Hindawi
Journal of Mathematics
Volume 2021, Article ID 5808221, 9 pages
https://doi.org/10.1155/2021/5808221

Research Article

Research on Energy Efficiency Management of Forklift Based on Improved YOLOv5 Algorithm

Zhenyu Li,[1] Ke Lu,[2] Yanhui Zhang,[3] Zongwei Li ,[1] and Jia-Bao Liu [4]

[1]*School of Economics and Management, Shanghai Institute of Technology, Shanghai 200030, China*
[2]*School of Management Science and Engineering, Anhui University of Technology, Maanshan 243032, China*
[3]*Business School, East China University of Science and Technology, Shanghai 200030, China*
[4]*School of Mathematics and Physics, Anhui Jianzhu University, Hefei 230601, China*

Correspondence should be addressed to Zongwei Li; lzw0118@163.com

Received 6 September 2021; Revised 28 September 2021; Accepted 6 December 2021; Published 21 December 2021

Academic Editor: Clemente Cesarano

As an important tool for loading, unloading, and distributing palletized goods, forklifts are widely used in different links of industrial production process. However, due to the rapid increase in the types and quantities of goods, item statistics have become a major bottleneck in production. Based on machine vision, the paper proposes a method to count the amount of goods loaded and unloaded within the working time limit to analyze the efficiency of the forklift. The proposed method includes the data preprocessing section and the object detection section. In the data preprocessing section, through operations such as framing and clustering the collected video data and using the improved image hash algorithm to remove similar images, a new dataset of forklift goods was built. In the object detection section, the attention mechanism and the replacement network layer were used to improve the performance of YOLOv5. The experimented results showed that, compared with the original YOLOv5 model, the improved model is lighter in size and faster in detection speed without loss of detection precision, which could also meet the requirements for real-time statistics on the operation efficiency of forklifts.

1. Introduction

With the continuous development of intelligent logistics centered on industrial production, the demand for machine vision is increasing. In the industrial logistics system, forklifts play an important role in transferring and storing goods. However, in most factories, due to the huge number of forklifts and the wide variety of goods, the main impediment to the traditional management methods is the inability to effectively evaluate the efficiency of forklifts.

In recent years, intelligent logistics applying machine vision and deep learning has become a research hotspot. As a basic research direction in intelligent logistics, object detection has a profound impact on energy efficiency management [1]. Himstedt and Maehle [2] proposed a forklift detection solution based on 3D camera and SVM classifier, which could accurately detect the object. However, the distance range and size range of the object needed to be preset, and the generalization ability of the model was poor. Mohamed et al. [3] combined 2D laser rangefinder and Faster R-CNN model for pallet localisation. Although the accuracy was high, the efficiency was low. Li et al. [4] used TITAN X GPU to detect forklift pallets. The detection speed was fast, but the hardware cost was expensive and the embedding effect was poor. Iinuma et al. [5] used single shot multibox detector (SSD) as a detection model. Although the model had good mobility, the detection accuracy was limited due to the insufficient features. In summary, although a large number of scholars have done extensive research on forklift object detection, in real industrial production, data collection, hardware selection, and model selection have limited the application of deep learning.

To address the above problems, achieve a balance regardless of speed, accuracy, and model size, and adapt to complex and diverse operating environments, this research improves the backbone network structure of YOLOv5 and uses a lighter feature extraction network to reduce the redundancy features. In this process, the introduction of

mechanism module maintains the detection accuracy. The experimental results showed that our model performed well on the self-built complex scene forklift goods dataset. The key contributions of this work are as follows:

(i) The YOLOv5 model is improved by combining the GhostNet module and squeeze-and-excitation attention mechanism, and then the improved model is used to detect forklifts.

(ii) The improved image hash algorithm based on PCA is used to remove similar images in image preprocessing section.

(iii) Compared with the original model, the improved YOLOv5 model reduces the amount of calculation by 2/3 while not reducing the precision.

(iv) The improved YOLOv5 model is more robust and effective for mobile edge computing devices.

2. Related Work

As one of the core segments in the domain of machine vision, object detection is a technology which digs the object potential category and location information from an image. Since there are many types of objects, the size, position, and posture of a similar object in the image are often different, and the interference caused by different imaging conditions also brings some difficulties, so object detection is full of challenges.

Before the widespread application of deep learning, traditional algorithms for object detection determined the object location and size by traversing the image using sliding windows of different sizes and simultaneously extracted artificially defined robust features, for instance, scale-invariant feature transform (SIFT) [6] and histogram of oriented gradients (HOG) [7]. Therefore, object detection combined with deep learning uses convolutional neural network to extract features to break the limitations of manual feature extraction.

2.1. Faster R-CNN. Faster R-CNN [8], originated from R-CNN [9], is widely utilized in object detection work. In R-CNN, 4 independent steps are used: candidate regions generation by selective search, feature extraction by CNN, SVM classification, and bounding box regression, which consumes a lot of time. Fast R-CNN [10] reduces the time consumed and improves accuracy through operations such as mapping candidate regions to features, ROI pooling, and FC layer. Since Fast R-CNN is not a true end-to-end work, Faster R-CNN unifies the 4 independent steps into one neural network. After that, on the basis of Faster R-CNN, many scholars proposed a variety of object detection algorithms to adapt to different tasks. The method of increasing the center loss function to reduce the intra-class variation of the learned features performed well in face detection [11]. Zhong et al. [12] replaced the bounding box regression module with the LocNet-based positioning module, which improved the positioning precision of natural scene text detection. Although the accuracy of these models has gradually reached the accuracy limit of machine vision tasks, the scale of the models has also grown exponentially. An excessive model size leads to higher requirements for hardware, which causes great resistance to achieve real-time detection in embedded devices.

2.2. YOLOv5. Compared with the R-CNN series, the most significant advantage of the YOLO (You Only Look Once) series is that they have faster detection speed. Redmon et al. [13] first proposed YOLOv1, which unified object classification and bounding box regression into a regression problem. This frame design makes YOLOv1 extremely fast in image processing, but compared with R-CNN, YOLOv1 has a larger coordinate error. Thus, Redmon and Farhadi [14] proposed YOLOv2, which improved the detection accuracy by improving the network structure and training methods. Later, on the basis of YOLOv2, Redmon and Farhadi [15] further proposed YOLOv3 by expanding the network to Darknet-53, which significantly improved the ability of small object recognition.

As the detection accuracy of YOLOv3 still has a gap with Faster R-CNN, Bochkovskiy et al. [16] proposed YOLOv4. YOLOv4 combines different detection techniques to achieve the best counterpoise between detection precision and inference speed based on a massive convincing experiments. In the same year, Ultralytics released YOLOv5. YOLOv5 is a classic representative of one-stage object detection algorithm, including four parts: Input, Backbone, Neck, and Prediction. In Input, YOLOv5, like YOLOv4, uses the mosaic method to enhance data, which is very effective for small object detection. Compared with YOLOv4, YOLOv5 not only uses Cross Stage Partial Network (CSPNet) [17] for Backbone but also uses the same for Neck to enhance feature fusion. It is also worth mentioning that YOLOv5 uses Path Aggregation Network (PAN) [18] and Feature Pyramid Network (FPN) [19] operations on Neck. FPN conveys powerful semantic features through upsampling, and PAN is used to convey dense positioning features.

YOLOv5 initially provides four object detection network models: yolov5s, yolov5m, yolov5l, and yolov5x, which contain different network depths and feature map widths. From these models, yolov5s shows its character for the lightest size and the fastest speed. On the contrary, it has the lowest average precision (AP), but it is ideal for detecting large objects. For satisfying the demands for real-time object detection on the basic processor, it is meaningful to further improve the YOLOv5 model.

3. Method

First, a monocular 2D camera is deployed on the top of the forklift cab to photograph the goods on the pallet in front of the forklift. After obtaining the video stream of the actual scene, we intercept the images at the same number of frames to form an image resource library. Then, the images are clustered, and an improved image hash algorithm is used to filter duplicate images to avoid manual filtering of differences in subjective judgments and save a lot of time cost.

The final obtained images are used as the source files of the dataset, and the category and location data are obtained through manual marking. In this paper, YOLOv5 is used as the machine vision detection algorithm, and the network framework is improved to achieve real-time and accurate acquisition of the forklift transportation status. Our object detection method is demonstrated in Figure 1.

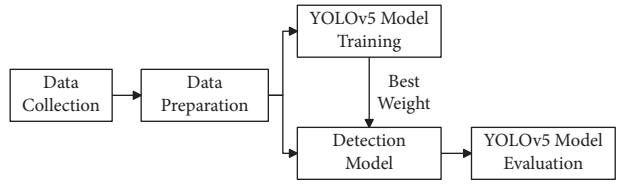

Figure 1: Object detection process.

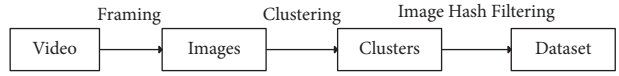

Figure 2: Data preprocessing.

3.1. Data Preprocessing. This paper constructs a forklift dataset to detect the status of goods. Video was obtained by following the driver's driving process in the field workshop. Complex samples of different weather conditions, different time periods, and different locations were collected. Through the operations shown in Figure 2, a dataset containing four different statuses of full tray, half tray, empty, and loading-unloading was constructed. The self-built forklift dataset is close to the complex and changeable industrial reality scene, which poses greater challenges to the network performance of object detection.

Since the amount of data after framing is large and there are many similar images, the workload of direct deletion is too large. Therefore, a clustering algorithm can be used to peel off the semantic information of the images. For balancing the clustering effect and computing time, the number of clusters in this experiment was set to 9. After clustering, it is easy to delete images.

After clustering, the images in the same cluster are relatively similar, and a large-scale comparison is required to eliminate similar images. Hash algorithm [20], as a single mapping function, can compress a fixed-size input into a fixed-length output, which has the advantages of improving storage data utilization and improving data query efficiency. The image hash algorithm [21] takes the human visual system as a reference to extract the perceptual robust features in the image and map images with the same visual perception to the same or similar hash value. For different visual perception images, the hash algorithm generates completely different hash values.

Image hash algorithm based on principal component analysis (PCA) can quickly generate the image hash values [22]. First, the original image with a size of 608×304 is subjected to grayscale processing and a filter is used to eliminate the noise of image. Then, the image is segmented into 32 nonoverlapping image fragments with a size of 76×76. The pixels of the image fragment are connected in the order of left to right and top to bottom to construct 32 5776-dimensional vectors. Because the vector dimension is too high, the calculation speed will be reduced, so PCA is used to reduce the data dimension to 10 dimensions by the following equation:

$$p^k \times v = v^k, \tag{1}$$

where p^k is the base and v is the high-dimensional vector representing the image. v is mapped to p^k to obtain the reduced dimensionality target v^k.

Finally, a secret key is designed to generate a hash value, and a 32-dimensional feature vector is output to represent the original image. Figure 3 typically illustrates the circuit of the image hash algorithm.

The correlation coefficient between the hash values of different images is calculated in the same cluster, and a threshold is set to filter similar images to solve the problem of self-built dataset redundancy. The similarity function is given by the following equation:

$$c(h_1, h_2) = \frac{\operatorname{cov}(h_1, h_2)}{\sqrt{\operatorname{var}(h_1)}\sqrt{\operatorname{var}(h_2)}}, \tag{2}$$

where h_1 is the hash value of image 1, h_2 is the hash value of image 2, $\operatorname{var}(h_1)$ is the variance of h_1, $\operatorname{var}(h_2)$ is the variance of h_2, and $\operatorname{cov}(h_1, h_2)$ is the covariance between h_1 and h_2.

3.2. Improved YOLOv5 Model. Although the accuracy of the original YOLOv5 model meets our demand for forklift object detection, the detection speed needs to be improved for embedded devices and mobile terminal operations with limited computing power. Based on the analysis of YOLOv5 network structure, a new lightweight object detection model is rebuilt in this research. The modified model uses GhostBottleneck (GB) module to replace the original network layers and introduces Squeeze-and-Excitation (SE) attention mechanism. While improving the detection speed and making the model more miniature, this model can ensure the accuracy of the detection.

3.2.1. GhostBottleneck Module. Aiming at settling the problem of limited computing power of mobile devices, we adopt the GhostNet [23] structure specially designed for mobile devices. The core of GhostNet is to generate rich feature maps using linear operations. In the convolution module of the original YOLOv5 network, feature extraction produces too many similar redundant feature maps. The GB module used in this paper first uses ordinary convolution to obtain partial feature maps and then performs linear convolution operations to amplify them to the same number of feature maps as the original network. At the same time, because the calculation amount of linear convolution is much smaller than that of ordinary convolution, the calculation amount of the

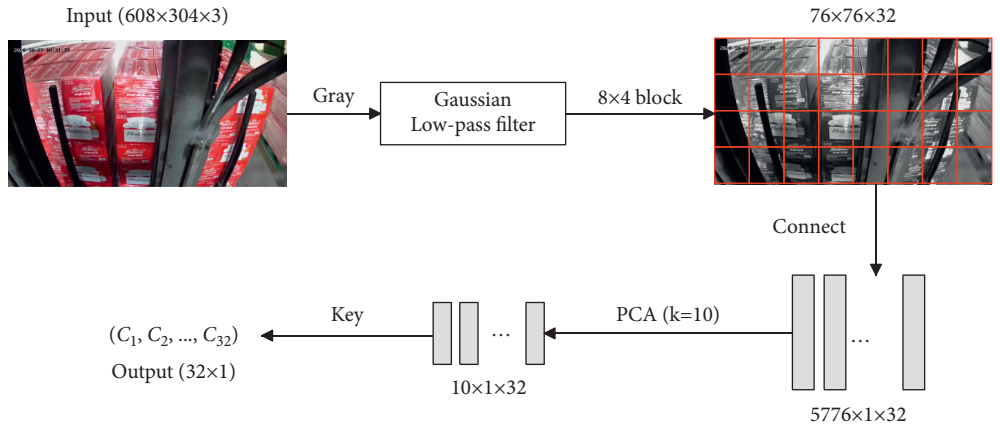

Figure 3: Image hash algorithm based on PCA.

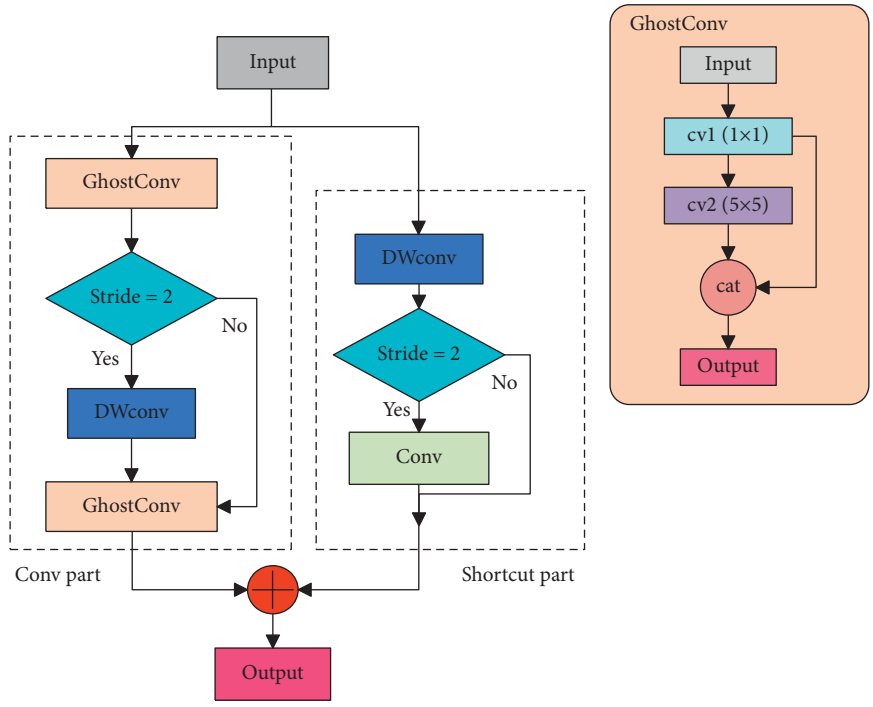

Figure 4: GhostBottleneck module.

model is reduced by about half. The GB module is divided into a Conv part and a Shortcut part, with the framework shown in Figure 4.

Figure 4 can be noticed that the feature map is used as the input of GB module. In the Conv part, the first GhostConv layer is used to realize the channel expansion, and then the second GhostConv layer is executed to match the Shortcut part. Due to the divergence of the gradient, simply deepening the network can hardly ensure the improvement of network performance. Actually, Shortcut part and Conv part are added as the output, which adaptively adjusts the quantity of network output channels while ensuring the effect of the model.

GhostConv in the GB module is connected by two different convolutional layers, cv1 and cv2. First, the cv1 layer uses a 1×1 convolution kernel to achieve deeper feature extraction. Then, the cv2 layer uses a 5×5 convolution kernel to separate multiscale local feature information through linear transformation. Finally, the results of cv1 layer and cv2 layer are connected and output together. The GhostConv network guarantees the convolution effect through grouped convolution while greatly reducing the model complexity.

3.2.2. Squeeze-and-Excitation Module. The forklift pallet occupies a large area in the image, and all channels are of the same importance. There is still room for improvement of detection accuracy in this aspect. SE block was proposed by Hu et al. [24], which adaptively adjusts the feature responses of different channels by paying attention to the relationship between channels.

The SE module includes two parts, Squeeze and Excitation. After the continuous convolution stacking of the GB layer, problems such as model overfitting may occur. In the Squeeze part, the global feature is generated by performing global average pooling operation on the feature map layer. Then, the entire network is regularized to prevent overfitting. The output of $1 \times 1 \times C$ is given by the following equation:

$$z_c = F_{\mathrm{sq}}(U_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j), \tag{3}$$

where u_c is the result of the previous layer of convolution and H and W denote the height and width of the feature map, separately.

Subsequently, the Excitation part obtains the connection between the channels by connecting the FC layer. The equation is as follows:

$$F_{\mathrm{ex}}(z, W) = \sigma(g(z, W)) = \sigma\left(W_2 \mathrm{ReLU}\left(W_1 z\right)\right), \tag{4}$$

where W_1 is the parameter of dimensionality reduction layer and W_2 is the parameter of dimensionality enhancement layer. Such an operation balances performance and calculation. To guarantee that the weight of the output is between 0 and 1, the sigmoid activation function is chosen.

Finally, in the scale layer, the normalized weights are multiplied with the original features for output. On our self-built dataset, the SE layer is used to extract more directional features. Although SE block inevitably increases some parameters and calculations, the improved network structure shows better performance.

The improved YOLOv5 model framework in this paper is mainly composed of Input, Backbone, Neck, and Prediction. First, Backbone is utilized to refine fine-grained features of different input images to obtain rich semantic information and location information. Then, the design of FPN + PAN occupies Neck. The FPN of path combination uses upsampling to fuse the features extracted by Backbone to convey strong semantic features. PAN's feature pyramid structure strengthens the model to convey strong positioning features, which is conducive to the detection of an object at different scales. Finally, the Prediction part predicts the bounding box, category, and other information and maps them to the corresponding image. After replacing the network layer with the GhostBottleneck module and introducing the attention mechanism, we cut the quantity of parameters sharply and lower the complexity of the model effectively, while maintaining the precision compared with the original model. The overall improved YOLOv5 model is shown in Figure 5, and the computational complexity is 5.6 GFLOPS.

4. Experiment and Discussion

4.1. Experimental Environment. In this research, two different configurations were used for model training and testing. Table 1 lists the specific configuration of the training environment.

After obtaining the weights after training, the model was deployed on the mobile edge computing device Jetson Nano for performance testing. The specific information of the device is shown in Table 2. The experimental environment was close to the actual application scenario.

4.2. Training Result Analysis. For objective evaluation, we compared the improved yolov5s model with the original YOLOv5 v3.0 yolov5s model and the YOLOv5 v4.0 yolov5s model on the self-built dataset. The only difference was that YOLOv5 v3.0 used the BottleneckCSP module, and YOLOv5 v4.0 used the C3 module, so we called them the former yolov5s_CSP, the latter yolov5s_C3, and our model yolov5s_GS. Table 3 gathers and compares layers, parameters, and GFLOPS of the three different models.

According to Table 3, our model network was built in a deeper manner through the improvement of the backbone network, while the model parameters were reduced by about 2/3, thereby reaching the goal of model complexity reduction effectively.

4.2.1. Indexes and Training Details. The most commonly used indexes for quantitatively evaluating the effectiveness of object detection algorithms are precision and recall, which are expressed by equations (5) and (6):

$$\mathrm{precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{5}$$

$$\mathrm{recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{6}$$

where TP refers to the quantity of objects that we judged correctly, FP refers to the quantity of objects that we judged incorrectly, and FN refers to the quantity of objects that we should have judged correctly but missed.

This paper uses mAP@0.5 and mAP@0.5 : 0.95, which are related to both precision and recall, as indexes to quantitatively judge whether the object detection methods meet accuracy and speed requirements [25].

The training process was monitored, and in each iteration, mAP@0.5 and mAP@0.5 : 0.95 were calculated. After spending 0.732 h to train yolov5s_C3, 0.758 h to train yolov5s_CSP, and 0.849 h to train our model yolov5s_GS, we obtained two line graphs of the three models of mAP, as shown in Figure 6. The figure reveals that our model had less fluctuation and faster convergence, compared with the original YOLOv5 model.

At the same time, the cls_loss (class loss) and obj_loss (object loss) [26] of each iteration in the training process are shown in Figure 7, indicating the good convergence of our model.

4.3. Performance Test on Mobile Devices. As a small computer, Jetson Nano has good computing power that can complete object detection tasks, and its small size can also meet the needs of embedded development and mobile terminal operation. The model was deployed on Jetson Nano to simulate the object detection reasoning process in real industrial scenarios. In Table 4, the performance indexes of different models are displayed.

On mAP@0.5 : 0.95, it can been found from Table 4 that our model yolov5s_GS is only about 1.2% lower than the best

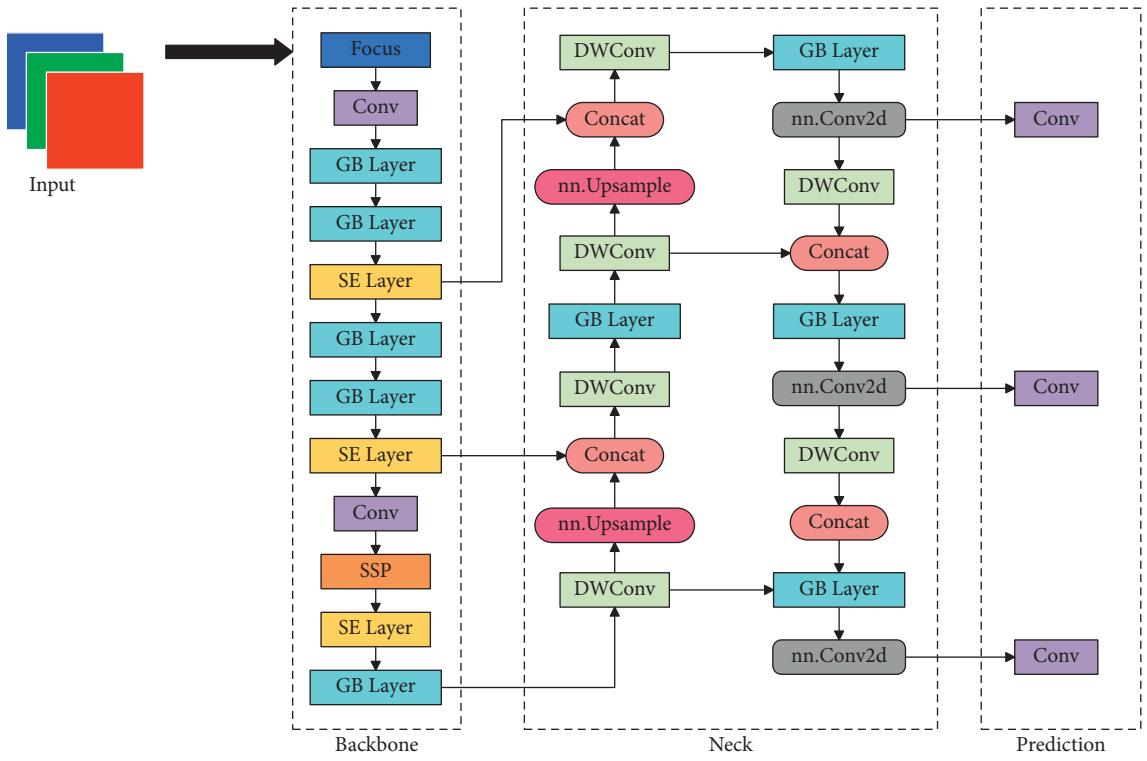

Figure 5: Improved YOLOv5 network.

Table 1: Configuration of training environment.

Item	Item value
Operation system	Ubuntu18.04
CPU	Intel Core i9-10980XE @ 3.00 GHz
GPU	GeForce RTX 3070
Hardware acceleration	CUDA10.1

Table 2: Configuration of inferencing environment.

Item	Items value
Operation system	Ubuntu18.04
CPU	4-core ARM A57 @1.43 GHz
GPU	128-core Maxwell
Hardware acceleration	CUDA10.1

Table 3: Models compared.

Model	Layers	Parameters	GFLOPS
yolov5s_C3	283	7071633	16.4
yolov5s_CSP	283	7263185	16.8
yolov5s_GS	419	2551101	5.6

performing model yolov5s_C3, while yolov5s_GS is about 0.85% higher than yolov5s_C3 on mAP@0.5%. In terms of weight, it can be seen that the size of our model after training was only 5.4 MB. From the perspective of detection time, the detection time of our model was reduced to 0.118 s/frame compared with the original network. At the same time, larger frames per second (FPS) also mean that our model could detect more images per second.

Combined with actual application scenarios, our model realized embedded development and met the requirements of real-time detection. Compared with the original YOLOv5 model, the size of our model is reduced to 1/3, and the detection speed is significantly expedited without reducing the detection precision. It can be found from Figure 8 that in complex industrial scenarios, our improved model was more robust.

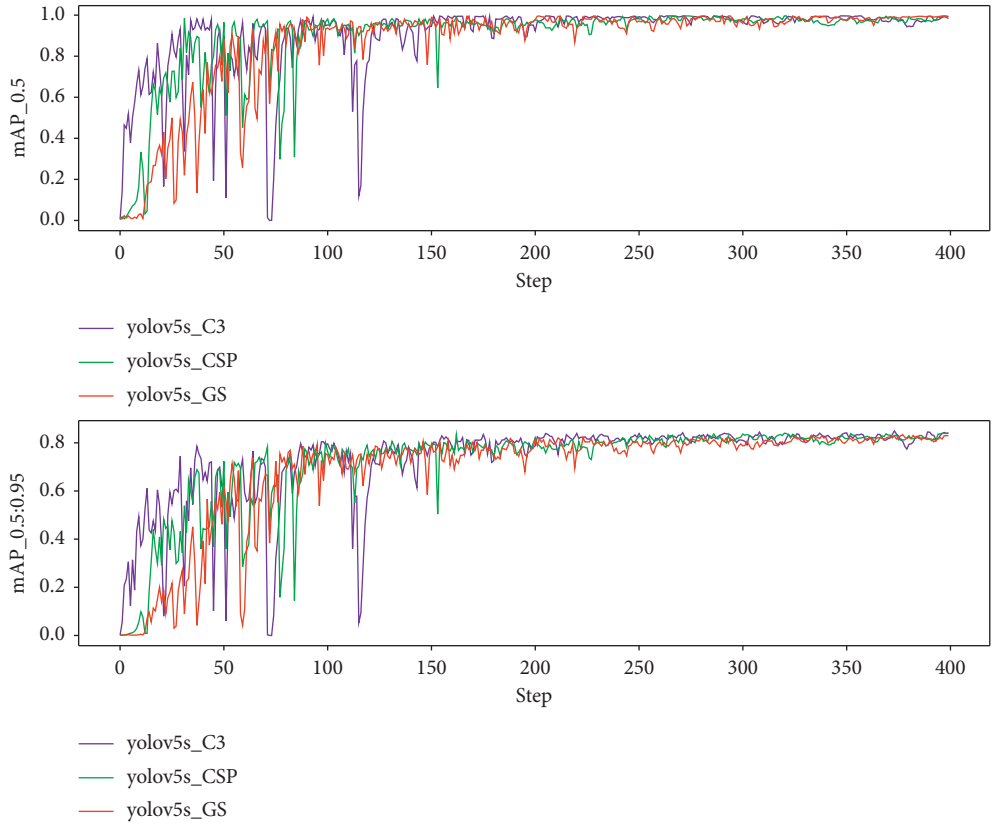

Figure 6: mAP of different models.

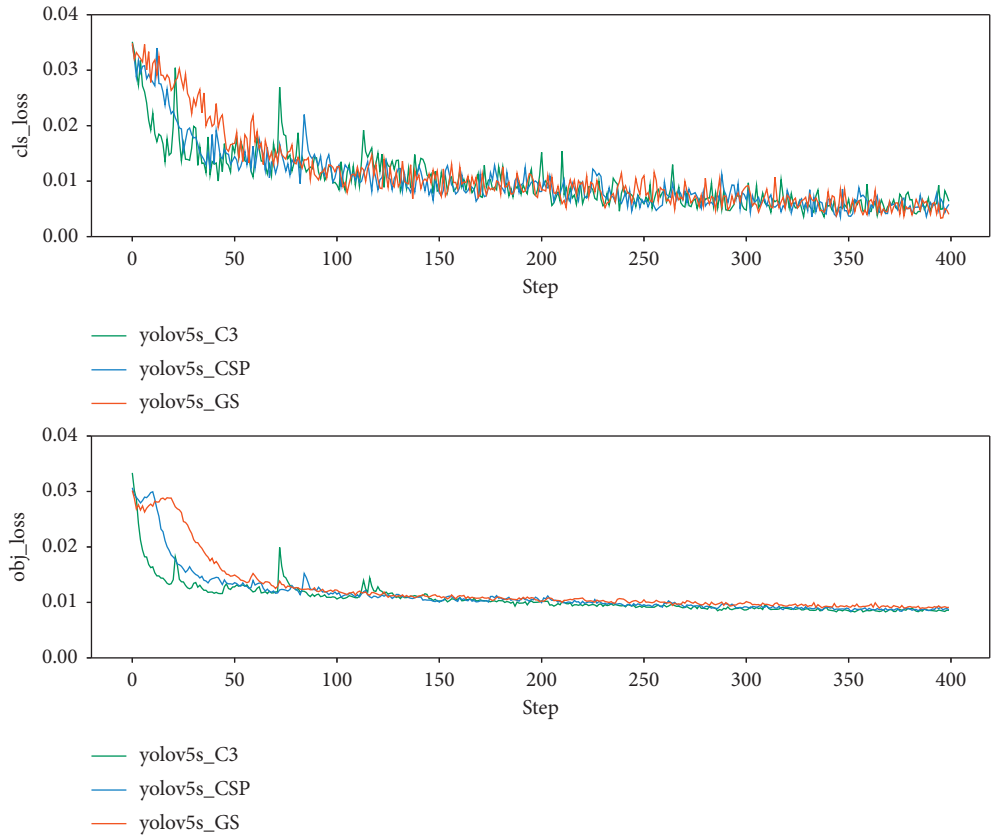

Figure 7: Loss of different models.

Table 4: Performance indexes of different models.

Model	mAP0.5	mAP0.5 : 0.95	Model size (MB)	Time (s)	FPS
yolov5s_C3	0.98321	0.84177	14.4	0.146	6.85
yolov5s_CSP	0.99371	0.84001	14.8	0.155	6.45
yolov5s_GS	0.99172	0.82959	5.4	0.118	8.47

(a) (b) (c)

Figure 8: Real scene forklift detection images of yolov5s_C3 (a), yolov5s_CSP (b), and yolov5s_GS (c).

5. Conclusions

We present an improved object detection method in this paper which can be applied to forklifts. First, a complex scene forklift goods dataset is constructed. The reason why YOLOv5 is chosen as the object detection algorithm is that compared with Faster R-CNN, YOLOv5 has faster detection speed, smaller model, and lower hardware requirements, which is suitable for mobile device operation and embedded development. Then, in the object detection section, specific modifications are made to the YOLOv5 model, which further enhance the detection speed of YOLOv5 and reduce the model size compared to the original model while maintaining the detection accuracy. Finally, our proposed method performs well on forklift object detection tasks. Due to being lightweight and having extremely fast speed, our method is also fit for other scenarios restricted by hardware resources and applications that have high requirements for real-time detection, such as mobile device QR code positioning, natural scene text detection, and autonomous driving. In the future, we will also consider migrating this method to other fields to orient diverse and complex object detection tasks.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (71974130), the National Social Science Fund of China (18BGL093), and the Shanghai Pujiang Program (2019PJC096). The authors are thankful for the support.

References

[1] A. Al-Shaebi, N. Khader, H. Daoud, J. Weiss, and S. W. Yoon, "The effect of forklift driver behavior on energy consumption and productivity," *Procedia Manufacturing*, vol. 11, pp. 778–786, 2017.

[2] M. Himstedt and E. Maehle, "Camera-based obstacle classification for automated reach trucks using deep learning," in *Proceedings of the ISR 2016: 47st International Symposium on Robotics*, pp. 1–6, Munich, Germany, June 2016.

[3] I. S. Mohamed, A. Capitanelli, F. Mastrogiovanni, S. Rovetta, and R. Zaccaria, "Detection, localisation and tracking of pallets using machine learning techniques and 2D range data," *Neural Computing & Applications*, vol. 32, no. 13, pp. 8811–8828, 2019.

[4] T. Li, B. Huang, C. Li, and M. Huang, "Application of convolution neural network object detection algorithm in logistics warehouse," *Journal of Engineering*, vol. 2019, no. 23, pp. 9053–9058, 2019.

[5] R. Iinuma, Y. Kojima, H. Onoyama, T. Fukao, S. Hattori, and Y. Nonogaki, "Pallet handling system with an autonomous forklift for outdoor fields," *Journal of Robotics and Mechatronics*, vol. 32, no. 5, pp. 1071–1079, 2020.

[6] T. Lindeberg, "Scale invariant feature transform," *Scholarpedia*, vol. 7, no. 5, Article ID 10491, 2012.

[7] P. E. Rybski, D. Huber, D. D. Morris, and R. Hoffman, "Visual classification of coarse vehicle orientation using histogram of oriented gradients features," in *Proceedings of the 2010 IEEE Intelligent Vehicles Symposium*, pp. 921–928, La Jolla, CA, USA, June 2010.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, Columbus, OH, USA, June 2014.

[10] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, Santiago, Chile, December 2015.

[11] H. Wang, Z. Li, X. Ji, and Y. Wang, "Face R-CNN," 2017, https://arxiv.org/abs/1706.01061.

[12] Z. Zhong, L. Sun, and Q. Huo, "Improved localization accuracy by LocNet for Faster R-CNN based text detection in natural scene images," *Pattern Recognition*, vol. 96, Article ID 106986, 2017.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, Las Vegas, NV, USA, June 2016.

[14] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, Honolulu, HI, USA, July 2017.

[15] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[16] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: optimal speed and accuracy of object detection," 2020, https://arxiv.org/abs/2004.10934.

[17] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: a new backbone that can enhance learning capability of CNN," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 390-391, Seattle, WA, USA, June 2020.

[18] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8759–8768, Salt Lake City, UT, USA, June 2018.

[19] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, Honolulu, HI, USA, July 2017.

[20] D. Eastlake and P. Jones, "US secure hash algorithm 1 (SHA1)," *IETF Request for Comments*, vol. 3174, 2001.

[21] R. Venkatesan, S.-M. Koon, M. H. Jakubowski, and P. Moulin, "Robust image hashing," in *Proceedings of the 2000 International Conference on Image Processing*, pp. 664–666, Vancouver, BC, Canada, September 2000.

[22] X. Liang, Z. Tang, X. Xie, J. Wu, and X. Zhang, "Robust and fast image hashing with two-dimensional PCA," *Multimedia Systems*, vol. 27, no. 3, pp. 389–401, 2021.

[23] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: more features from cheap operations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1580–1589, Seattle, WA, USA, June 2020.

[24] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, Salt Lake City, UT, USA, June 2018.

[25] F. Borisyuk, A. Gordo, and V. Sivakumar, "Rosetta: large scale system for text detection and recognition in images," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 71–79, New York, NY, USA, July 2018.

[26] S. Hoory, T. Shapira, A. Shabtai, and Y. Elovici, "Dynamic adversarial patch for evading object detection models," 2020, https://arxiv.org/abs/2010.13070.