electronics

Article

Applying Image Analysis to Build a Lightweight System for Blind Obstacles Detecting of Intelligent Wheelchairs

Jiachen Du [1], Shenghui Zhao [2,*], Cuijuan Shang [2] and Yinong Chen [3]

[1] School of Computer Science and Engineering, Anhui University of Science and Technology, Huainan 232000, China
[2] School of Computer and Information Engineering, Chuzhou University, Chuzhou 233100, China
[3] Computer Science and Engineering, Arizona State University, Tempe, AZ 85287, USA
* Correspondence: zsh@chzu.edu.cn

Abstract: Intelligent wheelchair blind spot obstacle detection is an important issue for semi-enclosed special environments in elderly communities. However, the LiDAR- and 3D-point-cloud-based solutions are expensive, complex to deploy, and require significant computing resources and time. This paper proposed an improved YOLOV5 lightweight obstacle detection model, named GC-YOLO, and built an obstacle dataset that consists of incomplete target images captured in the blind spot view of the smart wheelchair. The feature extraction operations are simplified in the backbone and neck sections of GC-YOLO. The backbone network uses GhostConv in the GhostNet network to replace the ordinary convolution in the original feature extraction network, reducing the model size. Meanwhile, the CoordAttention is applied, aiming to reduce the loss of location information caused by GhostConv. Further, the neck stem section uses a combination module of the lighter SE Attention module and the GhostConv module to enhance the feature extraction capability. The experimental results show that the proposed GC-YOLO outperforms the YOLO5 in terms of model parameters, GFLOPS and F1. Compared with the YOLO5, the number of model parameters and GFLOPS are reduced by 38% and 49.7%, respectively. Additionally, the F1 of the proposed GC-YOLO is improved by 10% on the PASCAL VOC dataset. Moreover, the proposed GC-YOLO achieved mAP of 90% on the custom dataset.

Keywords: target detection; YOLOV5s; attention mechanism; lightweighting

Citation: Du, J.; Zhao, S.; Shang, C.; Chen, Y. Applying Image Analysis to Build a Lightweight System for Blind Obstacles Detecting of Intelligent Wheelchairs. *Electronics* **2023**, *12*, 4472. https://doi.org/10.3390/electronics12214472

Academic Editor: Arkaitz Zubiaga

Received: 24 August 2023
Revised: 25 October 2023
Accepted: 26 October 2023
Published: 31 October 2023

1. Introduction

With the aging of the population, senior care communities have become an important place to meet the needs of the elderly in their daily lives. The use of wheelchairs among the elderly and individuals with physical disabilities is increasing. The cited paper [1] provides a statistical test of wheelchair usage. However, due to the deterioration of the elderly's physical functions and diminished perceptual abilities, the barriers on both sides of the wheelchair can pose a number of safety risks when using a wheelchair. Smart wheelchairs offer significant assistance to individuals in such situations. To improve performance, they distinguish the gaze of electric wheelchair passengers by introducing the distance between objects in the visual field as a new feature vector. This addition helps to reduce errors caused by unintentional gaze [2]. When image recognition is combined with the wheelchair control system, facial visual recognition is primarily employed to govern the movement direction of intelligent wheelchairs [3]. In terms of smart care, remote intelligent systems are designed to provide care [4,5]. They have achieved significant progress, and this paper focuses primarily on the impact of the blind spots on both sides of the wheelchair, emphasizing the areas not easily perceived by vision. To ensure the safety and comfortable use of smart wheelchairs for the elderly in senior care communities, the detection of dangerous obstacles in the blind zones on both sides of the smart wheelchair has become an important task. In target detection algorithms across various fields, diverse sensor-based methods, including lidar, millimeter-wave radar, and ultrasonic radar sensors, are employed to address different scenarios. Employing the 3D point cloud encoding of lidar for the purpose of detection [6], which is characterized by high computational complexity and data sparsity, and has limitations on small mobile devices; 3D target detection from LiDAR data for autonomous driving has shown good performance in 3D vision detection such as autonomous driving [7], but has the limiting issues of high cost and complexity of deployment on lightweight mobile devices, which can be a challenge for real-time applications or resource-limited devices. In regard to model lightweight design [8], the guide dog robot realizes traffic light and motion target detection based on the actual scene requirements using the MobileNet algorithm. The algorithm's lightweight advantage is effectively utilized to address the problem, highlighting the importance of lightness on mobile devices. The deep learning target detection algorithm used in this paper has made significant contributions to the field of computer vision, providing an effective method for solving the problem of detecting safety hazards in a wheelchair's blind field of view. This algorithm, which is based on deep learning and designed to detect targets, analyzes real-time visual information around wheelchair users, helping the elderly avoid accidentally hitting dangerous obstacles in the blind zones on either side of the wheelchair, such as dogs, cats, potholes, and human bodies that are incompletely represented at low angles, such as feet, legs, and wheels. These targets were gathered to construct a custom dataset. Collaborative annotation and video data management tools can be utilized for curation purposes [9]. In this paper, the approach to handling video data involves processing it frame by frame, resulting in an image dataset. When used in these areas, the following issues must be addressed.

- Target Specificity. The targets displayed on both sides of the wheelchair are incomplete. In the case of oversized targets, only part of the target's feature map is captured, such as feet, legs, and wheels.
- Model lightweighting issues. Adapting to resource-constrained environments and meeting the needs of resource-constrained devices.
- The issue of heavy performance loss caused by lightweighting. Losing model performance while lightweighting is difficult to balance.

Aiming at the above problems, this paper mainly starts from three parts, the first is to collect the target information under the low view angle to form the unique dataset, the second in the model is to go through the Ghost [10] module to obtain the sparse feature maps, and at the same time, utilize the CoordAtt [11] attention to obtain the channel information and the position information, and finally, the two parts of the features will be integrated, then, through the residual block adjustment in the neck part, the channel information of the feature map is enhanced by the SE [12] attention, and more features are captured to compensate for the feature loss problem caused by the convolution in the GhostNet idea, and a richer feature output is obtained through the residual connection, finally, trained on the PASCAL VOC dataset, the number of model parameters is nearly 3/5 of the original, and the GFLOPS are equivalent to 1/2 of the original, with almost the same detection time, but the overall accuracy and F1 value are significantly improved.

2. Related Work

Target detection models generally have a complex network structure and a large number of parameters, resulting in slow operation, large memory occupancy, and power consumption on low-end devices when the model is deployed on the mobile side. To solve these problems, in recent years, researchers have proposed that the study of lightweight target detection models focuses on two aspects: one is a lightweight model based on the network structure, and the other is based on some special tricks to reduce the computational and parametric quantities of the model, so that it can be efficiently operated on low-end devices.

Network-based lightweight modeling is a common technique used to compress lightweight models. Among these, the most seminal lightweight model designs involve a deep learning approach for MMR utilizing the SqueezeNet V1.1 architecture [13], while SqueezeNet mainly uses the Fire module to reduce the computation and parameter size of the network, MobileNets (V1 [14], V2 [15], V3 [16]) proposed by the Google team, which uses DSConv [17] (Depthwise Separable Convolution depth-separable convolution) and lightweight bottleneck structure to reduce the computation and number of parameters, ShuffleNet [18], proposed by the Megvii Inc (Face++) team, introduces a channel rearrangement mechanism to accelerate the convolutional computation, and EfficientNet [19], published by Google, uses a composite scaling factor to balance the depth, width, and resolution of the network, and GhostNet [10], proposed by Noah is Ark Lab, Huawei Technologies, introduces a Ghost module to enhance the feature representation capability, etc. These networks are usually designed as lightweight networks that focus on reducing the number of parameters and computational complexity to achieve efficient inference, but they all suffer from common limitations in terms of reduced detection performance, inability to adequately capture complex features in the image, especially in complex scenes or with small targets, lower performance in terms of accuracy, and possible limitations in terms of multiscale detection that does not fully exploit multiscale information.

Based on some special techniques to reduce the computational and parametric sizes of the model, which include model pruning, quantization, distillation, and so on. Among them, model pruning can reduce the computation and parameter count of the model by deleting unimportant connections, for example, the indoor target detection task, Zhang et al. used a specific channel pruning strategy in the YOLOv3 model to achieve up to 40% computational compression [20], but it also relies on the training model, which is not very suitable for the scenarios that require retraining. The authors of the [21] used a block perforation pruning method to achieve a 14 × compression rate with 49.0 mAP for YOLOv 4. However, the implementation needs to be adapted to different hardware architectures and device characteristics, and may face limitations in terms of computational resources, memory, and power consumption on older or low-end mobile devices. Quantization can reduce the storage and computational overhead of the model by reducing the bit width of the weights and activation values, for example, for the target detection task, Liu et al. [22] used 8-bit quantization in Faster R-CNN, which reduced both the storage and computation to one-fourth of the original, but it requires more complex training processes and optimization techniques, and may be more complicated to implement and debug. Moreover, knowledge distillation can effectively enhance the performance of compact models, the Adaptive Reinforcement Supervised Distillation (ARSD) framework to improve the recognition of lightweight models [23], but it requires a large model to be used as a baseline, which may require more computational resources and training time.

Different lightweight models have their own advantages and disadvantages. Network-based models usually have better speed, but may require more storage space. Model reduction based on some special tricks can greatly reduce the storage space and computation of the model, but it may have an impact on the accuracy of the model, the model of distillation technique can obtain a smaller model without decreasing the accuracy, but it requires larger computational resources to train the large model. Hence, it is crucial to achieve a balance between model lightweighting and model performance, and flight delays can be predicted using the ECA-MobileNetV3 algorithm [24], which balances model performance and weight by improving the feature extraction capability of the lightweight algorithm through the attention module. This module has been reported as performing well in the paper. This paper addresses the aforementioned issues by focusing on wheelchair blind obstacle detection in the elderly community environment. It combines both the advantages and disadvantages of the model to prioritize performance, reduce model complexity, and enhance the feature extraction through a better attention mechanism. The aim is to strike a balance between model compression, detection accuracy, and speed to comprehensively improve the model's performance.

3. Questions and Methods

3.1. Problem Description

In the semi-enclosed environment of an elderly community, real-time obstacle detection for the blind spots of mobile intelligent wheelchairs is achieved. Traditional algorithms often require substantial computational resources, which makes them unsuitable for mobile devices with limited resources, secondly, in much of the current lightweight research, the reduction in model size often comes at the cost of decreased model performance. This trade-off makes it challenging to achieve a balance between detection accuracy and model lightweightness. Therefore, there is a need to design a lightweight target detection algorithm that can perform the target detection task quickly and accurately on mobile devices.

The goal of this thesis is to design a lightweight target detection algorithm with high detection performance for smart wheelchair devices. In this paper, we will explore deep-learning-based target detection algorithms and reduce the complexity and computational overhead of the algorithms by optimizing the network structure, reducing model parameters, and using quantization techniques. Meanwhile, in this paper, the detection accuracy and speed of the algorithm will be evaluated in experiments and compared with existing lightweight target detection algorithms. Specifically, our research will cover the following issues:

- Reduce model parameters and computational complexity by controlling network depth and width. Design a lightweight target detection network structure suitable for mobile devices.
- While lightweighting the model, the performance of the model feature extraction is improved to compensate for the feature loss problem caused by lightweighting.
- Experimental evaluations were performed on the publicly available PASCAL VOC dataset. Targets at low viewing angles were collected to construct a custom dataset and to test the experimental effectiveness of the custom dataset.

Through the above research, an efficient, accurate and lightweight target detection algorithm for mobile devices can be proposed.

Model Quantification

To reduce the model parameters and computation for network depth and width, the model is mathematically modeled using two metrics, GFLOPS (the model's floating point operations, which denotes the amount of computation in billions of floating point operations required by the model to perform inference) and Parameters (the model's parameter count, which denotes the total number of parameters to be trained in the model). Backbone partially improves the efficiency of residual feature extraction in the C3 module, reducing computational complexity and the number of parameters. Assuming that the GFLOPS of the original Backbone with C3 module is F_{backbone} and the number of parameters is P_{backbone}, $\alpha(0 < \alpha < 1)$ is a scaling factor for reducing the computational complexity and the number of parameters, the GFLOPS and Parameters of the improved Backbone module are $\alpha \times F_{\text{backbone}}$ and $\alpha \times P_{\text{backbone}}$, respectively. The original GFLOPS of the Neck part is Fneck and the number of parameters is Pneck, and the computational overhead is reduced by a scaling factor $\beta(0 < \beta < 1)$, so that the quantized GFLOPS and Parameters are $\beta \times F_{\text{neck}}$ and $\beta \times P_{\text{neck}}$, respectively. In summary, the parameters and computational quantities of the model before and after definition are

$$F = \alpha \times F_{\text{backbone}} + \beta \times F_{\text{neck}} \tag{1}$$

$$P = \alpha \times P_{\text{backbone}} + \beta \times P_{\text{neck}} \tag{2}$$

3.2. Model Method Description

This paper is inspired by Ghost convolutions in Ghostnet, one of the lightweight state-of-the-art models designed for efficient inference on mobile devices. Its main component is the Ghost module, which uses low-cost operations to generate more feature maps instead of the original convolution. Given an input feature $X \in R^{H \times W \times C}$ with height H, width W, and number of channels C, a typical Ghost module can replace the standard convolution in two steps. First, a 1 × 1 convolution is used to generate the original features, i.e.,

$$Y' = X \times F_{1\times1} \tag{3}$$

$F_{1\times1}$ is a point-by-point convolution, and $Y' \in R^{H \times W \times C'_{out}}$ are intrinsic features whose size is usually smaller than the original output features. Then, the cheap operation (F_{dp} for depth-separated convolution) is used to generate more features based on the intrinsic features. The two parts of the feature are linked along the channel dimension, so that

$$Y = \mathrm{Concat}\left(\left[Y', Y' \times F_{dp}\right]\right) \tag{4}$$

In the Ghost module, only half of the features, the essential features, are smaller than the original output features, which will lose the captured spatial and position information, and to consider this loss, this paper will use the attention module to enhance its spatial and position features.

4. Model Structure

4.1. Yolov5 Algorithm Principle

The YOLOv5 network structure consists of four main parts: Input, Backbone, Neck and Head. The four parts, respectively, perform data input processing, feature learning, feature enhancement processing, and target detection and classification.

Input performs Mosic operations on the input data, mainly cutting, splicing, resizing, and optimizing the input image data to compute the anchor frames. Mosic data augmentation is used to increase the diversity of the dataset, thus increasing the robustness and generalizability of the model.

Backbone is mainly used for feature learning, and the main constituent modules are C3 and SPPF (Spatial Pyramid Pooling—Fast).

The C3 module is similar to the original CSP (Cross-Stage Partial Network) structure, which is mainly used to simplify the network structure, reduce the number of convolutional layers and channels, and maintain the performance, and the SPPF module is the fusion of deep and shallow information to improve the feature extraction ability of the network.

The Neck structure uses a PANet structure to achieve feature enhancement through multi-layer feature fusion of top-down and bottom-up deep and shallow features, thereby increasing the robustness of the model and improving the accuracy of the target detection.

The Head structure obtains the position of the prediction frame target in the input image as well as the category information by designing three detection heads for detecting targets of different scales, each of which acquires feature information of different scale sizes from different layers of the Neck.

4.2. Improve the Structure of the Model

The most time-consuming part of the model is the C3 module, which is used to extract features and enhance the receptive field by reducing the number of channels of the input feature map with a 1 × 1 convolutional layer, then using a set of 3 × 3 sequential convolutions to extract the features, and finally using 1 × 1 convolutions with residual linking to sum the output of the previous step with the output of that layer. In this paper, the C3 modules of the Backbone and Neck sections are quantified. The Backbone part reduces the computational complexity and the number of parameters to the model through the Ghost Module idea and uses Coordinate Attention (CoordAtt) to focus on the global information. Coordinate attention has some unique advantages over other attention mechanisms such as SE (Squeeze-and-Excitation), CBAM [25], SAM [26], ECANet [27], and others. Spatially adaptive: it is able to focus on different locations of the input feature map to capture important contextual information in the image.

Parameter-efficient: it is more advantageous in terms of parameter efficiency compared to SE Attention, which is realized by simple linear transformations and softmax operations, making it more feasible in the case of limited computational resources. The design of Coordinate Attention makes it more flexible and can be used in combination with other attention mechanisms. Combining the various features mentioned above, Coordinate Attention has good assistance in improving the capture of spatial and positional information of features, improving the ability of module feature extraction, and reducing the number of parameters of modules. In the Neck part of the input feature map, $X \in R^{H \times W \times C}$ already contains a large amount of feature information, in the original Neck, the SE attention and Ghost module are used to improve the C3 module, reduce the number of parameters of the module and the extraction of channel features, and the overall structure of the model is shown in Figure 1.

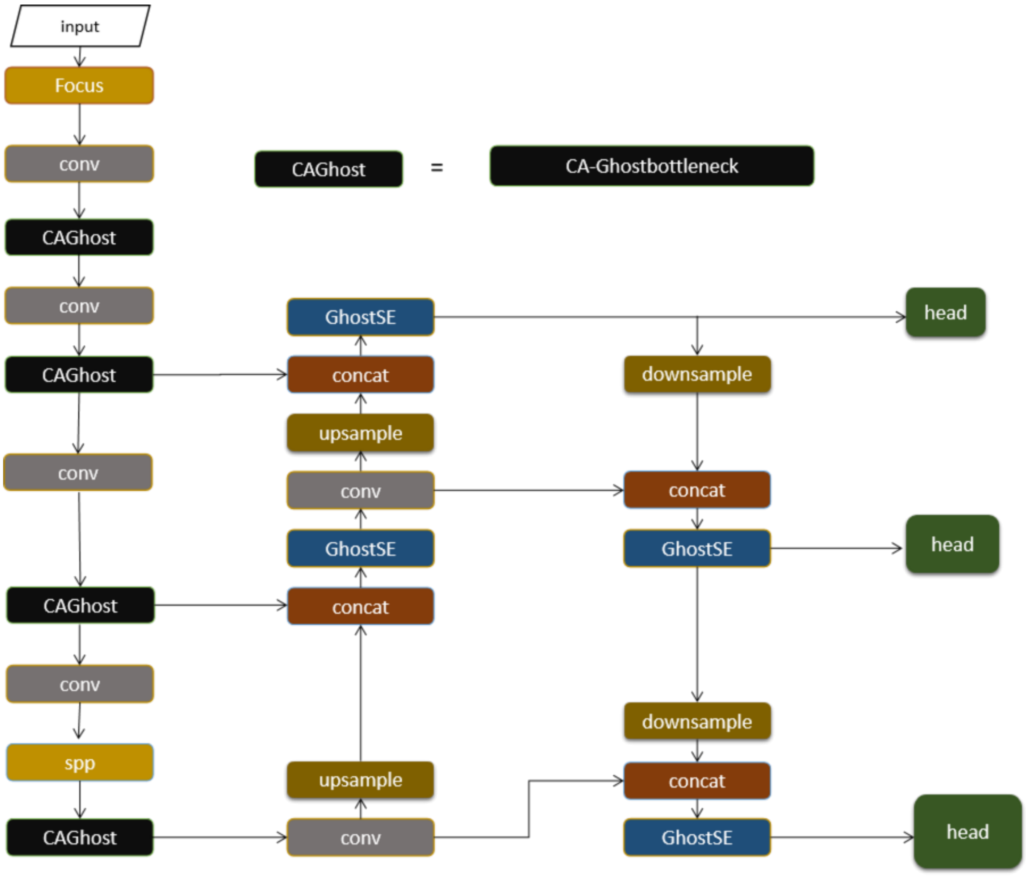


Figure 1. Improved GC-YOLO model diagram. Compared to the native YOLOv5 model, the enhanced GC-YOLO model replaces the original C3 module in the backbone section with the CAGhost and replaces the original C3 module in the Neck section with the GhostSE module.

4.3. *CA-Ghostbotelneck*

CA-GhostBotelneck (shown in Figure 2), as a key network module in the backbone network, adopts ideas from GhostnetV2 [28]. The CA-GhostBotelneck in this paper takes into account the fact that the Ghost module is only half functional and the nature features are smaller than the original output features, and when extracting the features for the input feature map $X \in R^{H \times W \times C}$, the output $Y \in R^{H \times W \times C_{out}}$ is obtained, and the features of Y are lost in both the channel information and the position information. In this paper, the input X is processed in two stages. First, the sparse feature map Y is obtained by the Ghost module, and second, the channel information and position information are obtained by the CoordAttention module, and finally, the two parts of the features are integrated to obtain a new output. The benefits of using CA-GhostBotelneck are as follows:

- Reduce the number of parameters: the Ghost module can use sparse convolution to obtain the nature features, improving the lightness of the effect.
- Improve model expressiveness: CoordAttention captures channel and position information, allowing more flexible access to global feature information and improving model expressiveness.

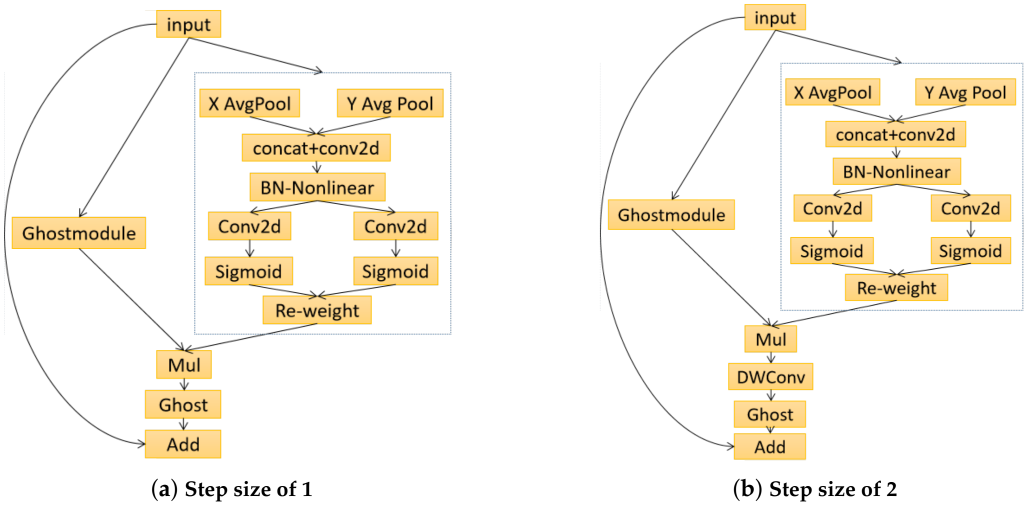


Figure 2. CA-GhostBotelneck with step size 1 on the left, CA-GhostBotelneck with step size 2 on the right.

Given an input feature $X \in R^{H \times W \times C}$ with height H, width W, and number of channels C, the CA-GhostBotelneck module can replace the normal convolution in two steps. First, a 1×1 convolution is used to generate the intrinsic features, i.e.,

$$Y' = X \times F_{1\times 1} \tag{5}$$

$Y' \in R^{H \times W \times C'_{out}}$ are intrinsic features whose sizes are usually smaller than the original output features, which compensate for the lack of original channel and position information by having stronger feature information from CoordAttention than Deep-WiseConv, i.e.,

$$F_{\text{coordAtt}} > F_{dp} \tag{6}$$

$$Y = \text{Concat}\left(\left[Y', Y' \times F_{\text{coordAtt}}\right]\right) \tag{7}$$

4.4. GhostSE

In this paper, the GhostSE structure is used in the Neck part (as shown in Figure 3), and the intrinsic features obtained by 1×1 convolution have fewer output features than those obtained by ordinary convolution. Improved access to channel information of feature maps using SE attention to capture more features to compensate for the feature loss problem caused by convolution in the Ghost idea, second, residual joining is used to obtain richer feature output, and finally residual joining is performed using the GhostConvSE module and GhostBottelneck, reducing the number of parameters and float calculations while keeping as much feature-rich information as possible.

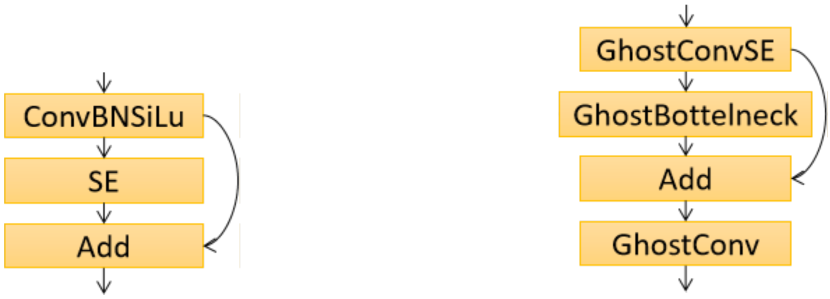
Given an input feature $X \in R^{H \times W \times C}$ with height H, width W, and number of channels C, output Y' via 1×1 Conv,Y' via SE Attention to output Y'', input Y'' in GhostSE, and output Z' after Add, and finally output the feature map Z, i.e.,

$$Y' = X \times F_{1\times 1} \tag{8}$$

$$Y'' = \text{Concat}\left(\left[Y', Y' \times F_{SE}\right]\right) \tag{9}$$

$$Z' = \text{Concat}\left(\left[Y'', Y'' \times F_{\text{GhostBottelneck}}\right]\right) \tag{10}$$

$$Z = Z' \times F_{\text{GhostConv}} \tag{11}$$



(**a**) **GhostConvSE** (**b**) **GhostSE**

Figure 3. The left image shows GhostConvSE, which uses SE attention to obtain more feature information; the right image shows GhostSE.

5. Experiment

5.1. Experimental Environment

The experiment selects the PASCAL VOC dataset commonly used for target detection for training, which is mainly used for detecting the four major classes of vehicle, household, animal and person in the environment, and the detection target samples are relatively abundant. The computer configuration for the experiment is GPU: RTX 3060, CPU: I5-10400, 16G RAM; the training network environment is Python: 3.9, CUDA12.1.

5.2. Model Evaluation

In target detection tasks, it is often necessary to compare the predicted results with the true labels, and three metrics are used to evaluate model performance in this process.

1. *TP*: Means labeled as a positive sample and predicted as a positive sample.
2. *FP*: Means that the label is a negative sample and the prediction is a positive sample.
3. *FN*: Refers to samples labeled as positive samples but predicted to be negative.
4. *TN*: Means that the label is a negative sample and the prediction is a negative sample.
5. *Precision*: Indicates the percentage of samples that were correctly predicted out of those predicted as positive examples.

$$P = TP/(TP + FP) \tag{12}$$

6. *Recall*: indicates the proportion of positive samples that are true positive samples.

$$R = TP/(TP + FN) \tag{13}$$

7. *mAP*: Used to evaluate overall model detection performance in multiple categories. Where n is the number of categories, AP_i is the average precision of the *i*-th category, and *r* is the *recall*.

$$AP = \int_0^1 P(r)dr, r \in (0,1) \tag{14}$$

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{15}$$

8. *F1 score*: Combines Precision and Recall to evaluate the performance of the model and is defined as the harmonic mean of Precision and Recall.

$$F1 = 2 \times P \times R/(P + R) \tag{16}$$

By balancing the lightness, detection accuracy and detection speed of the model, this paper improves the model. By calculating the Efficient value, the model M that balances the detection efficiency and speed is finally obtained.

$$\text{Efficient } = E(F, P, mAP, R, F1) \tag{17}$$

$$M = \max_{\mathcal{M} \in \mathbb{M}} \text{ Efficient } {}_{i}^{\mathcal{M}} \tag{18}$$

5.3. Experiment

In order to verify the overall improvement of GC-YOLO of the design model, this paper designs several comparative experiments with typical lightweight networks. The PASCAL VOC dataset is selected, and the experimental dataset is divided into the training set and the validation set with a ratio of 9:1, the image size is 640 × 640, the training batch is set to 32, and all reference models are trained for 300 epochs according to this parameter. The experiments compared the number of model parameters, GFLOPS, mean accuracy *mAP*: 0.5, and harmonic mean *F1*.

As shown in Table 1, the original Yolov5s had 7.28 M parameters and 17.16 G GFLOPS. With CA-GhostBotelneck and GhostSE's GC-YOLO, the number of parameters is 2.8 M less and GFLOPS are 8.53 G less with a slight increase in *mAP* and *F1*. The results show that the model's feature extraction capability is significantly improved and at the same time the model's parameter number is reduced.

Table 1. Comparative testing of models.

Model	Parameter (M)	GFLOPS (G)	*mAP* (%)	*F1*	*FPS*
YOLOV5s	7.28	17.16	84.06	0.62	25
Yolov4-MobileNetv3	11.73	18.22	69.13	0.68	28
Yolov4-tiny	6.1	6.96	64	0.54	30
Yoloxs	8.95	26.73	83.8	0.74	18
Yolov7-tiny	6.23	13.86	80.83	0.76	26
GC-Yolo(our)	4.48	8.63	84.19	0.72	24

The partial detection results of the GC-YOLO model are shown in figures. Figure 4 shows the harmonic mean F1 after training the model on the VOC dataset with the threshold set to 0.5. The F1 value combines the accuracy and completeness of the model, and is particularly useful for dealing with category imbalance or focusing on improving both accuracy and recall. Higher F1 values are an indication of better model performance in positive sample detection and negative sample exclusion, from the twenty categories in the figure, it is evident that the F1 values of 12 categories are higher than the average value of 0.72, and there are only a few major fluctuations, which proves that the model achieves a relatively balanced performance in different categories, has a better generalization ability in each category, can adapt to different categories, and has an advantage in dealing with multicategory problems; Figure 5 shows the average accuracy of the model mAP on each category, combining the prediction accuracy and recall of the model on different target categories, which is used to measure the overall performance of the model on multiple target categories, as shown in the figure, the data are more tightly clustered, with 13 categories exceeding 84.19%, five categories surpassing 90%, and two categories falling below 70%. This suggests that the model is successful in accurately localizing and identifying the target object across multiple categories, without excessively focusing on certain categories and disregarding others, and with exceptional overall performance. Figure 6 shows the "loss rate" of the model, especially in the security and surveillance area. It reflects the proportion of targets missed by the model during target detection, and a lower leakage rate indicates that the model performs better in target detection and is able to capture targets more comprehensively, from the figure, it is evident that the leakage rate is mainly below 0.3, with five categories exceeding this threshold. However, the highest leakage rate is only 0.56, indicating that the model has a high recall rate and can detect most of the target objects. It is also robust to the size and location of different targets, ensuring a consistently low leakage rate.

In addition to testing GC-YOLO detection on images outside the VOC dataset, in this paper, an image downloaded from the Internet was used for detection, and the comparative detection experiment is shown in Figure 7, and it is evident that (a) enhances target recognition accuracy by approximately 0.05 in unobstructed and approximately 0.1 in obstructed targets compared to (b). The model's overall accuracy for recognizing categories is enhanced, including the recognition of yellow cars.

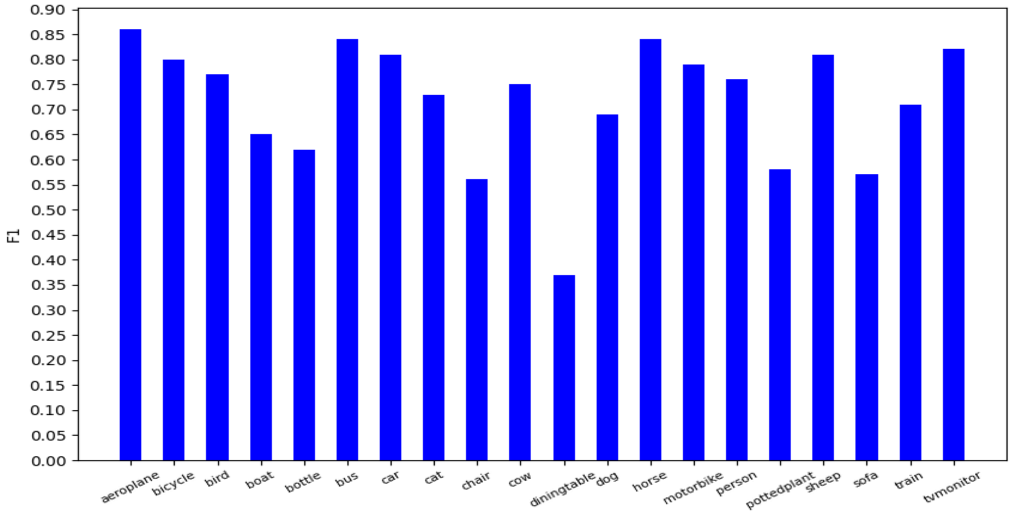


Figure 4. Harmonic mean F1 values of the GC-YOLO model for the VOC dataset.

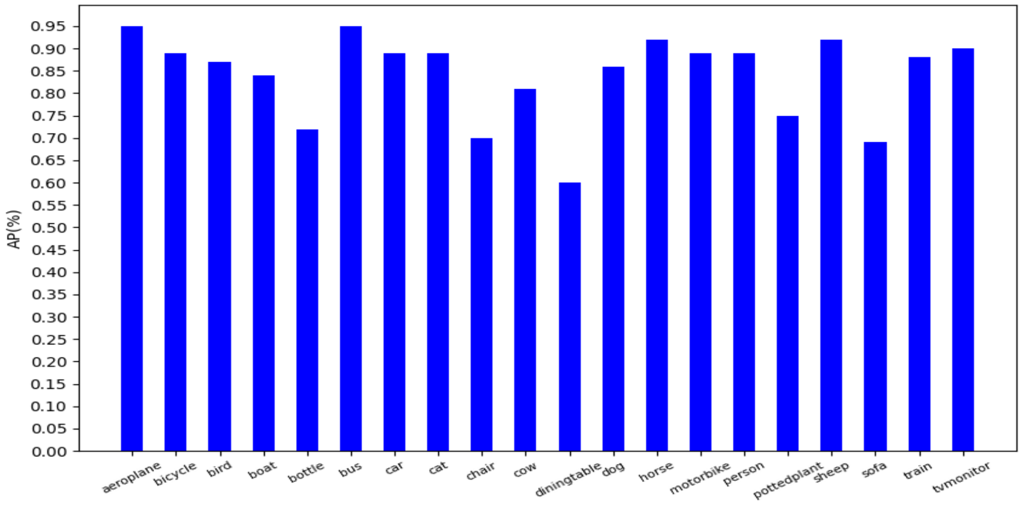


Figure 5. Average accuracy of the GC-YOLO model on the VOC dataset (mAP = 84.19%).

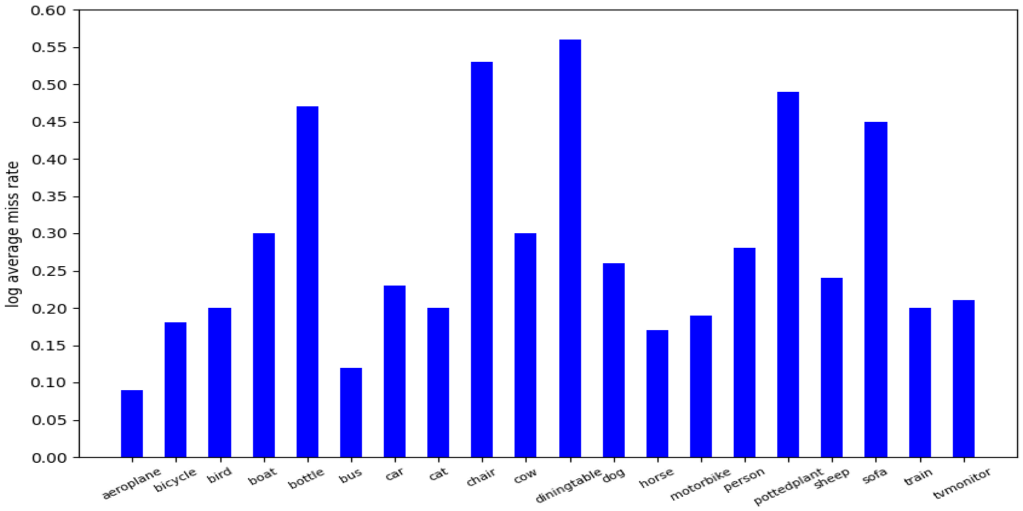
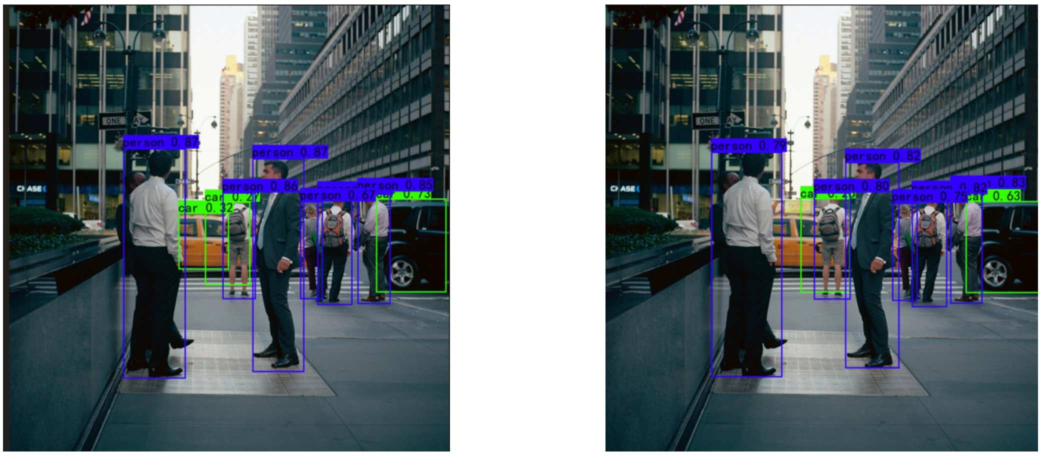


Figure 6. GC-YOLO model miss rate in VOC dataset.



(**a**) (**b**)

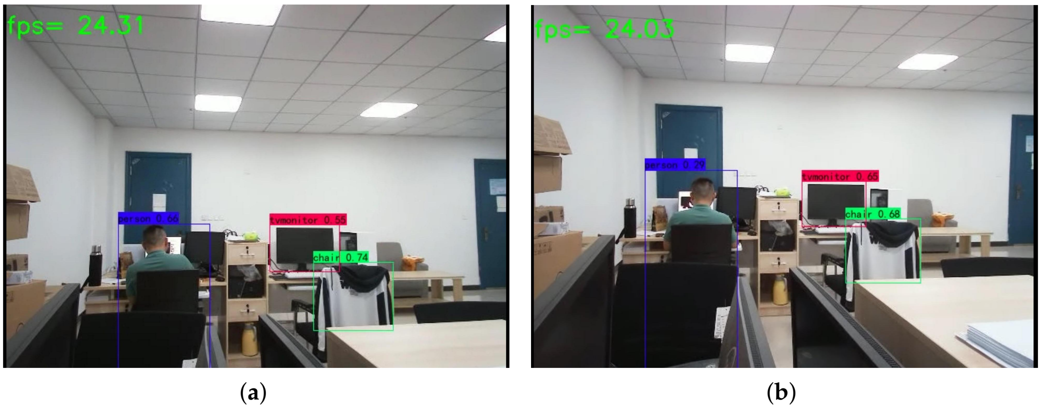
Figure 7. (**a**) GC-YOLO model detection results; (**b**) original model detection results. In this context, blue boxes indicate people, while green boxes represent cars. Compared to Figure (**b**), Figure (**a**) displays superior accuracy in identifying individuals, successfully detecting the concealed yellow car, and demonstrating an increased confidence in identifying the black car. Furthermore, there is a decreased likelihood of misidentifying a person as a vehicle.

Comparison experiments were also performed between the GC-YOLO model and the FPS of Yolov5's real-time detection, as shown in Figure 8. The model introduces attention to improve the extraction of features, while ensuring the real-time performance of the lightweight model, and the model's FPS is relatively smooth.

5.4. Scenario Experiments with Custom Datasets

This GC-YOLO lightweight model is used in the realization of intelligent wheelchair devices on the blind spot obstacle detection to help the user to avoid the blind spot on both sides of the obstacles caused by safety issues.

It was found that the visual angle of wheelchair blindness is different from the unusual visual angle, and is more a differentiator caused by the low viewing angle, which is reflected in the overhead angle and the incompleteness of the display target. In this paper, dangerous obstacles for wheelchair blindness in the community are defined as cats, dogs, potholes, water bottles, and feet, legs, and wheels at the top view, and the seven categories are designed to avoid physical injury from animal attacks, falls, and wheel accidents. For this purpose, field targets were collected for dataset production, mainly through mobile phone simulation of wheelchair heights and perspectives in different scenarios, based on different age groups, scenarios, and time periods, to build diverse sample data. In this paper, the captured video data are processed, and the video file is sliced by frames, and one image is selected every ten frames to obtain the foot and leg photos under different angles and different poses. To ensure the diversity of the dataset and the amount of data, some of the categories of the dataset were obtained from publicly available datasets and web crawling, respectively, totaling 7916 image data. To solve the problem of unbalanced data volume in the target dataset, this paper sets the data enhancement rate of mosaic to fifty percent, which expands the data diversity and improves the generalization ability of the training model. The experiment is shown in Figure 9, and the precision for various categories exceeds 0.85, some even surpassing 0.95. The model demonstrates excellent robustness in its capacity to generalize across targets of differing sizes such as feet, legs, and wheels. which shows that the performance of the overall model on the custom dataset is relatively stable, with a mAP of 90.34%, and Figure 10 shows that the average accuracy of the F1 values is 0.84 at score threshold = 0.5, on the custom dataset, the F1 score is no worse than the average of the model trained on the VOC dataset (0.72), and the model can adapt to different categories, demonstrating its ability to generalize.



(**a**) (**b**)

Figure 8. (**a**) Shows the FPS detection effect of the GC-YOLO model; (**b**) shows the FPS detection effect of the original model. The detection rates of the two graph algorithms have been compared, and the modified algorithm consistently maintains high performance without any reduction in detection rates.

The trained model is tested in real scenarios, and the results of the test scenarios are shown in Figure 11. For intelligent wheelchair obstacle detection in the blind zones on both sides of the wheelchair in a senior living community environment, side safety is judged mainly based on the display of incomplete targets. The four images reflect wheels, legs, feet, and potholes in low vision, and the first three images judge obstacle targets based on human targets with incomplete displays in low vision.

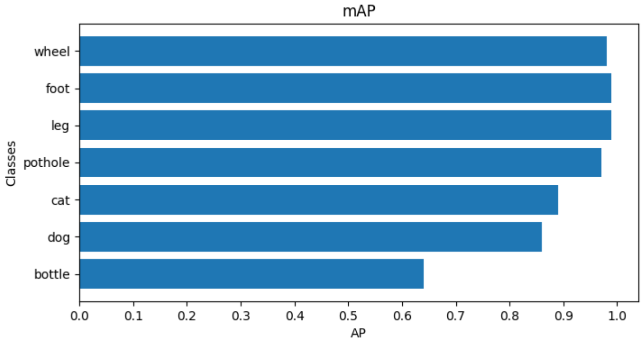


Figure 9. Average accuracy of the GC-YOLO model on the custom set (mAP = 90.34%).

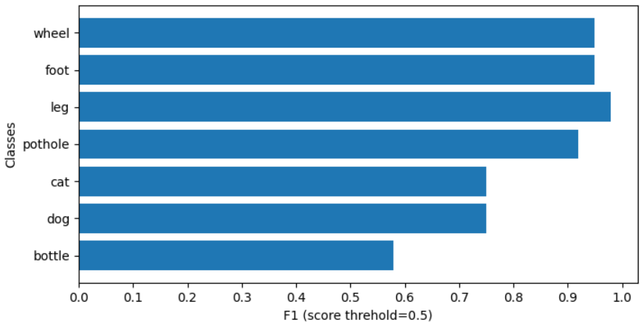
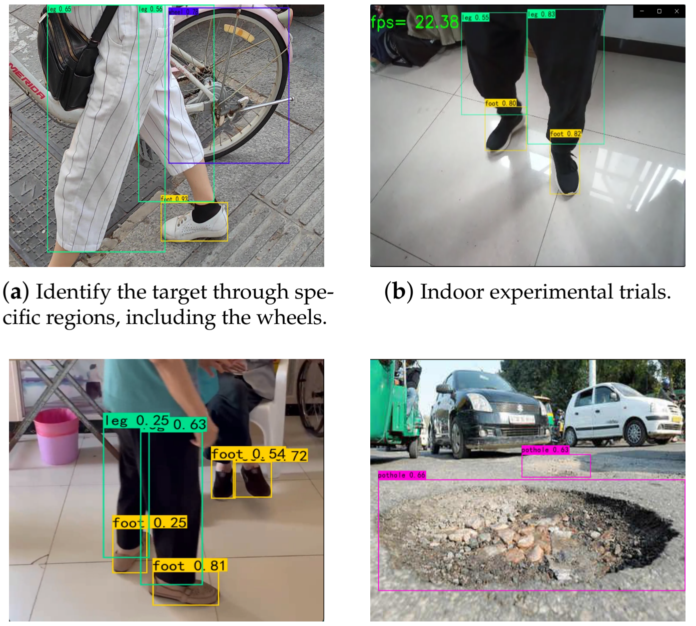


Figure 10. F1 of the GC-YOLO model on the homebrew set (score threshold = 0.5).



(**a**) Identify the target through specific regions, including the wheels.

(**b**) Indoor experimental trials.

(**c**) Data within the elderly community, illustrating the algorithm's detection efficacy.

(**d**) From a low angle, the algorithm's efficacy in detecting potholes.

Figure 11. The four panels show the real scene model detection effect. Distinctive colors serve to discern and display diverse categories: green boxes denote legs, purple boxes symbolize wheels, yellow boxes indicate feet, and red boxes represent potholes.

The above experimental results show that compared with YOLOv5s, YOLOv4-mobilenetv3 and other lightweight model algorithms, the real-time performance of the GC-YOLO model is as stable as that of the native YOLOv5s, but it has been improved in the number of parameters, GFLOPS, mAP, and the evaluation of the F1 value, and it also has a very good performance in the custom dataset to perform the safety supervision of the blind zones of intelligent wheelchair detection.

6. Conclusions

In this paper, we propose a lightweight target detection algorithm GC-YOLO based on YOLOv5. By improving the network, the model is able to achieve good detection performance while being lightweight, balancing the relationship between lightweight and detection performance. In intelligent wheelchairs for the elderly community that have a good lightweight performance, the research found that the model has good detection performance in blind spot obstacle detection to avoid potential safety threats. In future work, the algorithm will deploy on *Nvidia Jetson Nano*, and cameras will install on both sides of the wheelchair to detect each side independently. Subsequent experiments will aim to further improve and optimize the system. However, limitations may arise during the experimental process, as well as during the maintenance and retraining of the model after deployment on the mobile terminal. When major environmental changes occur, the model's performance may diminish. In our future studies, we will explore the integration of multi-modal or unsupervised learning approaches to improve the model's responsiveness to environmental fluctuations and continue our research in this area.

Author Contributions: Conceptualization, S.Z. and J.D.; methodology, Y.C. and J.D.; validation, S.Z. and C.S.; formal analysis, J.D. and S.Z.; investigation, S.Z.; data curation, S.Z., C.S. and J.D.; writing—original draft preparation, J.D.; writing—review and editing, S.Z., Y.C., C.S. and J.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Chuzhou University. This work was supported by Anhui Higher Education Research Program Project under Grant 2022AH010067 (Title: Smart Elderly Care and Health Engineering Scientific Research Innovation Team).

Data Availability Statement: Data sharing not applicable. Further research is needed.

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

References

1. Ahmadi, A.; Argany, M.; Neysani Samany, N.; Rasooli, M. Urban Vision Development in Order To Monitor Wheelchair Users Based on The Yolo Algorithm. *Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2019**, *XLII-4/W18*, 25–27. [CrossRef]
2. Okuhama, M.; Higa, S.; Yamada, K.; Kamisato, S. *Improved Visual Intention Estimation Model with Object Detection Using YOLO*; IEICE Technical Report; IEICE Tech: Tokyo, Japan, 2023; Volume 122, pp. 1–2.
3. Chatzidimitriadis, S.; Bafti, S.M.; Sirlantzis, K. Non-Intrusive Head Movement Control for Powered Wheelchairs: A Vision-Based Approach. *IEEE Access* **2023**, *11*, 65663–65674. [CrossRef]
4. Hashizume, S.; Suzuki, I.; Takazawa, K. Telewheelchair: A demonstration of the intelligent electric wheelchair system towards human-machine. In Proceedings of the SIGGRAPH Asia 2017 Emerging Technologies, Bankok, Thailand, 27–30 November 2017; p. 1.
5. Suzuki, I.; Hashizume, S.; Takazawa, K.; Sasaki, R.; Hashimoto, Y.; Ochiai, Y. Telewheelchair: The intelligent electric wheelchair system towards human-machine combined environmental supports. In Proceedings of the ACM SIGGRAPH 2017 Posters, Los Angeles, CA, USA, 30 July–3 August 2017; p. 1.
6. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. Pointpillars: Fast encoders for object detection from point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12697–12705.
7. Meyer, G.P.; Laddha, A.; Kee, E.; Vallespi-Gonzalez, C.; Wellington, C.K. Lasernet: An efficient probabilistic 3D object detector for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12677–12686.
8. Chen, Q.; Chen, Y.; Zhu, J.; De Luca, G.; Zhang, M.; Guo, Y. Traffic light and moving object detection for a guide-dog robot. *J. Eng.* **2020**, *13*, 675–678. [CrossRef]
9. Ferretti, S.; Mirri, S.; Roccetti, M.; Salomoni, P. Notes for a collaboration: On the design of a wiki-type educational video lecture annotation system. In Proceedings of the IEEE International Conference on Semantic Computing (ICSC 2007), Irvine, CA, USA, 17–19 September 2007; pp. 651–656.
10. 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.
11. 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, 20–21 June 2021; pp. 13713–13722.
12. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
13. Lee, H. J.; Ullah, I.; Wan, W.; Gao, Y.; Fang, Z. Real-time vehicle make and model recognition with the residual SqueezeNet architecture. *Sensors* **2019**, *19*, 982. [CrossRef] [PubMed]
14. Sheng, T.; Feng, C.; Zhuo, S.; Zhang, X.; Shen, L. A quantization-friendly separable convolution for mobilenets. In Proceedings of the IEEE 2018 1st Workshop on Energy Efficient Machine Learning and Cognitive Computing for Embedded Applications (EMC2), Williamsburg, VA, USA, 25 March 2018.
15. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
16. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; 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.

17. Nascimento, M.G.; Fawcett, R.; Prisacariu, V.A. Dsconv: Efficient convolution operator. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 5148–5157.
18. 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.
19. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the 2019 International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 6105–6114.
20. He, Y.; Zhang, X.; Sun, J. Channel pruning for accelerating very deep neural networks. In Proceedings of the IEEE 2017 International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1389–1397.
21. Cai, Y.; Li, H.; Yuan, G.; Niu, W.; Li, Y.; Tang, X.; Ren, B.; Wang, Y. Yolobile: Real-time object detection on mobile devices via compression-compilation co-design. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; pp. 955–963.
22. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2704–2713.
23. Yang, Y.; Sun, X.; Diao, W.; Li, H.; Wu, Y.; Li, X.; Fu, K. Adaptive knowledge distillation for lightweight remote sensing object detectors optimizing. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–15. [CrossRef]
24. Qu, J.; Chen, B.; Liu, C.; Wang, J. Flight Delay Prediction Model Based on Lightweight Network ECA-MobileNetV3. *Electronics* **2023**, *12*, 1434. [CrossRef]
25. 25 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.
26. Zhu, X.; Cheng, D.; Zhang, Z.; Lin, S.; Dai, J. An empirical study of spatial attention mechanisms in deep networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6688–6697.
27. 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.
28. Tang, Y.; Han, K.; Guo, J.; Xu, C.; Xu, C.; Wang, Y. GhostNetv2: Enhance cheap operation with long-range attention. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 9969–9982.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.