*, 21–30. [CrossRef]
3. Liao, L.; Hu, Z.; Zheng, Y.; Bi, S.; Zou, F.; Qiu, H.; Zhang, M. An improved dynamic Chebyshev graph convolution network for traffic flow prediction with spatial-temporal attention. *Appl. Intell.* **2022**, *52*, 16104–16116. [CrossRef]
4. Liao, L.; Hu, Z.; Hsu, C.-Y.; Su, J. Fourier Graph Convolution Network for Time Series Prediction. *Mathematics* **2023**, *11*, 1649. [CrossRef]
5. Szegedy, C.; Toshev, A.; Erhan, D. Deep neural networks for object detection. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 10–15.
6. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
7. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
8. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 21–23. [CrossRef]
9. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *Proceeding of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Proceedings, Part I 14; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
10. Fu, C.Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. Dssd: Deconvolutional single shot detector. *arXiv* **2017**, arXiv:1701.06659.

11. Cui, L.; Ma, R.; Lv, P.; Jiang, X.; Gao, Z.; Zhou, B.; Xu, M. MDSSD: Multi-scale deconvolutional single shot detector for small objects. *arXiv* **2018**, arXiv:1805.07009. [CrossRef]
12. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
13. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
14. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
15. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
16. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
17. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
18. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
19. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
20. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788.
21. Li, A.; Sun, S.; Zhang, Z.; Feng, M.; Wu, C.; Li, W. A Multi-Scale Traffic Object Detection Algorithm for Road Scenes Based on Improved YOLOv5. *Electronics* **2023**, *12*, 878. [CrossRef]
22. Li, S.; Li, Y.; Li, Y.; Li, M.; Xu, X. Yolo-firi: Improved yolov5 for infrared image object detection. *IEEE Access* **2021**, *9*, 141861–141875. [CrossRef]
23. Benjumea, A.; Teeti, I.; Cuzzolin, F.; Bradley, A. YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles. *arXiv* **2021**, arXiv:2112.11798.
24. Inam, H.; Islam, N.U.; Akram, M.U.; Ullah, F. Smart and Automated Infrastructure Management: A Deep Learning Approach for Crack Detection in Bridge Images. *Sustainability* **2023**, *15*, 1866. [CrossRef]
25. Mahaur, B.; Mishra, K.K. Small-object detection based on YOLOv5 in autonomous driving systems. *Pattern Recognit. Lett.* **2023**, *168*, 115–122. [CrossRef]
26. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.
27. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
28. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
29. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
30. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6569–6578.
31. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9627–9636.
32. Kong, T.; Sun, F.; Liu, H.; Jiang, Y.; Li, L.; Shi, J. Foveabox: Beyound anchor-based object detection. *IEEE Trans. Image Process.* **2020**, *29*, 7389–7398. [CrossRef]
33. Liu, C.; Zhang, W.; Lin, X.; Zhang, W.; Tan, X.; Han, J.; Li, X.; Ding, E.; Wang, J. Ambiguity-Resistant Semi-Supervised Learning for Dense Object Detection. *arXiv* **2023**, arXiv:2303.14960.
34. Chorowski, J.K.; Bahdanau, D.; Serdyuk, D.; Cho, K.; Bengio, Y. Attention-based models for speech recognition. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 18–20.
35. Jaderberg, M.; Simonyan, K.; Zisserman, A. Spatial transformer networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 25–29.
36. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542.
37. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
38. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
39. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.

40. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
41. Han, S.; Mao, H.; Dally, W.J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv* **2015**, arXiv:1510.00149.
42. Rastegari, M.; Ordonez, V.; Redmon, J.; Farhadi, A. Xnor-net: Imagenet classification using binary convolutional neural networks. In *Proceeding of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Proceedings, Part IV; Springer International Publishing: Cham, Switzerland, 2016; pp. 525–542.
43. Yun, S.; Han, D.; Oh, S.J.; Choe, J.; Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6023–6032.
44. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
45. Howard, A.; Sandler, M.; Chen, B.; Wang, W.; Chen, L.C.; Tan, M.; Chu, G.; Vasudevan, V.; Zhu, Y.; Pang, R.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
46. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.