

Research Article

A Combined Detection Algorithm for Personal Protective Equipment Based on Lightweight YOLOv4 Model

Li Ma , Xinxin Li , Xinguan Dai , Zhibin Guan , and Yuanmeng Lu 

College of Communication and Information Engineering, Xi'an University of Science and Technology, Xi'an, 710600 Shaanxi, China

Correspondence should be addressed to Xinxin Li; 1505011424@qq.com

Received 27 February 2022; Accepted 13 April 2022; Published 4 May 2022

Academic Editor: Maode Ma

Copyright © 2022 Li Ma et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Improper wearing of personal protective equipment may lead to safety incidents; this paper proposes a combined detection algorithm for personal protective equipment based on the lightweight YOLOv4 model for mobile terminals. To ensure high detection accuracy, a channel and layer pruning method (CLSlim) to lightweight algorithm is used to reduce computing power consumption and improve the detection speed on the basis of the YOLOv4 network. This method applies L1 regularization and gradient sparse training on the scaling factor of the BN layer in the convolutional module: global pruning threshold and local safety threshold are used to eliminate redundant channels, the layer pruning threshold is used to prune the structure of the shortcuts in the Cross Stage Partial (CSP) module for inference speed improvement, and finally, a lightweight network model is obtained. The experiment improves the YOLOv4 and YOLOv4-Tiny models for CLSlim lightweight separately in GTX2080ti environment. Results show that (1) CLSlim-YOLOv4 compresses the YOLOv4 model parameters by 98.2% and increases the inference speed by 1.8 times with mAP loss of only 2.1% and (2) CLSlim-YOLOv4-Tiny compresses the original model parameters by 74.3% and increases the inference speed by 1.1 times with mAP increase of 0.8%, which certifies that this improved lightweight algorithm serves better for the real-time ability and accuracy of combined detection on PPE with mobile terminals.

1. Introduction

Personal protective equipment (PPE) is equipment for workers avoiding or lightening accident injury at work. Common PPEs include safety hard hats, reflective clothing, and protective clothing in construction scenes [1]. OSHA (occupational safety and health administration) stipulates that workers must wear safety hard hats when entering the construction site, and special types of work shall wear appropriate personal protective equipment. Workers working at heights must wear safety hard hats and safety belts [2]. Outdoor workers shall wear safety hard hats and reflective clothes, etc. [3]. The traditional image-based PPE detection algorithm needs to extract the key region features first and then use the edge information or classification algorithm to recognize the PPE. Reference [4] uses the template matching method to judge personnel wear safety belts. Reference [5] uses edge contour information to identify hard hats. In Reference [6], it demonstrates the application of Artificial Intel-

ligence (AI) and machine vision for the identification of personal protective equipment (PPE), particularly safety glasses in zones of the learning factory, where safety risks exist. Traditional PPE detection methods have the disadvantages of low precision and slow speed. However, with the rapid development of convolutional neural networks in the field of machine vision, many scholars use end-to-end target detection algorithms to detect PPE and achieve good results. Reference [7] uses the SSD target detection algorithm to detect the hard hat in real time and recognize its color information. In Reference [8], a convolutional neural network is used to identify workers and hard hats, and the normalization of wearing hard hats according to the overlap value of workers' heads and hard hats is verified.

Due to the complexity of the construction environment, workers wearing a single PPE could not fully protect their own safety, while the combined detection algorithm of multiple types of PPEs needs to verify the standardization of use at the same time. The verification method proposed in

Reference [8] will increase exponentially with the increase of PPE components, which will affect the recognition speed. Based on the YOLOv3 algorithm, Reference [9] detects multiple types of PPEs and verifies the standardization of wearing, which has high real-time performance but poor recognition accuracy for small targets or low resolution picture. At present, there are many ideas worthy of reference in academic circles to improve the detection accuracy of the algorithm, such as data enhancement methods that only increase the training cost without affecting the inference speed or inserting attention mechanism modules that only increase a small amount of reasoning cost in the training process [10, 11]. In 2020, Bochkovskiy et al. proposed their YOLOv4 algorithm [12]. Combining the popular convolutional network optimization techniques and using more complex network structures, this algorithm was able to proceed with fast and accurate training and detection on lower configuration servers and was identified as an excellent target detection algorithm. But the huge model and parameter calculation volumes make it not suitable for mobile ends in industrial scenarios with limited resources. Therefore, under the premise of ensuring high detection accuracy, reducing floating point operations, improving the inference speed, and making it deployable to mobile terminals with limited resources are an urgent problem that needs to be solved. By using the channel pruning method, the parameter quantity of YOLOv3 is compressed by 92%. Reference [13] maintains the detection accuracy of the original model and improves the inference speed by twice. By channel pruning of the improved YOLOv4 model, the algorithm in Reference [14] lost 2.43% of the detection accuracy, increased the prediction speed by 2.9 times, and compressed the model by 96%. By designing a lightweight convolutional neural network (CNN) which is named as Shuffle CNN, a Shuffle CNN-based AMC (Shuffle AMC) method is proposed for the ubiquitous IoT cyberphysical systems with orthogonal frequency division multiplexing (OFDM) in Reference [15]. For facial landmark detection, Reference [16] presents a novel loss function to train a lightweight student network (e.g., MobileNetV2).

To solve the problems of combined detection on the workers' multiple PPEs and improve the real-time performance and detection accuracy of terminals with limited resources in complex networks, this paper proposes a high accuracy PPE real-time detection algorithm with a smaller volume. Based on the popular YOLOv4 and YOLOv4-Tiny networks for model lightweight, it compresses the model efficiently by combining channel and layer pruning methods and gets a combined detection algorithm on PPEs with small volume and fast detection speed, which is suitable for mobile ends in industrial scenarios.

2. Improved YOLOv4 for PPE Real-Time Detection Algorithm

As the improved version on v3, YOLOv4 integrates the idea of the convolutional neural network algorithm based on the original YOLO frame and uses many strategies on the backbone network of feature extraction, neck network of feature

fusion and the detection head of classification, and regression for the improvement of the v3 algorithm.

The YOLOv4 network structure is shown in Figure 1, and CSPDarknet53 is used as the backbone network. In the structure, the CSP structure can be lightweight and simultaneously improve the learning ability of CNN, reduce computing bottlenecks, and reduce memory costs [17]. CBM and CBL are joined up with batch normalization (BN) operation after regular convolution (Conv), with commonly used activation functions of Leaky ReLU, Mish, etc. Before feature fusion, the SPP module is introduced, which can effectively increase the network receptive field and obviously separate the contextual features compared to max pooling operation. The Path Aggregation Network (PANet) is the enhanced feature pyramid network [18], which effectively improves the problem of losing shallow feature information in the deep network by combining the methods of bottom-up and top-down paths [19].

To improve real-time performance, a lightweight algorithm YOLOv4-Tiny is proposed on the basis of YOLOv4, which is showed in Figure 2. In this YOLOv4-Tiny lightweight network model, three residual modules are used in the CSPDarknet53 backbone network, the Leaky ReLU function is used as the activation function, the FPN network is used in the multiscale feature fusion module, and two detection heads are used in classification and regression of the prediction module.

2.1. Activation Functions for the Modification of Class Probability. In regular target detection tasks, one object may belong to multiple categories as Figure 3(a). When there are many overlapping categories in the data set, a single detection box can be used to detect multiple classes simultaneously (e.g., person and male and dog and pug). Therefore, the single-label classification method has limitations in real scenes; the original YOLOv4 algorithm supposes that all classes are nonmutually exclusive. And the activation function sigmoid is used for the calculation of class probability as shown in

$$\sigma_{\text{sigmoid}}(z_i) = \frac{e^{z_i}}{e^{z_i} + 1}. \quad (1)$$

The function processes each class i independently and normalizes the prediction probability z_i of each class between $[0, 1]$. If $\sigma_{\text{sigmoid}}(z_i)$ is bigger than a certain threshold, like 0.5, there is a class in the grid cell; that is to say, an object can be predicted as multiple classes. And this paper focuses on the combined detection of workers wearing different classes of PPEs as shown in Figure 3(b). A worker can only belong to one category. For instance, the semantic definition of the classes is given as W, WH, WV, and WHV. If one worker is detected as a certain class (i.e., WHV), the other classes (i.e., W, WH, and WV) of the same target will be replaced. Therefore, the activation function SoftMax is used to calculate the class probabilities as shown in

$$\sigma_{\text{SoftMax}}(z)_i = \frac{e^{z_i}}{\sum_{i=1} e^{z_i}}. \quad (2)$$

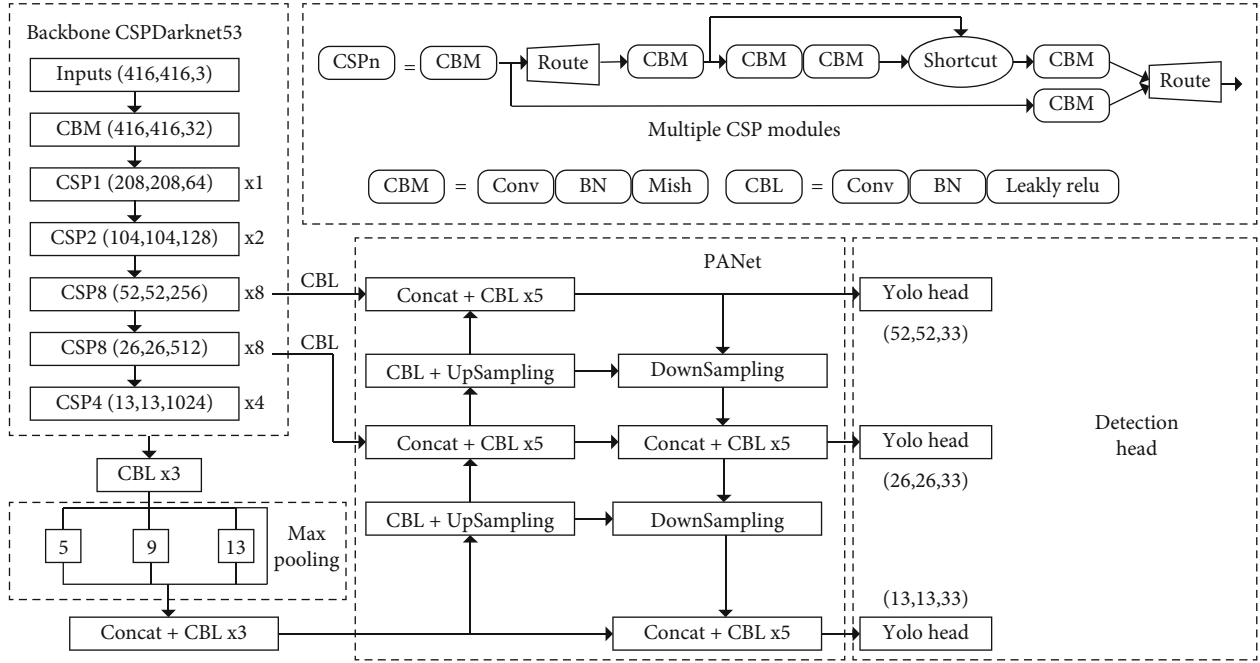


FIGURE 1: YOLOv4 algorithm structure.

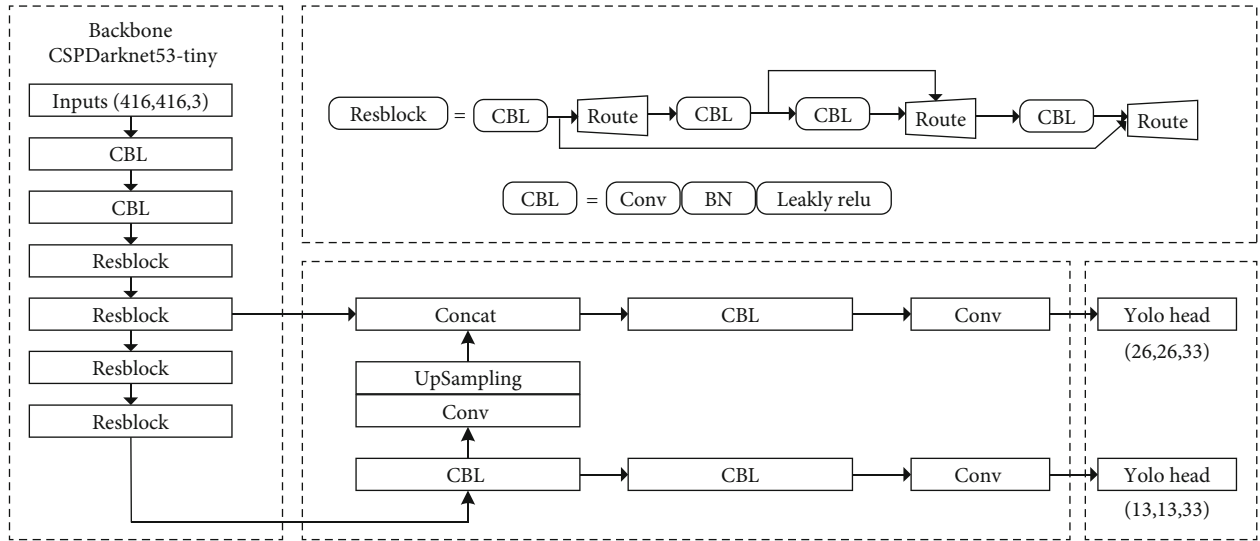


FIGURE 2: YOLOv4-Tiny algorithm structure.

The function supposes that all classes i are exclusive, and it normalizes the prediction probability z_i of each class and makes the sum of them 1.

2.2. Detection Box Modification and Duplication Strategy. In the prediction stage, the original YOLOv4 algorithm proceeds Nonmaximum Suppression (NMS) for one class each time because the combined detection method in this paper marks only the worker’s upper part of the body; the similarity of classes is high. If a prediction box belongs to Class A and Class B at the same time, it is redundancy. So, the regular NMS algorithm is used for the prediction box of a certain class and afterwards all classes to eliminate the duplication

of the same worker detected as multiple classes as shown in Figure 4.

3. Model Lightweight Based on CLSlim-YOLOv4

3.1. BN Layer and Scaling Factor. BN [20] was a data normalization method proposed, and it has been applied in most CNNs. Traditional standardization methods distribute the input of CNN between $[0, 1]$, but most of the activation functions in CNN, such as sigmoid and tanh, are linearly distributed in the interval $[0, 1]$, and standardization methods can reduce the nonlinear capability of the network.

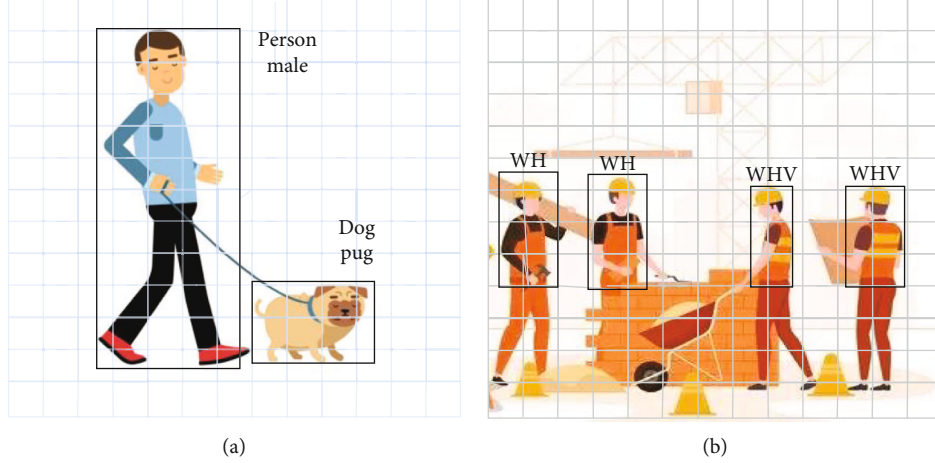


FIGURE 3: Classes in regular object detection vs. multiple PPE detection.



FIGURE 4: Two-stage method of NMS.

In order to reduce the effects of standardization on activation functions, two parameters under learning: scaling factor γ and shifting factor β are imported on the basis of standardization, and the values after standardization are zoomed and panned, which can regain the nonlinear expression ability of the convolution network to a certain extent. The flow of the BN algorithm is followed, in which $X_{\text{minibatch}}$ and y_i are the input and output of the BN layer; μ_X and σ_X^2 are the mean and variance values of the input of the BN layer; \hat{x}_i is the result after standardization.

In the YOLOv4 network, most convolution structures are composed of the convolution layer, BN layer, and activation function as shown in the CBM and CBL modules in Figure 1. If the scaling factor of the BN layer is very small, the value input into the activation function is very small, which represents the contribution of the corresponding channel to the network is also very low. Therefore, γ of the BN layer can be used as the scaling factor of channel pruning to evaluate the importance of the channel to the network without additional costs.

3.2. Sparse Training. In the process of network sparse training, γ in the CBM and CBL modules of the BN layer is con-

Input: $X_{\text{minibatch}} = \{x_1, x_2, \dots, x_m\}$;	
Parameter: γ, β ;	
Output: $\{y_i = BN_{\gamma, \beta}(x_i)\}$;	
$\mu_X \leftarrow 1/m \sum_{i=1}^m x_i$	//Mini-batch min
$\sigma_X^2 \leftarrow 1/m \sum_{i=1}^m (x_i - \mu_X)^2$	//Mini-batch variance
$\hat{x}_i \leftarrow (x_i - \mu_X) / \sqrt{\mu_X^2 + \epsilon}$	//Standardization
$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv BN_{\gamma, \beta}(x_i)$	//Scale and shift

ALGORITHM 1

sidered the scaling factor of channel pruning and multiplied by the corresponding channel. Then, a sparse model is contained after combined training on network weights and scaling factors. Most of the scaling factors of channels tend to 0, and the corresponding loss function is shown in

$$\text{Loss} = \sum_{(x,y)} l(f(x, W), y) + \lambda \sum_{\gamma \in \tau} g(\gamma), \quad (3)$$

where (x, y) are the training input sample and the corresponding label; Loss is the loss function of CNN normal

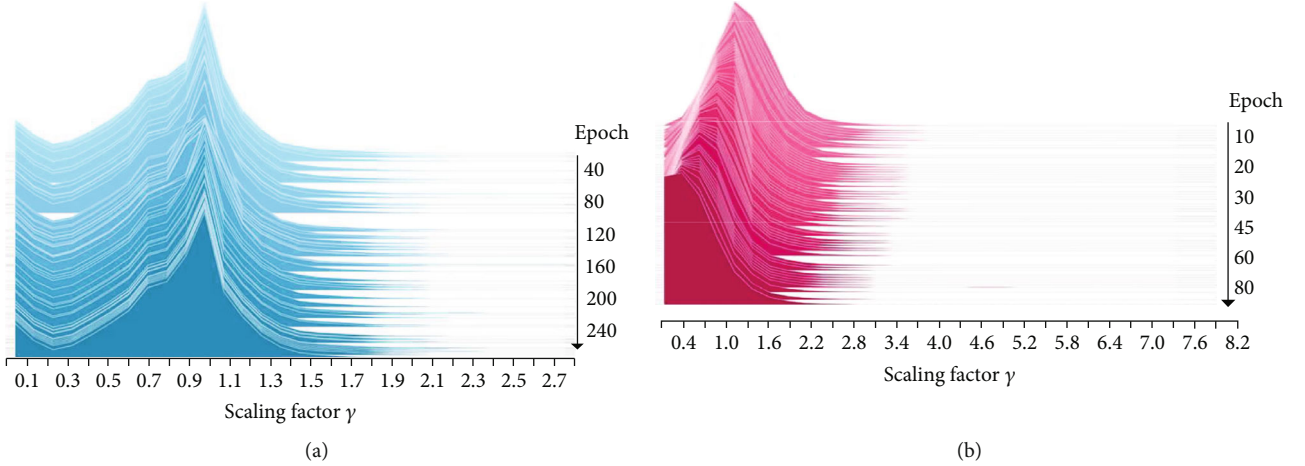


FIGURE 5: (a) Distribution of scaling factor without sparse process; (b) distribution of scaling factor with sparse process.

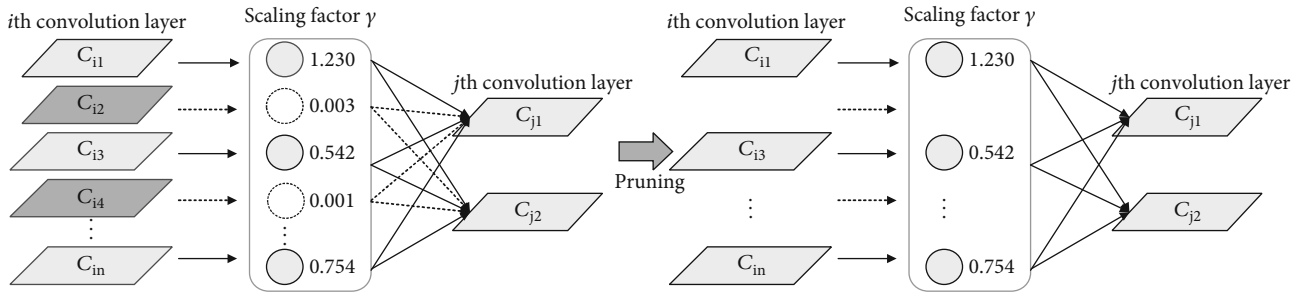


FIGURE 6: Model sparse training and channel pruning.

training; W is the weight of the network to be trained; $g(\gamma)$ is the penalty function on scaling factor, adopting $g = |s|$, i. e., L1 regularization; and λ is the penalty coefficient balancing two weights.

The essence of model channel pruning is cutting out the connection between the input and output related to a channel. Due to the combined optimization of loss function under normal training and scaling factor γ , a valuable channel can be chosen based on the scaling factor during the sparse training process. In the training process, the scaling factor γ of the BN layer shows approximately a normal distribution that expectation is 1 in the nonsparse YOLOv4 network as shown in Figure 5(a). When the penalty coefficient in equation (3) is set as 0.0005, after sparse training on the model, most of the scaling factor γ all go close to 0 as shown in Figure 5(b).

3.3. Channel and Layer Pruning. After sparse training, the scaling factor γ introduced in Section 3.1 is taken as the basis evaluating the importance of the channel. This paper defines a global threshold to control the pruning ratio and introduces a local safety threshold to prevent overpruning on the number of convolution layer channels and maintain the integrity of network connectivity. Figure 6 shows the model sparse training and channel pruning: each channel of the k th convolution layer is given a scaling factor; after sparse training, the scaling factor approaches 0; the absolute value of the scaling factor smaller than the global threshold

is removed; if the scaling factors of the channels in the whole layer are small than the global threshold, the channels with scaling factors bigger than the local safety threshold are reserved to prevent the whole layer pruned.

In the YOLOv4 network pruning process, some structures need to be handled, such as the CSPn module in the backbone network CSPDarknet53; and the max pool and unsample layers independent of the number of channels can be ignored directly. In the channel pruning process, the pruning ratio is settled firstly. And then, $|\gamma|$ of the BN layer to be pruned is ascending sort. And the global threshold $\tilde{\gamma}$ and local safety threshold π are determined based on the pruning ratio and channel ratio to be reserved for each layer. Channels to be deleted in each layer are set to 0, and others 1. Then, the pruning mask is obtained. The CFG structure of CSP1 in the Darknet framework is shown in Figure 7. As to the route layer, the characteristic chart of the corresponding index is output when there is only one parameter; concatenate operation is proceeded when there are two parameters. Therefore, the pruning masks of the corresponding input layers are connected to and used as their own pruning masks. The structure of the shortcut layer is similar to the residual module of ResNet. Therefore, all layers for shortcut connection need the same number of channels. The final pruning masks are generated after traversing the pruning masks of these layers and making logic or operation.

Channel pruning can greatly reduce the model and parameter calculation volumes but has little impact on

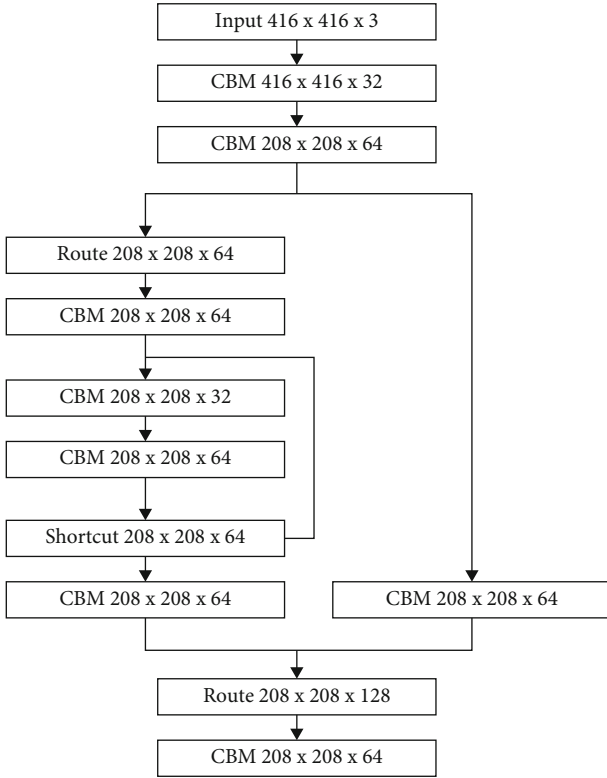


FIGURE 7: CFG structure of the CSP1 module.

improving the inference speed of the model. This paper proposes a layer pruning method on the basis of channel pruning, which works by sorting the scaling factor's mean value γ_{mean} for each convolution layer, pruning the layer with the smallest γ_{mean} , and introducing layer pruning coefficient S which is the number of shortcut structures to be pruned. There are 23 shortcut structures in the backbone network of YOLOv4. For the integrity of the network structure, each shortcut with the corresponding two convolution layers in its upper layer is pruned at the same time. If the layer pruning coefficient S is 8, there will be 24 layers pruned. Layer pruning can improve the inference speed of the model. Channel pruning and layer pruning are used to compress the width and depth of the model, respectively. The pruned model has a great improvement in parameter calculation, memory ratio, and inference speed.

3.4. Multiple Iterative Pruning and Fine-Tuning Model. Due to the reduction of the number of model channels and layers, there will be accuracy loss inevitably. Therefore, using the original data set to fine-tune the pruned model to regain accuracy of the model is necessary. In order to protect the model's complete network structure and high detection performance after pruning, the threshold with less precision loss can be used for pruning each time, and pruning and fine-tuning can be proceeded many times until the best pruning performance is achieved. The multiple pruning process is shown in Figure 8.

The CLSlim method proposed in the paper combines channel pruning and layer pruning to compress the model. After the CLSlim method used in YOLOv4, the number of

the channel and layer can be reduced largely and the structure of the original model can be kept meanwhile, which will provide the possibility for application on embedded devices. Setting different layer pruning coefficients will cut off different numbers of shortcut structures, so the structure diagram of the pruning model is not fixed. When the layer pruning coefficient is 20, 20 shortcuts of backbone network CSPDarknet53 in the YOLOv4 will be pruned. Then, the pruning model network structure can be obtained as Figure 9, where Slim-CSPn is the pruned CSPn structure. Compared with the model before pruning, the CSPDarknet53 backbone network has been reduced by 60 layers.

The YOLOv4-Tiny network can also be lightweight improved using the CLSlim method. Since YOLOv4-Tiny's backbone CSPDarknet-Tiny network does not contain the shortcut structure, it only needs to make channel pruning in YOLOv4-Tiny to achieve the model compression effect.

4. Experiment Analysis

4.1. Experiment Environment. The experiment environment in this paper is under the Windows 10 professional operating system. Pytorch and Darknet Deep Learning Frameworks are applied for the realization of the combined detection algorithm on PPE. The configuration of the server is the graphics card of NVIDIA GTX2080ti and processor of Intel Core i7; RK3399pro is adopted as the performance verification embedded platform.

4.2. Experiment Data Set. Construction scenes generally separate into aerial work and ground work. As the most common and basic PPE for workers, a safety hard hat needs to be worn regularly at any time entering the construction site. Ground workers are mostly exposed to and in the shade of the sun. Due to the heavy dust on the construction site, it is difficult to accurately identify the positions of workers in this kind of bad environment. Therefore, ground workers should wear reflective clothing throughout the whole working period to avoid collision accidents; aerial workers should wear safety belts to avoid falling accidents. It takes the two kinds of works as an example, and the construction data set for PPE combined detection is shown in Table 1.

Experiment data mainly comes from the surveillance video of a building construction site in Xi'an. The camera takes pictures of workers standing, squatting, walking, and in other positions from multiple angles. But there is not enough data against rules. With these unbalanced sample categories, it might cause inaccurate experimental results. Therefore, to expand the data set, pictures of some certain categories against rules are taken independently at the construction site. The labeling boxes and sample pictures of each category are listed in Table 2. Label image is used to label the pictures, and data set in VOC format is set up, and samples are divided into the training set and test set by 8:2 for model training and performance verification, respectively.

4.3. Model Training and Pruning. The multiple iterative pruning and fine-tuning process in Section 3.4 is applied

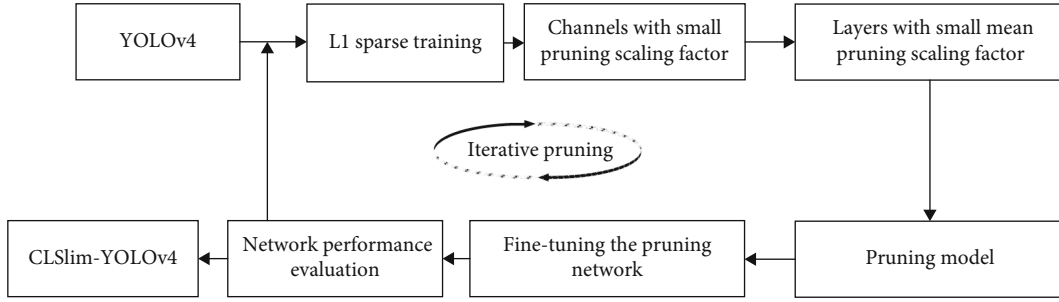


FIGURE 8: Multiple pruning process.

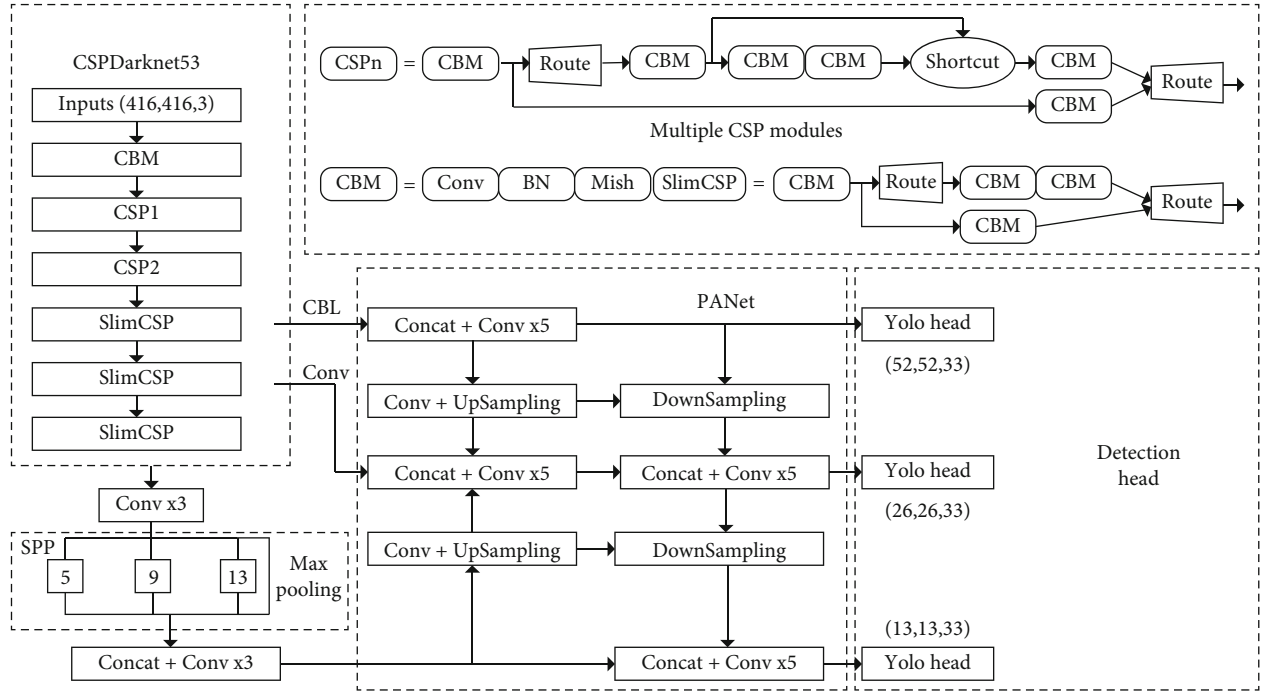

 FIGURE 9: CLSlim-YOLOv4 structure ($S = 20$).

TABLE 1: The data set of combined detection on personal protective equipment.




Scene	Unsafe behavior	Label	Description
Ground work	Without safety hard hat Without reflective clothing	W	Worker
		WH	Worker with safety hard hat
		WV	Worker with reflective clothing
		WHV	Worker with safety hard hat and reflective clothing
Aerial work	Without safety hard hat Without safety belt	W	Worker
		WH	Worker with safety hard hat
		WB	Worker with safety rope
		WHB	Worker with safety hard hat and safety rope

for channel pruning and layer pruning on the YOLOv4 network, and then, the CLSlim-YOLOv4 model is obtained. The performance of the model is verified by using the self-built PPE real-time detection data set. The model training needs to be trained on computers with high configuration hardware and high-performance GPU graphics cards, and the experimental environment should be set up, and various

dependency libraries should be installed on the server. For the comparison of the performances of the pruning model and nonpruning model, the self-built PPE real-time detection data set is imported to train the YOLOv4 model firstly; then, the trained model is used to prune. The model pruning experiment is also based on the self-built data set. And the pruning process is as follows:

TABLE 2: Numbers of categories and sample pictures.

(a)

Label	W	WH	WV
Number of labeling boxes	1756	5327	2227
Sample picture			

(b)



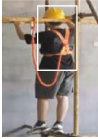
Label	WHV	WB	WHB
Number of labeling boxes	7284	3797	5442
Sample picture			

TABLE 3: Experimental parameters.

Parameter	Value
Learning rate (learning rate)	0.002324
Number of iterations (epoch)	400
Batch size (batch_size)	8
Momentum (momentum)	0.97
Weight decay (weight_decay)	0.0004569
Learning rate decay factor (lr_factor)	0.1
Penalty coefficient (λ)	0.0005

(1) *Sparse Training*. It is a game process between accuracy and sparsity of the model referring to the loss function during the model sparse training process in Section 3.2. If the penalty coefficient of the scaling factor is too high, the network is with fast sparsity but the accuracy drops fast too; else, the model's accuracy loss is low but with very slow sparsity. Therefore, choosing a suitable penalty coefficient is of great importance. Taking the experiment cases in previous studies, about 100 rounds of sparse training can reach maximum performance. The parameters applied in the sparse training experiments in this paper are shown in Table 3. A total of 400 rounds of training is proceeded to ensure sufficient time left after sparse training for further adjusting the model.

The sparse training process is shown in Figure 10(a). Large compression of the model is completed in about the first 100 rounds, and the accuracy is fine-tuned and restored in the following 300 rounds. The loss and mAP curves of the model are shown in Figures 10(b) and 10(c), in which while the scaling factor of the BN layer is in the period of substantial compression (20-100 epochs), the loss value of the model

increases continuously and then goes to the adjustment period. The loss value decreases rapidly in the 280 rounds, and the mAP of the model rebounds significantly; the learning rate is reduced in the last 120 rounds, and the model accuracy is repaired.

The performances of the model before and after sparse training are compared in Table 4. The model accuracy after sparse training is 2% less than the original YOLOv4 model.

(2) *Channel Pruning and Layer Pruning*. The pruning ratios of the experiments in this paper are set to 0.8, 0.9, and 0.95; the ratio of channels to be reserved on each layer is 0.01, corresponding to the global threshold $\tilde{\gamma}$ and local safety threshold π in Section 3.3. To get the best pruning parameters, three groups of comparative experiments are designed for channel pruning of the model: YOLOv4-0.8, YOLOv4-0.9, and YOLOv4-0.9, and the performance of the pruning model is evaluated based on five kinds of indices: model size (model_size), mean accuracy (mAP), floating point of operations (FLOPs), parameters (params), and inference speed (inference). The model detection performances with different pruning ratios are shown in Table 5, in which, with low precision loss, the YOLOv4-0.9 model compresses its volume smaller and reduces the volume of floating point operations. So, the YOLOv4-0.9 model is chosen as the benchmark model for the layer pruning experiment.

Channel pruning can reduce the volume of model parameter calculation and improves nothing on inference speed. Therefore, layer pruning on shortcut structures of the backbone network is required after channel pruning. In this paper, layer pruning on the best model YOLOv4-0.9 chosen from channel pruning experiments is proceeded.

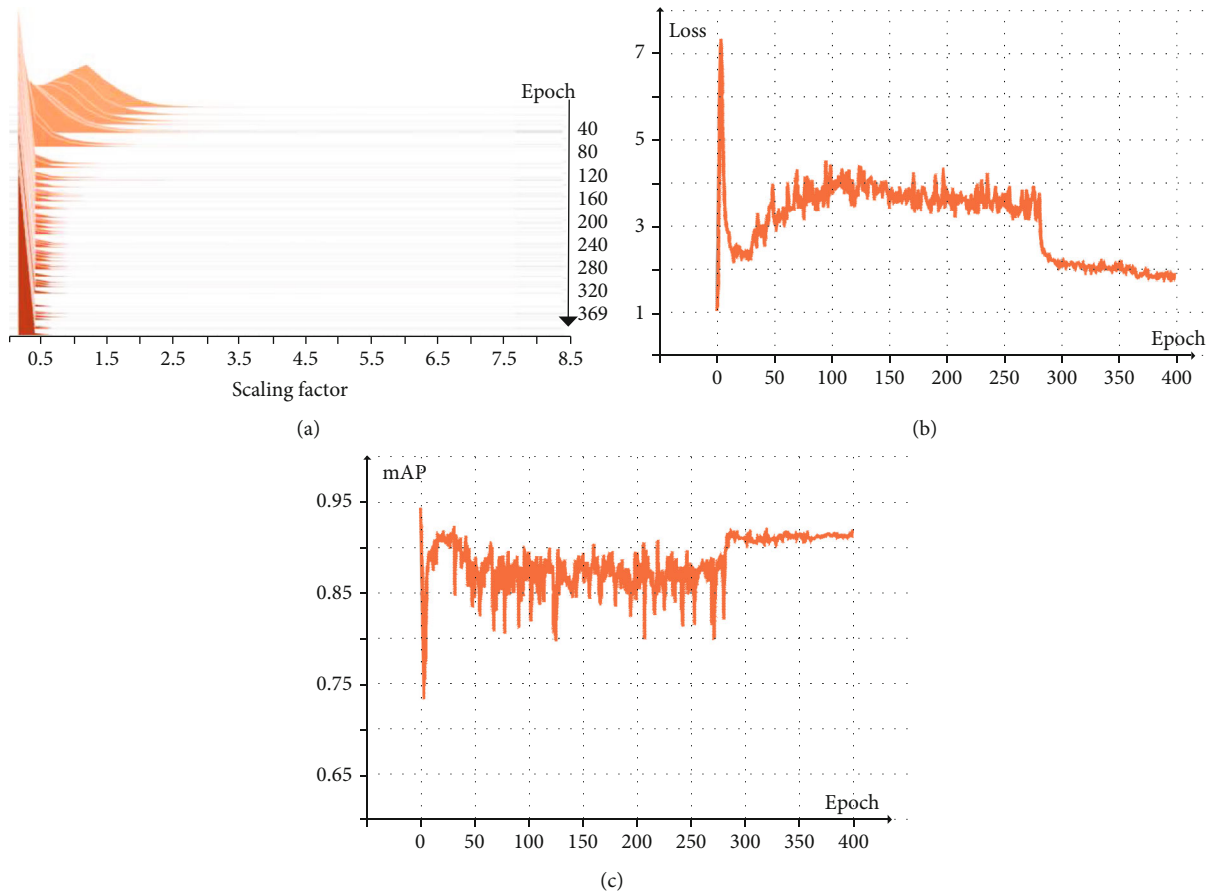


FIGURE 10: (a) Distribution of scaling factor in sparse training process; (b) loss curve in sparse training process; (c) mAP curve in sparse training process.

TABLE 4: Comparison of model performances before and after sparse training.

Model	Model_size (MB)	mAP (%)
YOLOv4	244 MB	93.4%
YOLOv4 after sparse training	244 MB	91.7%

TABLE 5: Comparison of the channel pruning experiments.

Experiment	Model_size (MB)	mAP (%)	FLOPs (G)	Params (M)	Inference (ms)
YOLOv4	235	94.9	59.80	63.96	33.9
YOLOv4-0.8	14.00	91.64	18.12	3.67	33.6
YOLOv4-0.9	4.46	91.40	7.64	1.16	33.8
YOLOv4-0.95	1.73	1.41	3.48	0.45	33.5

Layer pruning coefficient S is set as 8, 16, 20, 22, and 24, separately, and the five kinds of indices for model performance evaluation in channel pruning experiments are also applied in channel pruning. The model performances with different layer pruning ratios are shown in Table 6.

After the layer pruning on the YOLOv4-0.9 model, the model accuracy always decreases to a low level, so the origi-

nal data set needs to be used and fine-tuned to rebound the lost accuracy. YOLOv4-0.9-20 is finally chosen for fine-tuning the training. The mean detection accuracy rebounds from 84.2% to 92.8%. And after the fine-tuning, the final pruning model CLSlim-YOLOv4 is obtained.

4.4. Evaluation of Pruning Model. To verify the effectiveness of the pruning method, YOLOv4 and YOLOv4-Tiny are both improved with lightweight CLSlim in this paper. The corresponding results are shown in Table 7.

Table 7 compares the model size, mean accuracy, and other indices of YOLOv4, YOLOv4-Tiny, and pruning model CLSlim-YOLOv4 and CLSlim-YOLOv4-Tiny. The size of CLSlim-YOLOv4 is compressed to 4.15 M, 1.76% of YOLOv4’s, with inference speed increasing 1.8 times and 1.9 times in GTX2080ti and RK3399pro, respectively, FLOPs decreasing 12.1% and model accuracy decreasing 2.1%. Applying the model pruning method in YOLOv4-Tiny, the model size is compressed to 25.6%, parameter volume compressed to 25.5%, FLOPs decreasing 57%, inference speed increasing 1.1 times of the two type devices, and model accuracy increasing 0.8%. Parts of the CLSlim-YOLOv4-Tiny detection results in RK3399pro are shown in Figure 11.

TABLE 6: Comparison of the layer pruning experiments.

Experiment	Model_size (MB)	mAP (%)	FLOPs (G)	Params (M)	Inference (ms)
YOLOv4-0.9	4.46	91.40	7.64	1.16	33.8
YOLOv4-0.9-8	4.44	91.4	7.64	1.16	29.0
YOLOv4-0.9-16	4.34	89.4	7.52	1.13	23.0
YOLOv4-0.9-20	4.15	84.2	7.26	1.08	20.4
YOLOv4-0.9-22	4.06	65.1	6.78	1.05	19.5
YOLOv4-0.9-24	4.04	32.2	6.20	1.05	18.7

TABLE 7: Comparison of the model pruning experiments.

Model	Model_size (MB)	mAP (%)	FLOPs (G)	Params (M)	Inference (ms)	
					GTX2080ti	RK3399pro
YOLOv4	235	94.9	59.80	63.96	33.9	620
YOLOv4-Tiny	23.1	93.8	6.92	6.07	7.1	39
CLSlim-YOLOv4	4.15	92.8	7.26	1.08	18.7	320
CLSlim-YOLOv4-Tiny	5.92	94.6	4.00	1.55	6.5	33



FIGURE 11: Recognition results of CLSlim-YOLOv4-Tiny in RK3399pro.

5. Conclusions

A combined detection algorithm on personal protective equipment for mobile terminals is proposed to check whether it is worn properly. The algorithm is realized based on the YOLOv4 network. Firstly, this algorithm applies L1 regularization and gradient sparse training on the scaling factor of the BN layer in the convolutional module of YOLOv4, and a global pruning threshold is settled to eliminate channel redundant parameters; at the same time, layer pruning thresholds are set to maintain the network structure integrity. After channel pruning, model size and parameter calculation volume decrease significantly. Then, the mean values of the scaling factors of each layer of the backbone

network are sorted. Combining the layer pruning coefficient, several layers with small mean values of scaling factors are pruned, and the inference speed is improved. Afterwards, the pruning model CLSlim-YOLOv4 is obtained after 2-3 rounds of fine-tuning. To verify the effectiveness of the pruning method in this paper, with the same data set and test environment, the lightweight CLSlim method is imported into YOLOv4 and YOLOv4-Tiny. The test results show that with the premise of greatly reducing the parameter calculation volume and improving the inference speed, the accuracy losses of CLSlim-YOLOv4 are only 1%-2%; compared to YOLOv4-Tiny, CLSlim-YOLOv4-Tiny performs better in detection accuracy, parameter calculation, and inference speed. There might be false and missed

detection in the real-life test of this research. Data sets can be expanded and enriched to improve the fitting ability of the model in afterwards research.

The combined detection algorithm on PPE in this paper can detect several kinds of PPEs at the same time, and ensuring the strong feature extraction ability of complex models, the model lightweight improvement is made to maintain high accuracy even with substantial parameter compression. This method satisfies the real-time ability and accuracy requirements of the combined detection of PPE in the real construction environment. The follow-up research will continue to combine the ones with other model lightweight strategies to improve the model inference speed and find model lightweight methods more suitable for source-limited mobile terminals.

Data Availability

The (PPE combined detection) data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (52104215) and the Key Industry Innovation Chain Project of Shaanxi Key Research and Development Plan (2021ZDLGY07-08).

References

- [1] Osha, "Personal protective equipment," 2022, <https://www.osha.gov/personal-protective-equipment>.
- [2] Osha, "Fall protection," 2022, <https://www.osha.gov/fall-protection>.
- [3] Osha, "Occupational safety and health administration," 2022, <https://www.osha.gov/laws-regs/standardinterpretations/2002-03-11>.
- [4] Y. Han, J. J. Zhang, H. Sun, J. Y. Yao, and S. D. You, "Design and implementation of intelligent safety inspection system for construction workers based on image recognition," *Journal of Safety Science and Technology*, vol. 12, no. 10, pp. 142–148, 2016.
- [5] K. Shrestha, P. P. Shrestha, D. Bajracharya, and E. A. Yfantis, "Hard-hat detection for construction safety visualization," *Journal of Construction Engineering*, vol. 2015, 8 pages, 2015.
- [6] B. Balakrishnan, G. Richards, G. Nanda, H. Mao, R. Athinarayanan, and J. Zaccaria, "PPE compliance detection using artificial intelligence in learning factories," *Procedia Manufacturing*, vol. 45, pp. 277–282, 2020.
- [7] J. Wu, N. Cai, W. Chen, H. Wang, and G. Wang, "Automatic detection of hardhats worn by construction personnel: a deep learning approach and benchmark dataset," *Automation in Construction*, vol. 106, p. 102894, 2019.
- [8] Z. Xie, H. Liu, Z. Li, and Y. He, "A convolutional neural network based approach towards real-time hard hat detection," in *2018 IEEE International Conference on Progress in Informatics and Computing (PIC)*, pp. 430–434, Suzhou, China, Dec 2018.
- [9] N. D. Nath, A. H. Behzadan, and S. G. Paal, "Deep learning for site safety: real-time detection of personal protective equipment," *Automation in Construction*, vol. 112, p. 103085, 2020.
- [10] G. Han, M. Zhu, X. Zhao, and H. Gao, "Method based on the cross-layer attention mechanism and multiscale perception for safety helmet-wearing detection," *Computers & Electrical Engineering*, vol. 95, article 107458, 2021.
- [11] H. Wang, F. Lu, X. Tong, X. Gao, L. Wang, and Z. Liao, "A model for detecting safety hazards in key electrical sites based on hybrid attention mechanisms and lightweight Mobilenet," *Energy Reports*, vol. 7, pp. 716–724, 2021.
- [12] A. Bochkovskiy, C. Y. Wang, and H. Liao, "YOLOv4: optimal speed and accuracy of object detection," 2020, <http://arxiv.org/abs/2004.10934>.
- [13] P. Zhang, Y. Zhong, and X. Li, "Slim YOLOv3: narrower, faster and better for real-time UAV applications," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, Seoul, Korea (South), Oct. 2019.
- [14] F. U. Huitong, W. A. N. G. Peng, L. I. Xiaoyan, L. U. Zhigang, and D. I. Ruohai, "Lightweight network model for moving object recognition," *Journal Of Xi'an Jiaotong University*, vol. 7, 2021.
- [15] J. Yin, G. Liang, W. Jiang, S. Hong, and J. Yang, "ShuffleNet-inspired lightweight neural network design for automatic modulation classification methods in ubiquitous IoT cyber-physical systems," *Computer Communications*, vol. 176, pp. 249–257, 2021.
- [16] A. P. Fard and M. H. Mahoor, "Facial landmark points detection using knowledge distillation-based neural networks," *Computer Vision and Image Understanding*, vol. 215, article 103316, pp. 1077–3142, 2022.
- [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 *2020 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW)*, Seattle, WA, USA, June 2020.
- [18] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, Honolulu, HI, USA, July 2017.
- [19] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June 2018.
- [20] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, Lille, France, July 2015.