electronics

MDPI

Article

Integration of ShuffleNet V2 and YOLOv5s Networks for a Lightweight Object Detection Model of Electric Bikes within Elevators

Jingfang Su, Minrui Yang and Xinliang Tang *

School of Information Science, Hebei University of Science and Technology, Shijiazhuang 050018, China; sujingfang1980@hebust.edu.cn (J.S.); yangminrui@stu.hebust.edu.cn (M.Y.)
* Correspondence: tangxinliang@hebust.edu.cn

Abstract: The entry of electric bikes into elevators poses safety risks. This article proposes a lightweight object detection model for edge deployment in elevator environments specifically designed for electric bikes. Based on the YOLOv5s network, the backbone network replaces the original CSPDarknet53 with a lightweight multilayer ShuffleNet V2 convolutional neural network, achieving a lightweight backbone network. Swin Transformer modules are introduced between layers to enhance the feature expression capability of images, and a SimAM attention mechanism is applied at the end layer to further improve the feature extraction capability of the backbone network. In the neck network, lightweight and depth-balanced GSConv and VoV-GSCSP modules replace several Conv and C3 basic convolutional modules, reducing the parameter count while enhancing the cross-scale connection and fusion capabilities of feature maps. The prediction network uses the faster-converging and more accurate EIOU error function as the position loss function for iterative training. This article conducts various lightweighting comparison experiments and ablation experiments on the improved object detection model. The experimental results demonstrate that the proposed object detection model, with a model size of only 2.6 megabytes and 1.1 million parameters, achieves a frame rate of 106 frames per second and a detection accuracy of 95.5%. This represents an 84.8% reduction in computational load compared to the original YOLOv5s model. The model's volume and parameter count are reduced by 81.0% and 84.3%, respectively, with only a 0.9% decrease in mAP. The improved object detection model proposed in this paper can meet the real-time detection requirements for electric bikes in elevator scenarios, providing a feasible technical solution for its deployment on edge devices within elevators.

Keywords: electric bike; lightweight; YOLOv5s; ShuffleNet V2; Swin Transformer

Citation: Su, J.; Yang, M.; Tang, X. Integration of ShuffleNet V2 and YOLOv5s Networks for a Lightweight Object Detection Model of Electric Bikes within Elevators. *Electronics* **2024**, *13*, 394. https://doi.org/10.3390/electronics13020394

Academic Editor: Heung-Il Suk

Received: 28 November 2023
Revised: 13 January 2024
Accepted: 15 January 2024
Published: 18 January 2024

1. Introduction

Due to differences in natural geography and population density, countries such as China and Southeast Asia, with dense urban populations and relatively crowded environments, exhibit a strong demand for electric bikes as a convenient and affordable mode of transportation. Electric bikes have become the preferred means of travel for middle- and lower-income groups. Data indicate that by the end of 2022, the number of electric bikes in China exceeded 300 million [1]. However, in Chinese cities, residents often reside in high-rise buildings, and the availability of ground charging stations within residential communities is limited. This scarcity makes it difficult to meet people's charging needs for electric bikes. So, charging electric bikes indoors is a common phenomenon. Charging electric bikes poses safety hazards, including the risk of fire accidents [2]. If electric bikes are charged indoors, the potential dangers could pose a significant threat to people's lives. Installing a detection and alarm system for electric bikes entering elevators is an effective measure to prevent the use of elevators with electric bikes. Therefore, this paper focuses

on researching a lightweight object detection model that can be practically deployed in elevator environments.

In recent years, deep learning has been widely applied to object detection tasks in urban scenarios. There are numerous object detection models based on deep learning, and the YOLO [3,4] series models, with their end-to-end network architecture and shorter inference latency, are more practical for deployment compared to two-stage object detection models (such as Faster RCNN [5]). Consequently, practical deployments of object detection models in different real-world scenarios are based on the YOLO series models [6]. However, there is relatively limited research on object detection algorithms specifically tailored for the scenario of electric bikes entering elevators. The only relevant research that the authors have found is listed below.

The authors of [7] proposed a highly secure, intelligent electric bike management system tailored for elevators. The approach abandoned cloud deployment in favor of a safer edge-based deployment method. The monitoring images were processed and trained locally, utilizing data enhancement and hybridization strategies with MobileNet-SSD. The experimental results demonstrated a 19.6% reduction in delay compared to cloud planning deployment, achieving a high recall rate of 82%. In [8], an electric bike detection and early warning system for elevator scenarios was introduced. The method involved enhancing YOLOv3 networks to improve feature extraction and recognition accuracy. Additionally, integrating multi-frame direction fields was proposed to address challenges such as virtual police reduction and resolving target object cover. The authors of [9] focused on rebuilding YOLOv4's Featured Pyramid and corresponding backbone feature extraction network. The attention mechanism was integrated into the residual network of the backbone to enhance detection accuracy and enable the intelligent detection of various scenarios of electric bikes in elevators. In a different approach, in article [10], YOLOv3 algorithm was deployed on Raspberry Pi to identify electric bikes in elevators. Although it developed an online real-time monitoring system, the experiment's test accuracy was not notably high. Furthermore, the authors of [11] introduced a machine-learning algorithm for rapidly identifying electric bikes in elevators. The CBAM attention mechanism was incorporated into the main and head parts of YOLOv5 to enhance feature map expressiveness. The carAFE operator replaced the closest operator in the original model, resulting in precise details and auxiliary information. The improved model achieved a mAP of 86.35% and a recall rate of 81.8%, marking a 3.49% increase in average detection accuracy compared to the YOLOv5 model. The recall rate also increased by 5.6%. Ultimately, when deployed on the Jeston TX2 NX hardware platform, the model demonstrated the stable and effective identification of electric bikes.

Most of the above-mentioned studies focus on improving the model's detection accuracy and recall rate, with little consideration for lightweight model improvements. Electric bikes are considered large objects, and currently, YOLO models can achieve good performance in recognizing large objects on general computing devices. However, there is a lack of research on how to enable the model to perform real-time and accurate inference on devices with limited computational resources, such as mobile or embedded devices. In a mobile environment like elevators, adopting a detection method that involves cloud deployment and online calls makes it challenging to ensure real-time performance and security. Therefore, deploying a lightweight object detection model directly on edge devices such as cameras is a more reliable approach.

2. Related Work

In recent years, lightweight CNN neural networks that have emerged include MobileNet [12] proposed by Google in 2017, ShuffleNet [13] proposed by Face++ in 2017, and GhostNet [14] proposed by Huawei Noah's Ark Lab in 2020. MobileNet significantly reduces the computational complexity of ordinary convolutions by nearly an order of magnitude by incorporating the depthwise convolution (DW) and pointwise convolution (PW) calculation method in convolutional computations. MobileNet was updated to version 3

in 2019 [15]; it substantially decreases model parameters and computational requirements compared to traditional CNN networks, albeit with a slight decrease in accuracy. The core idea of ShuffleNet is the combination of Pointwise Group Convolution and channel shuffle strategies. It replaces pure convolutional computations with channel shuffle operations, reducing computational costs while enhancing the multi-channel feature extraction capability. In the v2 version [16], particular attention is given to memory access costs, leading to lightweight updates. Experimental results on ImageNet classification and MSCOCO object detection demonstrate ShuffleNet's superior performance compared to other structures. For instance, under a 40 MFLOP computational budget, it achieves a lower top-1 error in ImageNet classification compared to the latest version of MobileNet. GhostNet splits the traditional convolution into two steps, first generating feature maps with fewer channels through conventional convolution. Subsequently, based on this set of original feature maps, it applies a series of linear transformations at minimal cost to produce many "Ghost" feature maps that can extract necessary information from the original features. Finally, the two sets of feature maps are concatenated to obtain the final output. Theoretically, GhostNet achieves higher recognition accuracy than MobileNetV3, reaching a top-1 accuracy of 75.7% on the ImageNet ILSVRC-2012 classification dataset.

These lightweight convolutional neural networks have been updated in terms of model structure and computational methods compared to traditional CNNs, significantly reducing model complexity and computational costs. Each of the three networks has its strengths in terms of accuracy, inference speed, and model size. However, research indicates that in actual deployment on embedded devices, comprehensive performance metrics including top-1 accuracy, measured FLOPs, parameter size, memory consumption, model file size, inference speed, and model training time show that ShuffleNet performs the best. This point is further validated in the experimental section of this paper.

Based on the above research, in order to better suit deployment on embedded devices, this paper proposes the idea of replacing the original YOLOv5s backbone network with a stacked network of ShuffleNet V2 modules. Additionally, the Swin Transformer framework and SimAM attention mechanism are introduced to the backbone network to address the decrease in feature extraction accuracy caused by lightweighting. The neck network utilizes lightweight modules GSConv and VOV-GSCSP to enhance feature fusion capability. Finally, the EIOU loss function is employed to accelerate the model's convergence speed. Part 3 provides a detailed introduction to the improved network structure.

3. Models

3.1. YOLOv5s

The YOLOv5 algorithm constitutes a deep learning-based object detection network featuring four variants: YOLOv5l, YOLOv5m, YOLOv5x, and YOLOv5s. YOLOv5s, notable for its high accuracy and relatively swift processing, also boasts a reduced parameter count. The algorithm comprises four key components: input, backbone, neck, and prediction. Within the backbone, image feature extraction occurs through convolution, slicing, and spatial pyramid pooling operations. The neck leverages the FPN + PAN structure to amalgamate information from features at different scales. The CIoU loss function is applied in the prediction network, generating three types of target predictions: Bounding Boxes, confidence scores, and Category labels. The architecture of YOLOv5s is illustrated in Figure 1.

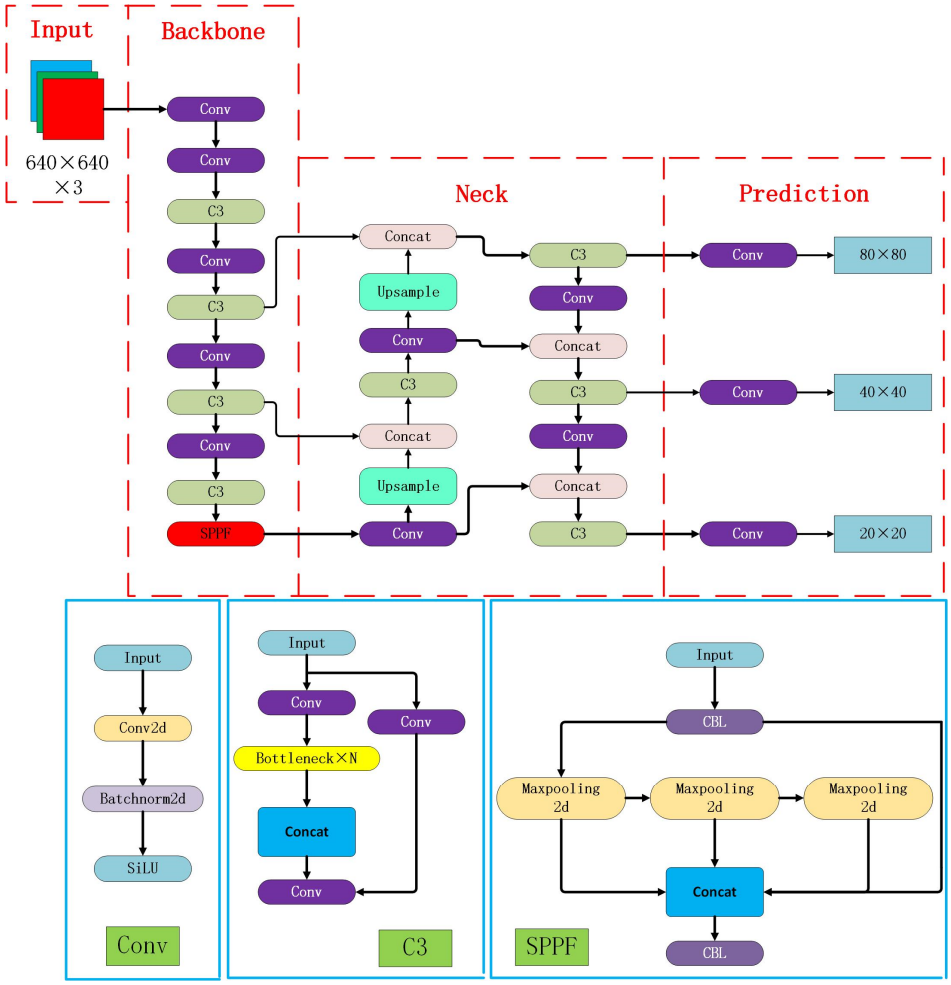


Figure 1. YOLOv5s algorithm framework.

3.2. Lightweight Improvements to YOLOv5s

To adapt to the limited computing and storage resources of mobile and embedded devices, this paper has enhanced the YOLOv5s network, as illustrated in Figure 2. Firstly, the backbone network section replaces the original CSPDarknet53 with multiple layers of ShuffleNet V2 modules to reduce computational load and model parameters. The Swin Transformer [17] framework is introduced between layers, providing enhanced feature extraction capabilities through its sliding window and hierarchical structure mechanisms. The SimAM [18] (Simple Attention Mechanism) module is employed in the last layer to strengthen crucial information in the image and suppress irrelevant details, further improving the feature extraction capabilities of the backbone network. In the neck section of the model, lightweight GSConv and VOV-GSCSP [19] modules are employed to replace the original Conv and C3 modules, aiming to extract richer feature information while reducing computational load. Finally, in the prediction network part, the EIOU [20] bounding box loss function is utilized to explicitly capture the direct differences in width and height between predicted and true boxes, thereby enhancing the model's convergence speed. The following provides detailed explanations of each improvement.

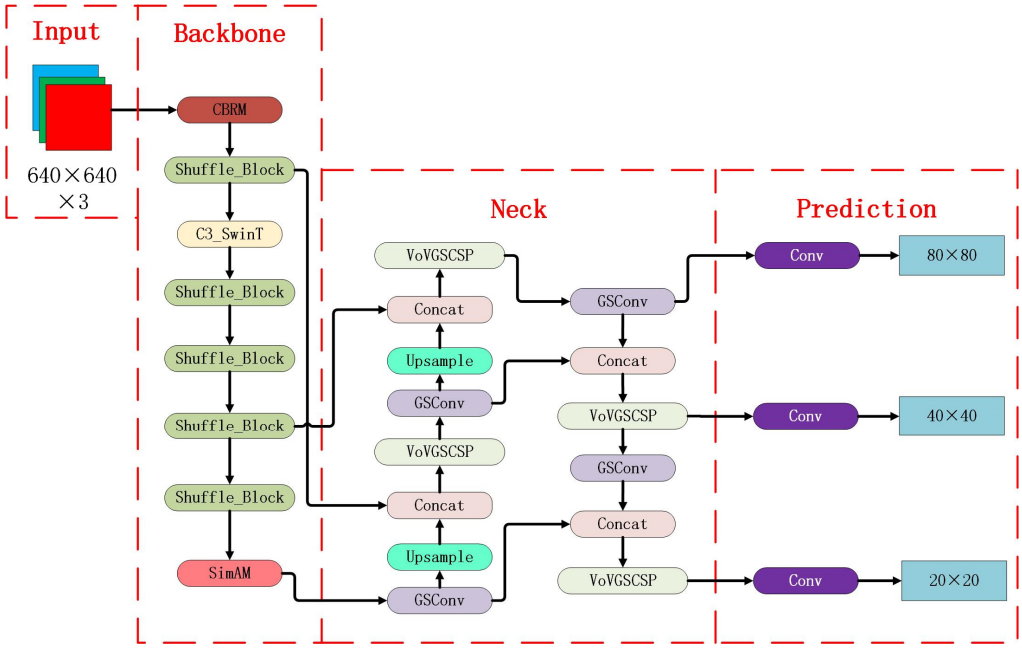


Figure 2. Improved lightweight YOLOv5 s model.

3.2.1. Improvements to the Backbone Network

The backbone network of YOLOv5s, known as CSPDarknet53, utilizes a multi-layered convolutional neural network comprising standard convolutional layers and batch normalization, and incorporates the CSPNets architecture with residual connections. This structure is relatively large and imposes high demands on computational hardware. In this paper, CSPDarknet53 is replaced with a hierarchical network composed of ShuffleNet V2 modules, aiming to reduce computational overhead. As illustrated in Figure 2, the backbone network consists of five stacked Shuffle Blocks.

ShuffleNet V2 primarily consists of two unit modules, Unit1 and Unit2. The structure of the Unit1 module is illustrated in Figure 3a. Firstly, the input data undergo channel separation, where the left channel is outputted unchanged, and the right channel undergoes 1×1 standard convolutions at the front and back ends, along with a 3×3 depthwise convolution in the middle. The output from the right channel is then merged with the left channel. The resulting feature map undergoes channel shuffle (Channel ShuffleNet). The Unit1 module focuses solely on feature extraction without altering the size and channel count of the input feature map. The structure of the Unit2 module is depicted in Figure 3b, eliminating the channel separation operation present in Unit1. Both the left and right channels undergo downsampling operations. They halve the height and width of the input feature map and expand the number of channels to 4 times, increasing the network's width without affecting the running speed. Finally, the channel rearrangement method achieves feature functions between different groups.

As depicted in Figure 4, the channel shuffle structure employs specific computational methods to decompose and reassemble channels, facilitating information exchange between different groups. This process reshapes the feature maps in the output layer, achieving rapid feature extraction.

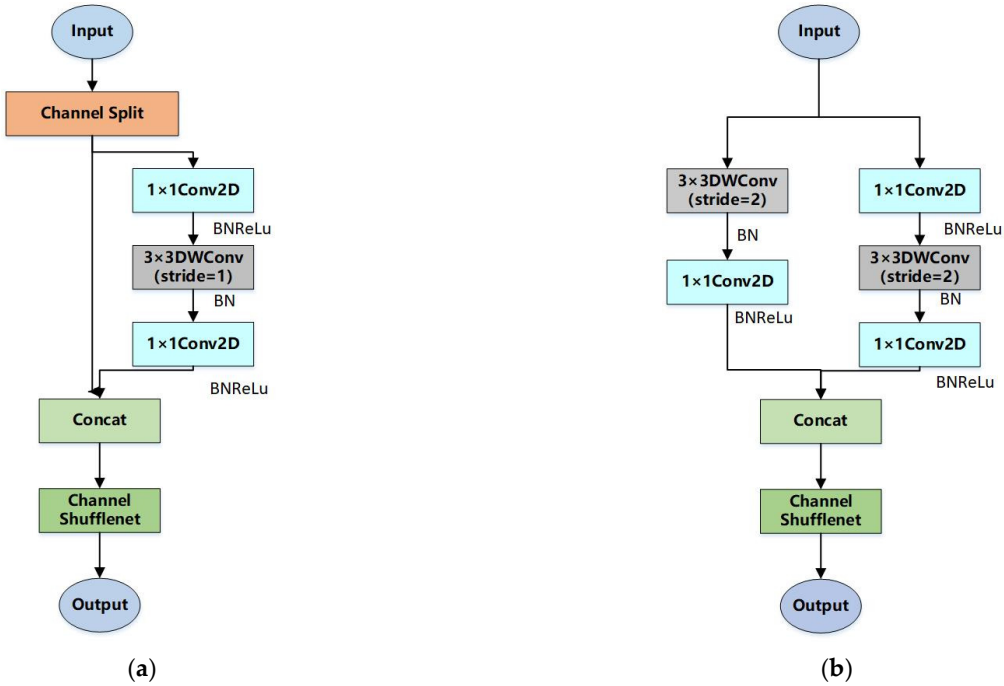


Figure 3. ShuffleNet V2 network structure (**a**) Unit1 (**b**) Unit2.

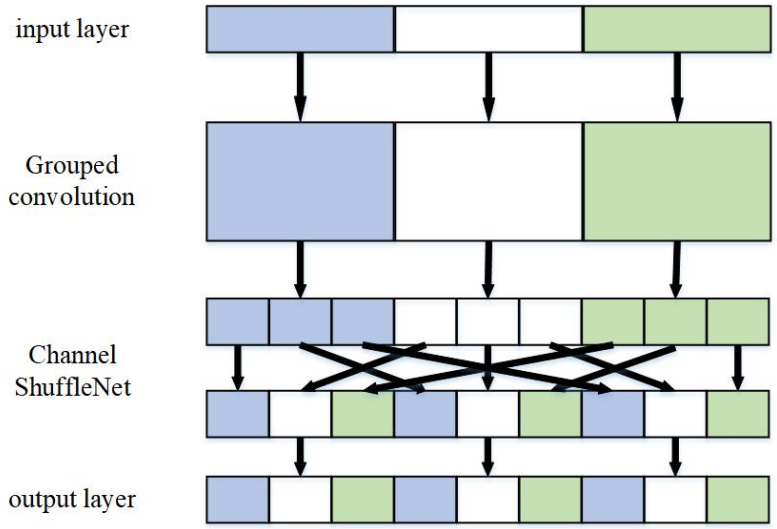


Figure 4. Channel ShuffleNet.

3.2.2. Improve Feature Extraction

Traditional Transformer models, despite their success in natural language processing, suffer from increased computational complexity and inefficiency due to the performance boost achieved through attention mechanisms. Swin Transformer addresses this issue by incorporating a sliding window operation, enabling the model to effectively capture both local and global information. In the context of detecting electric bikes in elevator scenarios, where the limited space can lead to target occlusion, this paper integrates the Swin Transformer framework into the backbone network to endow the network with powerful feature extraction capabilities, reducing the risk of missed detections.

Swin Transformer comprises primarily two components: the window multi-head self-attention mechanism layer (W-MSA) and the sliding window multi-head self-attention layer (SW-MSA), as depicted in Figure 5. Input features undergo layer normalization (LN) within the W-MSA layer for data distribution normalization. Subsequently, feature

extraction takes place through the W-MSA module. The process is then finalized by layer normalization and a multi-layer perceptron (MLP) for the integration and refinement of features.

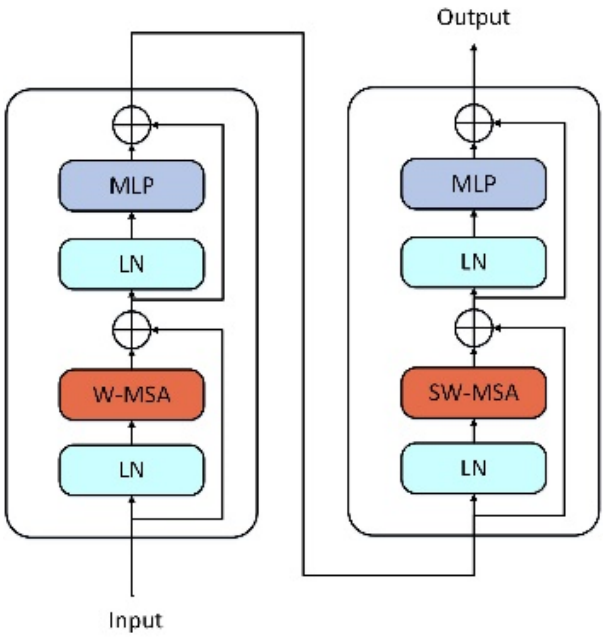


Figure 5. Swin Transformer module.

The W-MSA layer employs a windowed multi-head self-attention partitioning strategy, uniformly dividing the 8×8 feature map into 2×2 independent 4×4-sized windows for self-attention computations. However, this computation lacks connections between windows, preventing the ability to interact with texture features. Therefore, the $1 + 1$ layer structure SW-MSA is introduced, which utilizes a sliding window multi-head self-attention partitioning strategy to segment the feature map.

The SW-MSA first shifts the window based on the windowed multi-head self-attention partitioning strategy by redividing the window partitions with a window shift of $\left[\frac{M}{2}, \frac{M}{2}\right]$ $(M = 4)$, as illustrated in Figure 6. Subsequently, the windows are rearranged through a cyclic shift strategy, and the attention weights within each newly generated window are recalculated using the cross-window connection technique (MSA). Expanding the receptive field captures more contextual information, enhancing feature extraction capabilities, as shown in Figure 7.

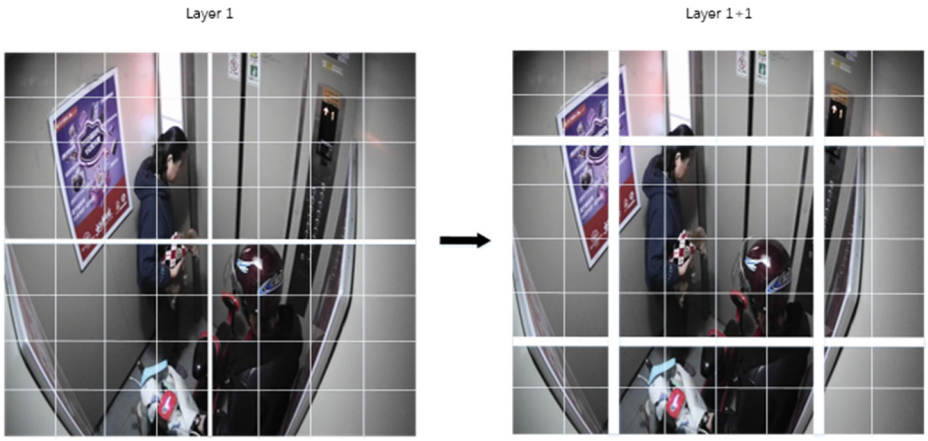


Figure 6. Swin Transformer window division.

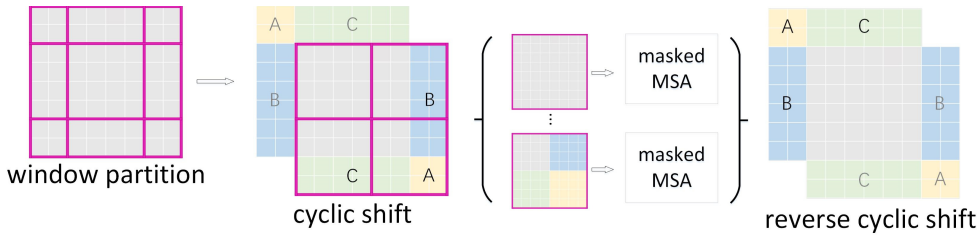


Figure 7. SW-MSA window division.

3.2.3. Introducing the SimAM Attention Mechanism Module

SimAM is a parameter-free attention mechanism. Unlike traditional channel attention, which generates one-dimensional weights, and spatial attention, which generates two-dimensional weights, SimAM is an attention mechanism with full three-dimensional weights. It can obtain different attention weights in depth, width, and height, refining features more effectively, as illustrated in Figure 8. Integrating the SimAM module into the deepest layer of the backbone network can further enhance the feature extraction capabilities without introducing additional parameters.

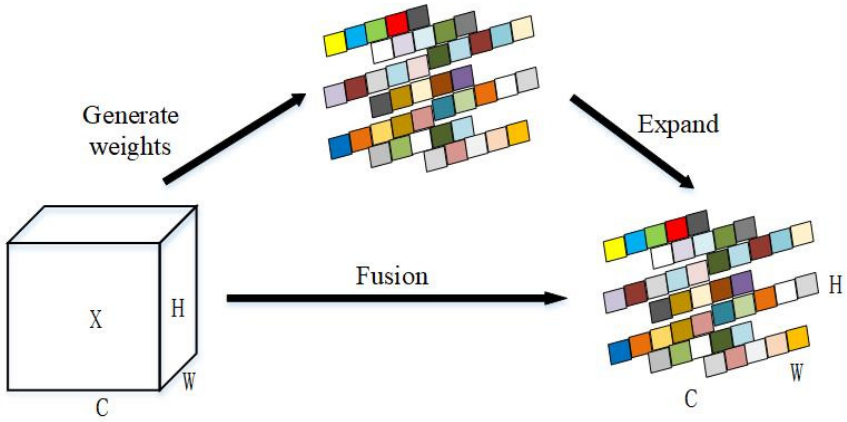


Figure 8. SimAM attention mechanism.

3.2.4. Introducing the GSConv and VOV-GSCSP Modules

The GSConv and VOV-GSCSP modules have demonstrated exceptional performance in fields such as autonomous driving, presenting a solution for rapid real-time object detection scenarios. Incorporating GSConv and VoV-GSCSP modules into the neck network enhances feature extraction and reduces computational complexity. The GSConv module's structure, illustrated in Figure 9, is initiated by generating a feature map with halved parameters through standard convolution. Subsequently, depthwise separable convolution is applied, followed by the channel concatenation of the two sets of feature maps. The module concludes with a shuffle operation to ensure channel interaction, improving information exchange between channels and enhancing the model's feature comprehension capabilities.

VoV-GSCSP is a network design based on GSConv and cross-level structures, as shown in Figure 10a. It efficiently processes and combines features from different levels through GS Bottleneck and multiple convolution operations, further optimizing feature extraction and combination. As illustrated in Figure 10b, the GS Bottleneck structure is composed of two concatenated GSConv modules and a regular convolution layer on a branch. In the neck network, VoV-GSCSP replaces the C3 structure, strengthening the interaction of information between deep and shallow features and achieving cross-scale connections in feature maps.

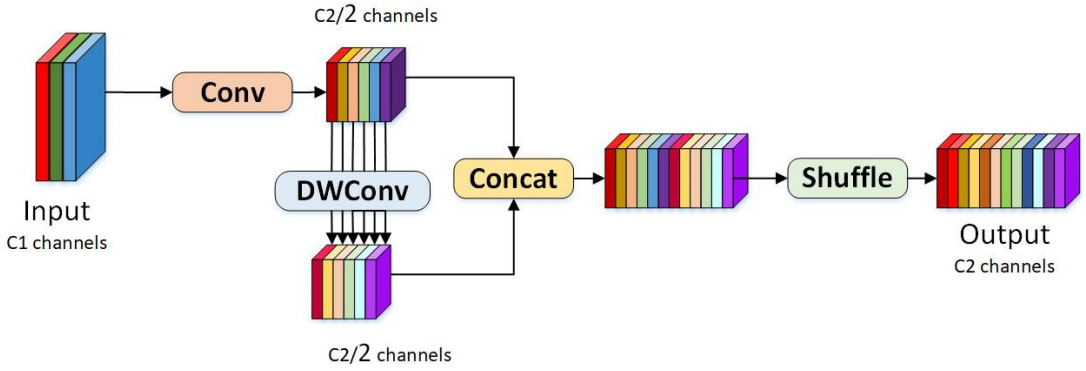


Figure 9. GSConv module.

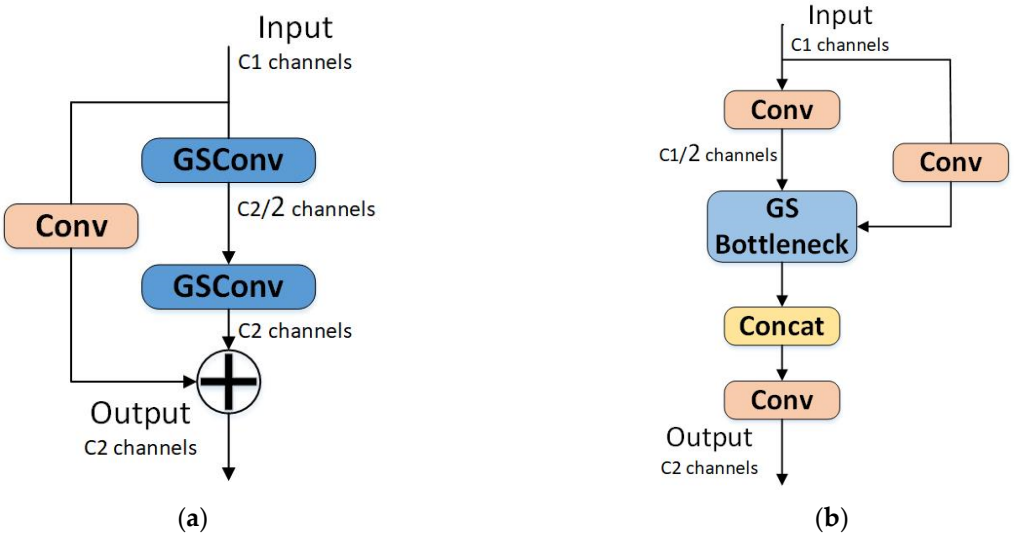


Figure 10. VoV-GSCSP module. (**a**) GS Bottleneck, (**b**) VoV-GSCSP.

3.2.5. Using the EIOU Loss Function

The YOLOv5s network employs the *CIOU* error function as the loss function, as shown in Equation (1):

$$L_{CIOU} = 1 - \left(IOU - \frac{\rho^2(b, b^{gt})}{c^2} - \alpha v \right) \tag{1}$$

This loss function comprises three components: the Intersection over Union (*IOU*) overlap loss, center distance loss $\frac{\rho^2(b,b^{gt})}{c^2}$, and aspect ratio loss αv. The aspect ratio loss component, v, reflects the difference in aspect ratios rather than the individual differences in width and height with corresponding confidences. This difference in representation sometimes hinders the effective optimization of the model's similarity.

The penalty term in *EloU* is derived by separately calculating the width and height of both the target and predicted boxes based on the influence factors of the aspect ratio. This calculation is an extension of the penalty term in *CIOU*. In Equation (2)'s terms, representing the penalty term in *EloU* directly minimizes the differences in width and height between the target box and the predicted box. This optimization supports faster convergence compared to alternative methods.

$$L_{EloU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \frac{\rho^2(w, w^{gt})}{c_w^2} + \frac{\rho^2(h, h^{gt})}{c_h^2} \tag{2}$$

In the equation, $\frac{\rho^2\left(w,w^{gt}\right)}{c_w^2}$ and $\frac{\rho^2\left(h,h^{gt}\right)}{c_h^2}$ replace the αv term in the $CIOU$ formula, where w and h are the width and height of the predicted box, w^{gt} and h^{gt} are the width and height of the ground truth box, and c_w^2 and c_h^2 are the normalized distance values for width and height. This design allows the direct calculation of the differences between the predicted and ground truth boxes in width and height when there is a significant disparity, thereby improving the localization capability for the target.

4. Experiments

4.1. Experiment Settings

In this experiment, the model training environment utilizes the Windows 10 operating system, with an AMD Ryzen 7 5800H with Radeon Graphics CPU, produced in the USA, clocked at 3.20 GHz, 16 GB of RAM, and an NVIDIA GeForce RTX 3070 GPU with 8 GB of VRAM, manufactured in the USA, is employed. The improved YOLOv5s model is based on the PyTorch deep learning framework developed by Facebook in the USA. Training is conducted in GPU mode using CUDA 11.3.1 and CUDNN 8.2.1, both provided by companies based in the USA. The training parameters are detailed in Table 1.

Table 1. Experimental parameter settings.

Parameter	Numerical Value
epochs	300
batch-size	32
learning rate	0.01
Mosaic	1.0
Weight-decay	0.0005
Img size	640 × 640

4.2. Experiment Dataset

In the experiment, due to the absence of a publicly available dataset specifically tailored to electric bikes in elevator environments, the authors manually curated 680 images from online resources. They captured an additional 320 images, resulting in 1000 images. Image dataset augmentation techniques such as cropping, flipping, scaling, and noise injection were employed to address scenarios involving occluded targets and enhance the model's generalization capabilities. After rigorous selection, the final dataset comprised 1900 images. Subsequently, Version 1.8.1 of the open-source software "labelimg" was used for image annotation. The annotated content primarily includes three classes: electric bikes, bicycles, and person. The original YOLOv5s model and the improved YOLOv5s model were both pre-trained using weight files from the COCO dataset, enhancing the models' generalization capabilities on a relatively smaller dataset.

The dataset was divided into training, testing, and validation sets in a ratio of 7:1.5:1.5. The partitioned dataset was then utilized to train the improved YOLOv5s network, generating the network model file.

4.3. Experiment Results

4.3.1. Lightweight Comparison Experiment

This study conducted comparative experiments on mainstream lightweight backbone networks, selecting MobileNet V3, GhostNet, and ShuffleNet V2 as the convolutional neural network modules for YOLOv5's backbone. A detailed comparison with YOLOv5s was performed, and the results are presented in Table 2.

Table 2 shows that when comparing the lightweight backbone networks of MobileNet V3 and GhostNet to ShuffleNet V2 regarding average precision, parameter count, computational load, and model volume, ShuffleNet V2 exhibits the highest detection average precision. It surpasses other network architectures regarding floating-point operations, computational load, and model volume, showcasing superior lightweight characteristics.

The YOLOv5s lightweight backbone feature extraction network is constructed using the ShuffleNet V2 module in this study. This variant network is referred to as YOLOv5s-Sh in the subsequent experiments.

Table 2. Comparison of lightweight network experimental results.

Model	mAP (%)	Parameters/M	FLOPs (G)	Size/MB
YOLOv5s	96.40	7.02	15.80	13.71
YOLOv5s + Mobilenet V3	77.10	1.38	2.30	11.10
YOLOv5s + GhosNet	78.80	3.68	8.00	10.00
YOLOv5s + ShuffleNet V2	86.20	0.84	1.83	2.00

4.3.2. Attention Mechanism Comparison Experiment

Attention mechanism experiments were conducted for comparison to validate the effectiveness of incorporating the SimAM module. Five mainstream attention mechanisms—CA, ECA, SE, CBAM, and SimAM—were individually added to the YOLOv5s-Sh baseline. The impact of these mechanisms on the four parameters of the model is summarized in Table 3.

Table 3. Comparison experiment of attention mechanism.

Model	mAP (%)	Parameters/M	FLOPs (G)	Size/MB
YOLOv5s-Sh	86.20	0.84	1.83	2.00
YOLOv5s-Sh + CA	88.50	0.85	1.80	2.01
YOLOv5s-Sh + ECA	84.90	0.84	1.83	2.05
YOLOv5s-Sh + SE	87.30	0.85	1.82	2.01
YOLOv5s-Sh + CBAM	85.60	0.86	1.83	2.01
YOLOv5s-Sh + SimAM	88.90	0.84	1.81	2.02

From the data presented in Table 3, it becomes evident that the integration of the SimAM module has the most substantial impact on the model's performance. By combining the ShuffleNet V2 backbone network with the SimAM module, the average detection accuracy is improved to 88.90%, which is superior to other attention mechanisms. This enhancement is achieved with a minimal increase in the model size, reaching only 2.02 MB, with the parameter count remaining nearly unchanged, and a reduction of only 0.02 G in computational load being observed. While the CA module excels in reducing computational load, its effect on average detection precision is less pronounced compared to SimAM. In summary, incorporating the SimAM module highlights its advantage in constructing efficient and lightweight object detection models. It elevates the model's detection capabilities while maintaining low computational demands.

4.3.3. Ablation Experiment

For a more intuitive validation of the impact of each improvement module in the YOLOv5s network, the authors conducted ablation experiments by sequentially adding each module. In Table 4, the first group represents the original YOLOv5s model, and "√" indicates the addition of the respective module. A higher mAP is preferable, while a smaller parameter count, model size, and computational load are desirable.

As can be seen from Table 4, after the backbone network is replaced with ShuffleNet V2, although the accuracy loss is obvious, the model volume, parameter number, and calculation amount are greatly reduced, which greatly reduces the demand for computing resources. Upon the introduction of SimAM, there was a notable improvement in average detection accuracy by 2.70%, even with minimal changes in other indicators. After incorporating the Swin Transformer, the computational load is slightly increased by 0.8 G. However, this enhancement further elevates the average detection accuracy to 91.3%. In the neck network, replacing original Conv and C3 modules with lightweight GSConv and

VOV-GSCSP modules will cause a slight increase in parameter count and model volume. However, this leads to slightly decreased calculating loads while improving the average detection accuracy. Finally, by replacing the original loss function with EIOU loss function while keeping other metrics unchanged, the average detection accuracy is elevated to 95.5%. This result confirms that the EIOU loss function can enhance detection performance without increasing the model size. With all modules replaced and introduced, the model size, parameter count, and computational load are merely 2.60 MB, 1.10 M, and 2.40 G, respectively. These experimental results demonstrate that the improved YOLOv5s network achieves a balanced combination of lightweight design and advanced detection capabilities.

Table 4. Comparison of evaluation of each module in the ablation experiment.

NO.	ShuffleNet V2	SimAM	Swin Transformer	GSconv + VoVGSCSP	EIOU	mAP (%)	Parameters/M	FLOPs (G)	Size/MB
1						96.40	7.02	15.80	13.71
2	√					86.20	0.84	1.83	2.00
3	√	√				88.90	0.84	1.81	2.02
4	√	√	√			91.30	0.86	2.61	2.10
5	√	√	√	√		93.60	1.10	2.40	2.60
6	√	√	√	√	√	95.50	1.10	2.40	2.60

4.3.4. Comparison Experiments

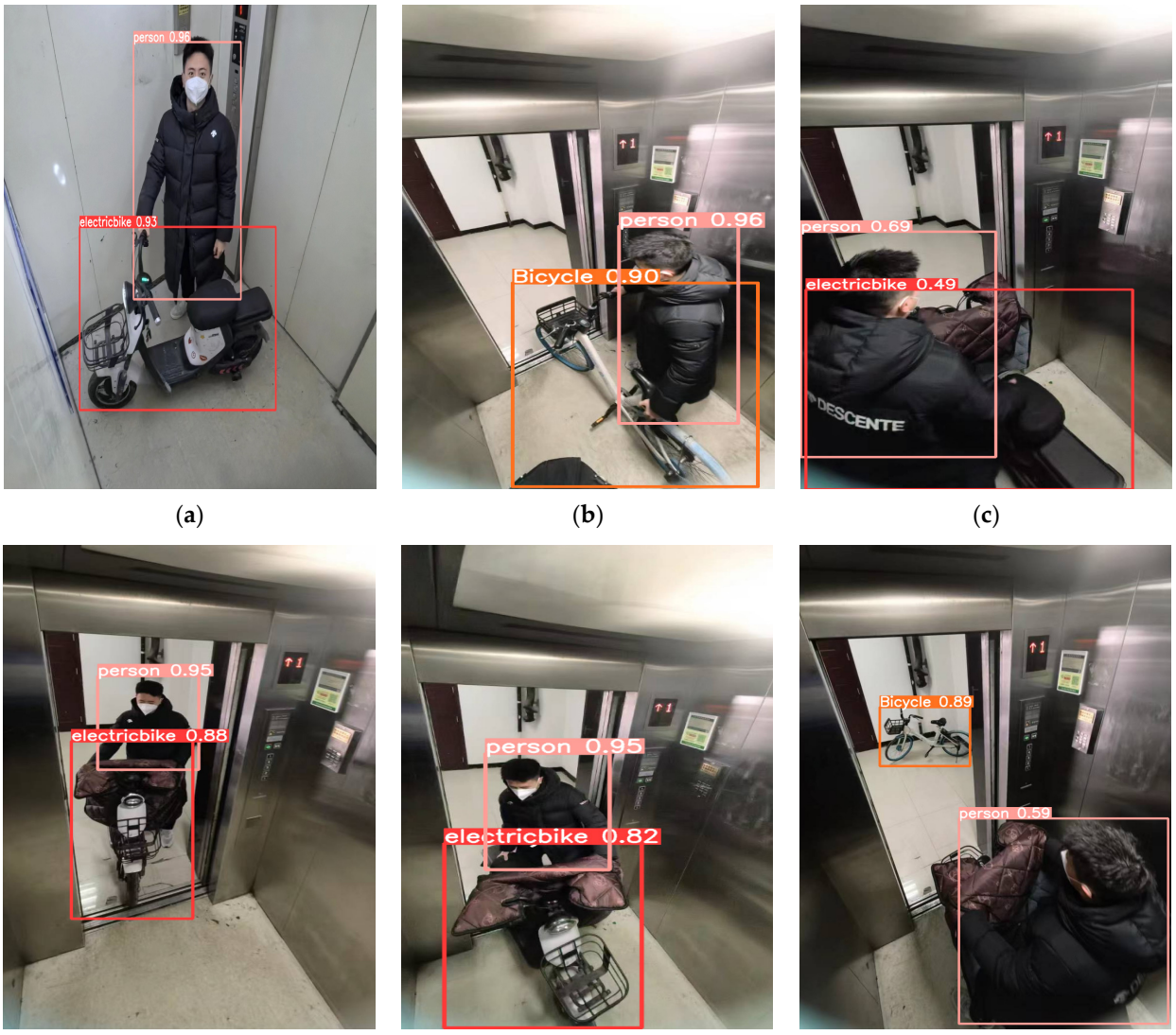
To further evaluate the object detection capabilities of the enhanced YOLOv5s network, experiments were conducted using mAP, parameter count, computational load, model size, and detection speed as performance indicators. A comparison was made with mainstream object detection models, including SSD, Faster-RCNN, YOLOv3-tiny, YOLOv4-tiny, YOLOv5s, YOLOv7-tiny, and YOLOv5s-Sh. Detailed results can be found in Table 5.

Table 5. Comparison of experimental results of mainstream algorithms.

Model	mAP (%)	Parameters/M	FLOPs (G)	Size/MB	FPS (Frame/s)
FasterR-CNN	80.50	137.00	194.30	108.00	65.62
SSD	78.30	23.70	115.70	91.60	83.71
YOLOv3-tiny	89.20	8.60	12.80	16.60	93.64
YOLOv4-tiny	93.10	6.22	15.90	12.90	102.69
YOLOv5s	96.40	7.02	15.80	13.71	75.71
YOLOv7-tiny	97.60	6.02	13.20	11.70	103.59
YOLOv5s-Sh	86.20	0.84	1.83	2.00	108.19
Ours	95.50	1.10	2.40	2.60	106.27

Table 5 shows that the YOLOv5s model has a large parameter count and slow detection speed. The YOLOv5s-Sh model, improved based on ShuffleNet V2, significantly reduces parameter count, computational load, model size, and detection speed. However, there is a substantial drop of 10.2% in average detection accuracy. For the improved model presented in this paper, compared to the original YOLOv5s network, there is only a slight loss in accuracy, a mere 0.9%, reaching 95.5%. On the other hand, there are significant optimizations in the other four key indicators. The parameter count is reduced by 84.3%, reaching only 1.10 M. The computational load decreases by 84.8%, reaching only 2.40 G. The model size shrinks by 81.0%, reaching only 2.60 MB. Additionally, the detection speed increases to 106 frames per second. While the average detection accuracy is lower than YOLOv7-tiny, the overall improvement in the YOLOv5s algorithm, considering the minimal parameter count, computational load, and model size, is evident. Through these comparisons, the algorithm's advantages in lightweight design are apparent, ensuring good real-time performance and better adaptation to the practical requirements of elevator scenes.

Figures 11 and 12 show the results of the original YOLOv5s and the lightweight improved model proposed in this paper for the recognition of people, bicycles, and electric bikes inside an elevator, respectively. Figures 11a,b and 12a,b present the recognition results for electric bikes and bicycles in unoccluded scenarios. As is evident from the figures, the lightweight improved model exhibits higher confidence in identifying electric bikes compared to the original model. Regarding bicycle recognition, the lightweight improved model also shows higher confidence than the original model, albeit at a slightly lower proportion. Figures 11c–e and 12c–e depict the recognition results for electric bikes under occluded scenarios. In comparison with the original model, the lightweight improved model demonstrates better recognition performance for occluded electric bikes. This suggests that the accuracy and robustness of the modified model have been enhanced. In Figure 11f, the original YOLOv5s model exhibits a missed detection in identifying occluded electric bikes. However, as seen in Figure 12f, the improved model successfully recognizes the electric bike, showcasing its enhanced capability to address occlusion challenges in complex scenarios. This not only reduces the risk of missed detections but also highlights its excellent generalization ability.



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

(**d**) (**e**) (**f**)

Figure 11. The YOLOv5s algorithm. (**a**) shows the detection results of electric bikes without occlusion, (**b**) displays the detection results for bicycles, and (**c**–**f**) illustrate the detection results of electric bikes with partial occlusion scenarios.

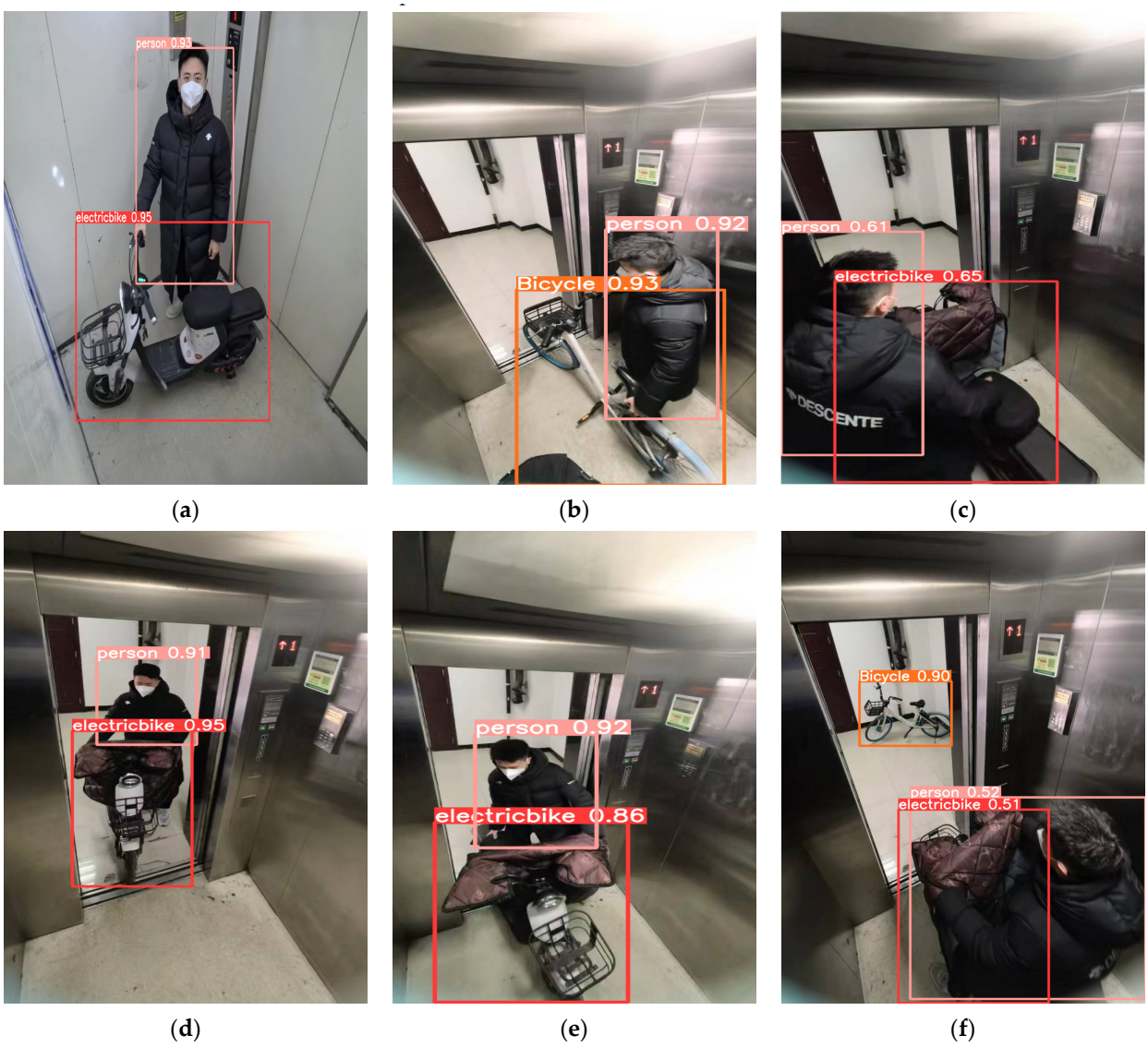


Figure 12. The improved YOLOv5s algorithm. (**a**) shows the detection results of electric bikes without occlusion, (**b**) displays the detection results for bicycles, and (**c**–**f**) illustrate the detection results of electric bikes with partial occlusion scenarios.

5. Conclusions

In order to achieve the real-time detection of electric bikes in elevator scenarios and enable the deployment of the network model on embedded edge devices, this paper proposes a method for detecting electric bikes inside elevators based on the lightweight improvement of YOLOv5s. The model employs a lightweight backbone network to reduce the model size and effectively addresses the issue of missed detections for occluded targets through improved feature extraction. It incorporates an attention mechanism to more accurately capture key image features and further utilizes a lightweight deep feature fusion module in the neck network to reduce computational complexity and improve model accuracy. Finally, the model uses an EIOU optimized bounding box regression loss function to accelerate model convergence. Through experiments on a self-built dataset, the improved model is validated, and the results demonstrate a significant reduction in both parameter count and computational complexity compared to the original YOLOv5s, with reductions of approximately 84.3% and 84.8%, respectively. Moreover, the model volume is reduced by 81.0%, while maintaining a high average detection accuracy of 95.5%. In summary, the improved network model achieves lightweight design while preserving high

accuracy, providing an efficient and practical solution for the real-time detection and edge deployment of electric bikes inside elevators.

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

Funding: This research was funded by the Youth Fund Project of the Hebei Provincial Department of Education (grant number QN2023185) and the Science and Technology Research Project of Hebei Province Colleges and Universities (grant number ZD2020318).

Data Availability Statement: Data are contained within the article.

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

References

1. Analysis of the Current Situation and Future Trends in the Layout of China's Bicycle and Electric Bicycle Industry. *China Bike* **2022**, *6*, 26–33.
2. Chen, Z. Discussion on the current situation and prevention and control measures of electric bike fires. *Fire Prot. Ind. (Electron. Ed.)* **2022**, *8*, 121–123.
3. Shao, Y.; Zhang, D.; Chu, H.; Zhang, X.; Rao, Y. Review of YOLO target detection based on deep learning. *J. Electron. Inf.* **2022**, *44*, 3697–3708.
4. Nan, X.; Ding, L. Review of typical target detection algorithms of deep learning. *Comput. Appl. Res.* **2020**, *37*, 15–21.
5. 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**, *9199*, 2969239–2969250. [CrossRef] [PubMed]
6. Zhao, Y.; Rao, Y.; Dong, S.; Zhang, J. Overview of deep learning target detection methods. *Chin. J. Image Graph.* **2020**, *25*, 629–654.
7. Zhu, Z.; Cao, J.; Hao, T.; Zhai, W.; Sun, B.; Jia, G.; Li, M. Highly secure edge-intelligent electric motorcycle management system for elevators. *J. Cloud Comput.* **2020**, *9*, 41. [CrossRef]
8. Huang, H.; Xie, X.; Zhou, L. Detection and Alarm of E-bike Intrusion in Elevator Scene. *Eng. Lett.* **2021**, *29*, EL_29_3_47.
9. Wang, W.; Xu, Y.; Xu, Z.; Zhang, C.; Li, T.; Wang, J.; Jiang, H. A Detection Method of Electro-bike in Elevators Based on Improved YOLO v4. In Proceedings of the 2021 26th International Conference on Automation and Computing (ICAC), Portsmouth, UK, 2–4 September 2021; pp. 1–6.
10. Zhang, Y.; Feng, Y. Design of electric bike detection system in elevator based on Raspberry Pi and YOLOv3. *Inf. Technol. Informatiz.* **2022**, *2*, 105–108.
11. Zhao, Z.; Li, S.; Wu, C.; Wei, X. Research on the Rapid Recognition Method of Electric Bicycles in Elevators Based on Machine Vision. *Sustainability* **2023**, *15*, 13550. [CrossRef]
12. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Hartwig, A. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**. [CrossRef]
13. 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, 2–18 June 2018; IEEE: New York, NY, USA, 2018; pp. 6848–6856.
14. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More features from cheap operations. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; IEEE Press: New York, NY, USA, 2020; pp. 1577–1586.
15. 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 2019 IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2020; IEEE Press: New York, NY, USA, 2020; pp. 1314–1324.
16. 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; Springer: Cambrige, UK, 2018; pp. 116–131.
17. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2017; IEEE Press: New York, NY, USA, 2022; pp. 9992–10002.
18. Yang, L.; Zhang, R.Y.; Li, L.; Xie, X. SimAM: A simple parameter-free attention module for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 11863–11874.

19. Li, H.; Li, J.; Wei, H.; Liu, Z.; Zhan, Z.; Ren, Q. Slim-Neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles. *arXiv* **2022**, arXiv:2206.02424.
20. Zhang, Y.F.; Ren, W.; Zhang, Z.; Jia, Z.; Wang, L.; Tan, T. Focal and efficient IOU loss for accurate bounding box regression. *Neuro-Comput.* **2022**, *506*, 146–157. [CrossRef]