An Efficient Face Mask Wearing Detection Algorithm Based on Improved YOLOv3

Bo Zhang, Xiaoxia Zhang, Zhuo Li

Abstract—**In recent years, the Corona Virus Disease 2019 (Covid-19) epidemic has raged around the world, with more than 500 million people diagnosed. Relevant medical research and analysis results on Covid-19 indicate that wearing masks is an effective method to prevent and restrain virus transmission. Mask detection stations have been set up in hospitals, railway stations, schools, where there is large crowd flow, but results are not as good as expected. In order to ameliorate pandemic preventing and control measures, a mask wearing detection algorithm YOLOv3-M3 was designed and proposed in this paper. The algorithm can effectively detect people without mask, while consequently reminding them. Firstly, we substituted the feature extraction network of YOLOv3 with MobileNetv3, a lightweight convolutional neural network. Secondly, we utilized K-Means++ to substitute the original ground truth clustering algorithm to improve prediction precision. In addition, the bounding box regression loss function was revised as *CIoU* loss function. This loss function solves the issues of overlapping between the ground truth and the anchor box, which has increased the training speed. After experiments, the precision of YOLOv3 algorithm on *mAP 0.5* and *mAP 0.75* is 93.5% and 71.9%, respectively. Elevating 3.1% and 2.6%, respectively, higher than that of YOLOv3 algorithm, and it was superior to SSD, SSD Lite, YOLOv3-Tiny and other one-stage object detection algorithms. The detection speed can reach 13.6 frame/s, which has met the requirements of pandemic prevention and control in most places and can be deployed on terminal devices for object detection.**

Index Terms—**Covid-19, Face mask wearing detection, YOLOv3, MobileNetv3, K-Means++, *CIoU* loss function**

I. Introduction

SINCE 2020, the Covid-19 continuously spreading has even mutated to coronavirus variants around the world. A variety of coronavirus variants have been classified by the World Health Organization (WHO) Listed as focus object. Variants have higher transmissible features than the original virus, with higher possibility to cause diseases, in addition to severely reducing the effectiveness of medicine treatment [1]. It poses a serious threat to people's lives and property. Covid-19 is mainly transmitted by droplet contact, indirect and airborne transmissions. Therefore, all countries' governments have appealed for wearing masks during the epidemic period, which effectively insulated the transmission of the virus. At present, mask detection stations are set up in the public around many countries, manual detection is used but this is not a perfect method because it will not only cause miss detection, but also there is a risk of infection [2]. It fails to reach perfect pandemic prevention. Therefore, it is very significant to install machines in these places for automatic detection.

Face mask wearing detection is a practical application based on object detection in computer vision. Computer vision can complete object detection, target tracking, instance segmentation, semantic segmentation and others. As an important branch of computer vision, object detection performs excellently in medical detection, industrial and agricultural product detection, auto driving and other practical problems [3]. Currently, object detection algorithms are fall into two categories. Representative algorithms of two-stage detection include RCNN and Faster-RCNN, the precision of these kinds of algorithms is comparatively higher but time-consuming. Apart from that, YOLO and SSD belong to one-stage detection which is fast in detection speed but the detection results for long-range and small objects are not ideal.

At present, in order to promote the detection efficiency and detection performance of YOLO series algorithms, researchers have enhanced and optimized the series algorithms. In addition, some literature has studied face mask wearing detection based on these improvements. Wan et al. [4] proposed an efficient traffic sign detection model. Based on YOLOv3, the network was appropriately prune to reduce the model volume, and a prediction layer was added to make the model predict at four scales. Experiments on Tsinghua -Tencent 100 K traffic sign dataset, the accuracy was 93.8% and the detection speed was 2.7 times higher. Deng et al. [5] proposed vehicle and lane line detection based on YOLOv4, they used MobileNetv3 as a feature extraction network and used atrous spatial pyramid pooling and feature pyramid networks to improve the feature extraction capability of MobileNetv3. Finally, the convolutional block attention module mechanism was introduced into YOLOv4. Trained on BDD100K dataset and tested on KITTI dataset, the model average precision was boosted by 1.1% over YOLOv4, 1.7 times faster than the original model. Shilpa et al. [6] designed a high-speed and accurate mask detection model, which was a combination of one-stage and two-stage algorithms. Resnet-50 was used as the backbone network, and the transfer learning method was applied to improve the stability of the model. The experimental accuracy was 98.2%. Yu et al. [7] proposed an efficient mask detection algorithm based on YOLOv4 detection model, which improved the backbone network

Manuscript received April 28, 2022; revised October 2, 2022.

B. Zhang is a Postgraduate Student of School of Computer Science and Software Engineering, University of Science and Technology LiaoNing. AnShan 114051, China (e-mail: 417122189@qq.com).

X. X. Zhang, the corresponding author, is a Professor of School of Computer Science and Software Engineering, University of Science and Technology LiaoNing, Anshan,11041, China (corresponding author, phone:86-0412-5929812; e-mail: aszhangxx@163.com).

Z. Li is a Postgraduate Student of School of Computer Science and Software Engineering, University of Science and Technology LiaoNing. AnShan 114051, China (e-mail: 2547244504@qq.com).

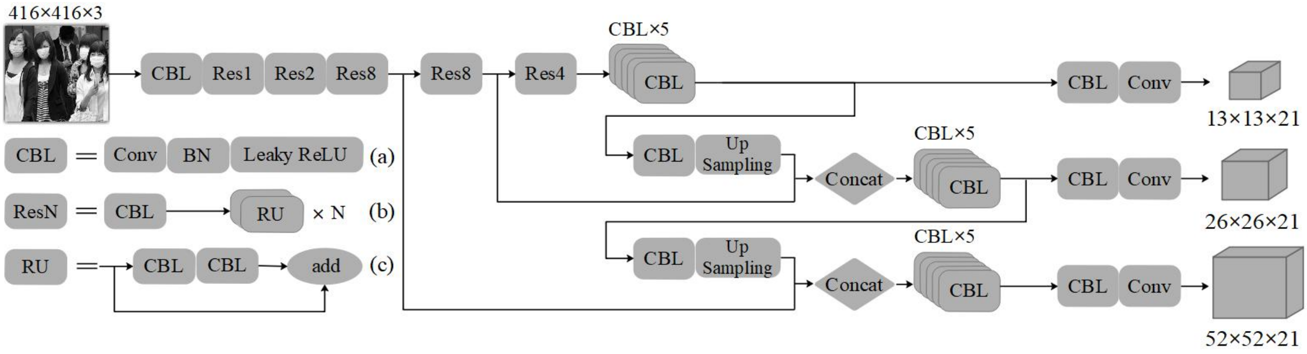


Fig. 1. The structure of YOLOv3 algorithm.

CSPDarknet-53 and decomposed the backbone network into CSP1_X and CSP2_X. The network parameters were reduced to a certain extent and the information fusion was improved. The mean average precision value of the final experiment was 98.3%. Francesco et al. [8] designed a transfer learning method, using MobileNetv2 model as the main trunk network to identify people without wearing mask in images or videos and remind them. The experiment was trained on 4095 face images, and the accuracy on the test set was 98%.

Although methods mentioned above have considerably boosted accuracy, the complex network structure has led to low detection speed and long training time, thus we propose a lightweight object detection algorithm YOLOv3-M3. First of all, since the general convolutional neural network contains enormous parameters and computation, we used lightweight convolutional neural network MobileNetv3 as the feature extraction network of YOLOv3. Moreover, K-Means++ clustering algorithm was applied to conduct clustering analysis on the coordinates, length and width of the ground truth to screen out the optimal anchor box. This clustering algorithm can not only greatly reduce the training loss value but also improve the precision of object predicted position. Finally, considering the regression loss, we replaced *IoU* with *CIoU* to alleviate low convergence speed caused by slow average loss decrease when the ground truth was wrapped by the anchor box.

The structure of other sections in this paper is as follows: Section 2 mainly introduces the principle and related techniques of YOLOv3 algorithm. We ameliorate the YOLOv3 algorithm and name it YOLOv3-M3. The improved method and detection process of the algorithm are illustrated in Section 3. Section 4 reports experiment with the improved algorithm and evaluation of the results. In Section 5, draw conclusions based on the results of the experiments and propose future improvements.

II. Materials And Method

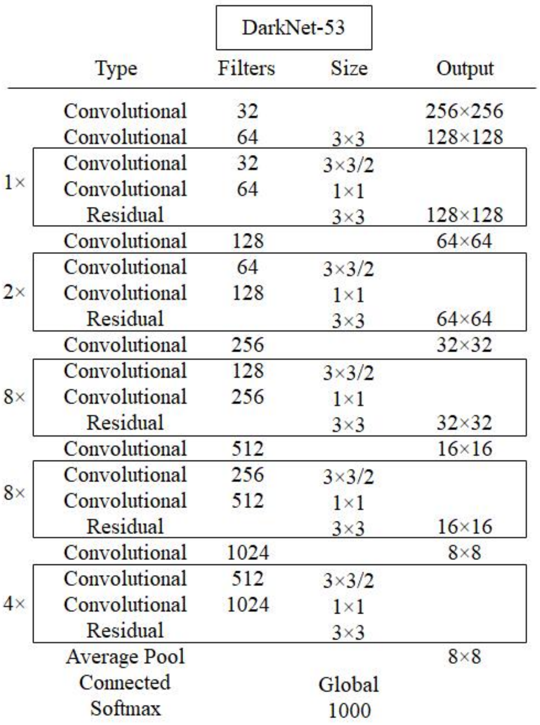
A. YOLOv3 Object Detection Algorithm

YOLOv3 is an object detection algorithm proposed by Joseph Redmon in 2018, integrated with the advantages of YOLOv1, YOLOv2, etc. In particular, three parts of network structure, network characteristics and loss function are improved. The enhanced strategy not merely improves the precision of small targets but keeps the original speed. The algorithm is mainly composed of four parts. Firstly, in the input layer, image sizes are multiples of 32×32 and has three channels. Secondly, feature extraction is carried out on the images through the backbone network. Thirdly, Feature Pyramid Network (FPN) is applied to collect and integrate the features of diverse scales in neck network. Finally, the object detection results of each scale are output. The structure of YOLOv3 algorithm is shown in Fig.1.

B. Backbone of YOLOv3—Darknet-53

The object detection algorithm uses backbone to process image information. From YOLOv1 to YOLOv3, the performance of each version is connected with the backbone network. YOLOv3 is derived from YOLOv2, and the backbone has been updated from DarkNet-19 to DarkNet-53. DarkNet-53 network is composed of 52 convolutional layers and one output layer, and without full connection layer, so with another appellation as full convolutional network. The structure of DarkNet-53 backbone is represented in Fig.2. The network depth of DarkNet-53 is higher than DarkNet-19, which improves the detection precision of different sizes objects.

The convolution layer of the backbone consists of several convolution kernels of 3×3 and 1×1. Besides, add a *BN* layer



DarkNet-53

	Type	Filters	Size	Output
	Convolutional	32		256×256
	Convolutional	64	3×3	128×128
1×	Convolutional	32	3×3/2	
	Convolutional	64	1×1	
	Residual		3×3	128×128
	Convolutional	128		64×64
2×	Convolutional	64	3×3/2	
	Convolutional	128	1×1	
	Residual		3×3	64×64
	Convolutional	256		32×32
8×	Convolutional	128	3×3/2	
	Convolutional	256	1×1	
	Residual		3×3	32×32
	Convolutional	512		16×16
8×	Convolutional	256	3×3/2	
	Convolutional	512	1×1	
	Residual		3×3	16×16
	Convolutional	1024		8×8
4×	Convolutional	512	3×3/2	
	Convolutional	1024	1×1	
	Residual		3×3	
	Average Pool			8×8
	Connected		Global	
	Softmax		1000	

Fig. 2. The structure of DarkNet-53 backbone.

and a *Leaky ReLU* layer after each convolution layer. These basic units are combined to form a convolution block *CBL*. The *CBL* block is shown in Fig.1. In addition, DarkNet-53 integrates residual network. YOLOv3 has a total of 5 *ResN* (Residual Block), and each *ResN* block consists of a *CBL* block and N *RU* blocks. Meanwhile, each *RU* blocks consist of two *CBL* blocks and a residual connection.

C. Neck Network—FPN (Feature Pyramid Network)

The previous object detection algorithms use single-scale features for prediction. However, in the convolutional neural network, semantic information is less in the low-level feature map, but object details are more abundant. On the contrary, the high-level feature map contains rich semantic, whereas the object location is sketchy. This is also why using advanced feature maps shows better detection results when dealing with large objects and worse detection results when dealing with small objects. After the image is down sampled, some small objects will disappear, resulting in object detection failure. But FPN emerged, which is a compatible and excellent solution to this difficult problem [9].

FPN fuses high-layer rich semantic features with low-layer rich detailed information by introducing top-down paths and lateral connections, so the network performance is improved. The neck network of YOLOv3 uses the idea of FPN to fuse the features of the three scales and integrate semantic information from the high-level feature map into the low-level feature map. Thus, the precision of small object detection is improved. In this paper, 13×13, 26×26 and 52×52 scales feature maps are used for prediction. The working process of FPN is illustrated in Fig. 3.

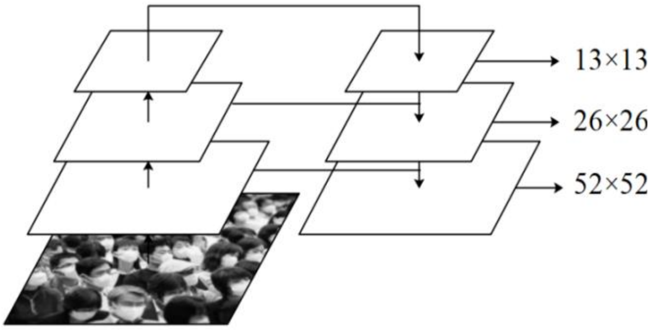


Fig. 3. The working process of FPN.

D. Multi-Scale Prediction

The method of YOLOv3 for multi-scale prediction is as follows: A feature map of size N×N is obtained by the feature extraction network, and then it is split into N×N grid cells. When the center point of the predicted object is on a grid cell, the grid cell will predict the object. Each grid cell has three anchor boxes with different scales. Ultimately, the anchor box that has the largest *IoU* with the ground truth is responsible for predicting this object. In all three prediction layers, the number of convolution kernels is 21, which is calculated by 3 × (2+4+1), where 3 is the number of anchor boxes, 2 is the two classes in this paper (face and face−mask), 4 is the center point coordinates and width and height of the bounding box, and 1 is the confidence. The dimensions of the three detection layers are 13×13×21, 26×26×21, and 52×52×21, respectively. These three different scales are shown in Fig. 4.

E. Prediction Process of Bounding Box and Anchor Box

The bounding box can complete prediction on the positions of the grid coordinates. As shown in Fig.5, the model predicts four values for each bounding box: t_x, t_y, t_w, t_h. Here, t_w and t_h are the predicted width-height offsets, t_x and t_y are the predicted coordinate offset. The output interval is normalized to the interval [0, 1] by the Sigmoid activation function so that its position within the current grid is controlled. Otherwise, the center of the bounding box can be located in any region of the image, which is not conducive to the convergence of the model.

In Fig.5, c_x and c_y denote the first grid coordinates, the black dashed box is the anchor box, and p_w and p_h are the horizontal length and vertical length of the dimensions of the anchor box mapped to the feature map. The solid box surrounded by the dashed box is the bounding box, and the predicted values are calculated as follows:

$$b_x = \sigma(t_x) + c_x \tag{1}$$

$$b_y = \sigma(t_y) + c_y \tag{2}$$

$$b_w = p_w e^{t_w} \tag{3}$$

$$b_h = p_h e^{t_h} \tag{4}$$

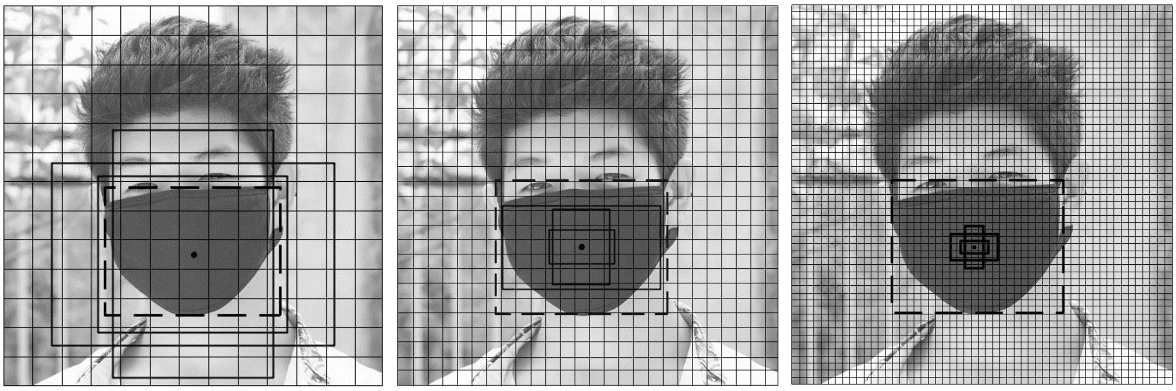


Fig. 4. Multi-scale prediction.

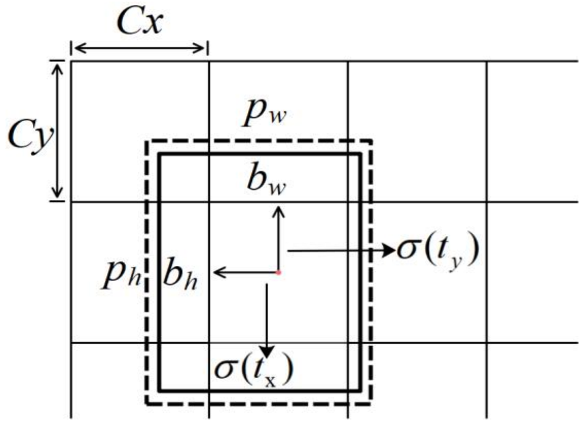


Fig. 5. Bounding box position of prediction.

In the calculation formula of the predicted value, σ is the Sigmoid activation function, b_x and b_y are the location information for the bounding box. b_w and b_h are the horizontal length and vertical length of the bounding box. After determining the prediction parameters, the loss of the bounding box and the ground truth is calculated by loss function. Then, reduce the loss value by gradient descent so that the predicted box slowly approaches the ground truth.

The anchor box mechanism of YOLOv3 continues the clustering algorithm in YOLOv2 to cluster the ground truth in the training set. It is worth mentioning that in the cluster algorithm of YOLOv3, the formula for calculating the distance between samples is based on *IoU*. The calculated formula is as follows:

$$d(box, centroid) = 1 - IoU(box, centroid) \quad (5)$$

Centroid is the clustering center, box is the ground truth in the training set, *IoU* is the ratio of the intersection to the union of the areas of two rectangular boxes. Since YOLOv3 uses three-scale prediction, each scale requires three sizes of anchor boxes, so a total of nine anchor boxes need to be prepared. After clustering, their dimensions are as follows (based on WIDER FACE dataset and MAFA data set): (12×15), (20×27), (31×41), (42×54), (58×75), (78×105), (114×152), (174×226), (316×390). Finally, the clustering results are assigned to feature maps of three scales according to the size.

F. Loss Function

The loss function of the YOLOv3 network is composed of the three different loss functions: position loss (L_{xywh}), confidence loss (L_{conf}) and classification loss (L_{cla}). Calculate the three loss values and sum them to get the total loss of YOLOv3. The loss function formula is as follows:

$$Loss = L_{xywh} + L_{conf} + L_{cla} \quad (6)$$

$$L_{xywh} = \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} I_{i,j}^{obj} [(x_i - \hat{x}_i^j)^2 + (y_i - \hat{y}_i^j)^2]$$
$$+ \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} I_{i,j}^{obj} [(\sqrt{w_i^j} - \sqrt{\hat{w}_i^j})^2 + (\sqrt{h_i^j} - \sqrt{\hat{h}_i^j})^2] \quad (7)$$

$$L_{conf} = -\sum_{i=0}^{s^2} \sum_{j=0}^{B} I_{i,j}^{obj} [\hat{c}_i^j \log(c_i^j) + (1 - \hat{c}_i^j) \log(1 - c_i^j)]$$
$$-\lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^{B} I_{i,j}^{noobj} [\hat{c}_i^j \log(c_i^j) + (1 - \hat{c}_i^j) \log(1 - c_i^j)] \quad (8)$$

$$L_{cla} = -\sum_{i=0}^{s^2} I_{i,j}^{obj} \sum_{c \in classes} [\hat{p}_i^j \log(p_i^j) + (1 - \hat{p}_i^j) \log(1 - p_i^j)] \quad (9)$$

After the image is input to the network, it will be divided into S×S grid cells, each grid cell generates B boxes, each boxes get the corresponding bounding box after passing through the network, and finally gets S×S×B bounding boxes. The neural network uses the loss function to calculate the loss of the predicted value and the ground truth, and updates the weights. In position loss, λ_{coord} is the coordinate coefficients, x_i, y_i, w_i^j and h_i^j are the predicted values of the bounding box's center coordinates and width and height, $\hat{x}_i^j$, $\hat{y}_i^j$, $\hat{w}_i^j$ and $\hat{h}_i^j$ represent ground truth. In confidence loss, λ_{noobj} indicates that the confidence coefficient of no object, $I_{i,j}^{obj}$ and $I_{i,j}^{noobj}$ are positive samples and negative samples. The former represents whether the j anchor box in the i grid cell serves as predicting the object. If true, value is 1, otherwise, value is 0. The latter represents that the j anchor box in the i grid cell is not responsible for predicting the object. c_i^j and $\hat{c}_i^j$ are the predicted value and the true value of object confidence, respectively. In the classification loss, p_i^j and $\hat{p}_i^j$ indicates that the prediction object is the predicted value and the true value of each class.

III. The Proposed Algorithm

This section mainly introduces the techniques and improvement strategies. Firstly, the basic technology of MobileNetv3 is introduced. Secondly, the parameters and computations of depthwise separable convolution and ordinary convolution are compared. Then the basic block of MobileNetv3 is compared with the basic block of MobileNetv2, and the improvements are analyzed. In addition, we optimized the loss function for bounding box regression and improved the algorithm for initial clustering of anchor boxes. At the end of this section, the entire idea and detection process of the YOLOv3-M3 algorithm is introduced in detail.

A. MobileNetv3 Network

The MobileNetv3 network continues to use the deeply separable convolutional technique in MobileNetv1 and the linear bottlenecks and inverted residual blocks in MobileNetv2, and adds a squeeze and excitation (SE) attention mechanism module. Moreover, MobileNetv3 uses (Neural Architecture Search) NAS technology to search its network structure and parameters and improve search results [10]. Meanwhile, the swish activation function is improved to H-swish to reduce computation amount, and the feature extraction capability is strengthened while maintaining the characteristics of small volume model.

B. Depthwise Separable Convolution

As the main technology of the MobileNetv3 network, the depthwise separable convolution greatly reduces parameters and computation, thereby improving the training speed of the model. This convolution includes depthwise convolution and pointwise convolution.

Assuming that the dimension of the input feature graph is $D_F \times D_F \times M$, the size of the convolution kernel is $D_K \times D_K$, and the number of the convolution kernel is N. The computations of ordinary convolution is C_{ord}, and the computations of depthwise separable convolution consists of addition of C_{dep} and C_{poi}, the calculation formulas are as follows:

$$C_{ord} = D_K \times D_K \times M \times D_F \times D_F \times N \tag{10}$$

$$C_{dep} = D_K \times D_K \times 1 \times D_F \times D_F \times M \tag{11}$$

$$C_{poi} = 1 \times 1 \times M \times D_F \times D_F \times N \tag{12}$$

Then the ratio of parameters and computations between depthwise separable convolution and ordinary convolution are as follows:

$$\frac{C_{dsc}}{C_{ord}} = \frac{D_K{}^2 \times D_F{}^2 \times M + D_F{}^2 \times N \times M}{D_K{}^2 \times D_F{}^2 \times N \times M} = \frac{1}{N} + \frac{1}{D_K{}^2} \tag{13}$$

$$\frac{P_{dsc}}{P_{ord}} = \frac{D_K{}^2 \times M + M \times N}{D_K{}^2 \times M \times N} = \frac{1}{N} + \frac{1}{D_K{}^2} \tag{14}$$

The measurement of depthwise convolution kernel is generally 3×3 and the number of channels is large, $1/N$ is negligible, so the computation and parameter number of depthwise separable convolution can be reduced to about 1/9 of that of ordinary convolution, which substantially improves the training efficiency.

C. The Basic Block

The basic block adopted by the MobileNetv2 network is shown in Fig.6, a 1×1 convolutional layer is used for dimension-raising operation, followed by a batch normalization structure and a *ReLU6* activation function. Next is a 3×3 depthwise convolution and a 1×1 pointwise convolution. Finally, when the stride is 1 and the channels of the input and output matrices are equal, the input matrix and the output matrix are connected shortcut. However, in MoboleNetv3 the basic blocks have been updated, SE modules are added and the activation function is updated.

The basic block structure of MobileNetv3 is shown in Fig.7. Firstly, an SE module is added to perform global average pooling on the data of the input feature matrix [11]. Then, it is connected with two fully connected layers and the corresponding activation functions are *ReLU* and *hard-sigmoid*, respectively. Finally, the obtained data is multiplied with the input feature matrix to obtain the final feature matrix. Since the activation functions used by each layer are different, the NL uniformly represents nonlinear activation functions.

D. Optimized the Regression Loss

IoU is one of the most commonly used evaluation indexes for object detection algorithms, and its meaning is the ratio of the intersection and union of the areas of two rectangular boxes. The calculation formula is as follows:

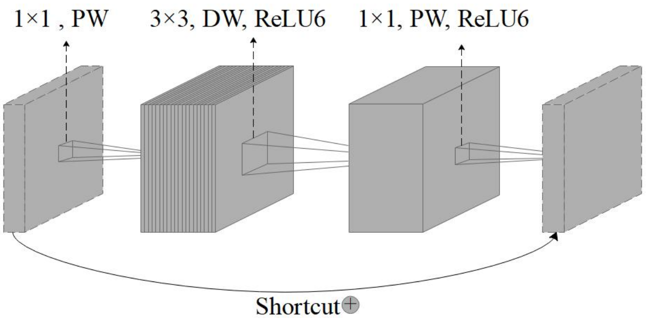


Fig. 6. The basic block of MobileNetv2.

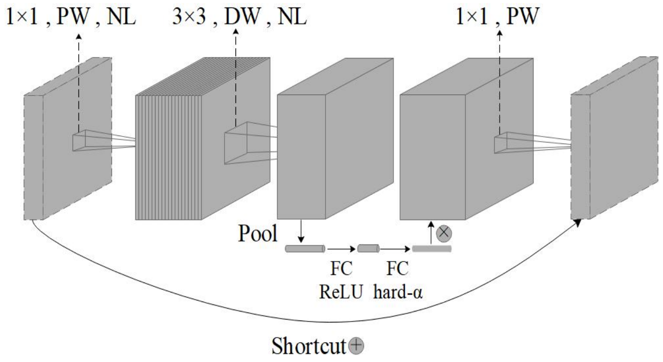


Fig. 7. The basic block of MobileNetv3.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{15}$$

A and B are the areas of the bounding box and ground truth. While *IoU* can simply represent the ratio between these two dimensions, it ignores the case where the ground truth and bounding boxes do not overlap, resulting in a value of 0 for *IoU*. In this case, the loss function is not differentiable and cannot be optimized. Therefore, *CIoU* is adopted as regression loss [12]. The *CIoU* formulas are as follows:

$$CIoU = IoU - \frac{\rho^2(a,b)}{c^2} - \alpha v \tag{16}$$

$$\alpha = \frac{v}{1 - IoU + v} \tag{17}$$

$$v = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2 \tag{18}$$

In the above formula, a and b represent the central point coordinates of the anchor box and ground truth, ρ^2 is the Euclidean distance between them. c represents the diagonal distance of the minimum rectangle containing bounding box and ground truth, α is the weight function and v is the parameter to measure the aspect ratio, w, h, w^{gt} and h^{gt} are the widths and heights of the predicted values and the ground truth, respectively. The corresponding loss function formula is as follows:

$$Loss_{CIoU} = 1 - IoU + \frac{\rho^2(a,b)}{c^2} - \alpha v \tag{19}$$

TABLE I
THE STRUCTURE OF MOBILENETV3-LARGE.

No.	Input	Conv	Input channels	Output channels	SE	NL	Stride	Output
1	416×416	Conv2d	3	16	×	Hard-Swish	2	208×208
2	208×208	Basic block-shortcut, 3×3	16	16	×	ReLU	1	208×208
3	208×208	Basic block, 3×3	16	24	×	ReLU	2	104×104
4	104×104	Basic block-shortcut, 3×3	24	24	×	ReLU	1	104×104
5	104×104	Basic block, 5×5	24	40	√	ReLU	2	52×52
6	52×52	Basic block-shortcut, 5×5	40	40	√	ReLU	1	52×52
7	52×52	Basic block-shortcut, 5×5	40	40	√	ReLU	1	52×52
8	52×52	Basic block, 3×3	40	80	×	Hard-Swish	2	26×26
9	26×26	Basic block-shortcut, 3×3	80	80	×	Hard-Swish	1	26×26
10	26×26	Basic block-shortcut, 3×3	80	80	×	Hard-Swish	1	26×26
11	26×26	Basic block-shortcut, 3×3	80	80	×	Hard-Swish	1	26×26
12	26×26	Basic block, 3×3	80	112	√	Hard-Swish	1	26×26
13	26×26	Basic block-shortcut, 3×3	112	112	√	Hard-Swish	1	26×26
14	26×26	Basic block, 5×5	112	160	√	Hard-Swish	2	13×13
15	13×13	Basic block-shortcut, 5×5	160	160	√	Hard-Swish	1	13×13
16	13×13	Basic block-shortcut, 5×5	160	160	√	Hard-Swish	1	13×13
17	13×13	Conv2d,1×1	160	960	×	Hard-Swish	1	13×13
18	13×13	Pool,7×7	960	—	×	—	1	1×1
19	1×1	Conv2d 1×1	960	1280	×	Hard-Swish	1	1×1
20	1×1	Conv2d 1×1	1280	—	×	—	1	1×1

CIoU loss function not only improves the problems existing in *IoU* loss function, but also adds penalty factor to make the regression process more stable.

E. K-Means++ clustering algorithm

The network adjusts the size of the anchor box through training, and the speed of network training can be improved by optimizing the initial size of the anchor box. YOLOv3 uses the K-Means algorithm to cluster all ground truths to get the initial anchor box, but this method has some shortcomings. The selection of initial points will affect the speed of convergence, while K-Means algorithm needs to specify the quantity of points and positions are random, resulting in local optimization rather than global optimization [13].

The YOLOv3-M3 algorithm improves the initial clustering algorithm of anchor box, replacing K-Means clustering algorithm with K-Means++ clustering algorithm. The implementation processes of the algorithm are as follows:

Step 1. From the dataset $X = \{x_1, x_2, ..., x_n\}$, a sample point x_i is randomly selected as the initial clustering center c_1.

Step 2. Connect each sample point x_i and all cluster centers into a straight line, and calculate the length of the line, as shown by $D(x_i)$. After that, take the ratio of the square of each $D(x_i)$ to the sum of the squares of all $D(x_i)$ as the probability of the next center, the formula is as follows:

$$P = \frac{D^2(x_i)}{\sum_{i=1}^{n} D^2(x_i)} \tag{20}$$

Step 3. Repeat Step 2 until k cluster centers are selected $C = \{c_1, c_2, ..., c_k\}$.

Step 4. Calculate the line distance from each point x_i to k cluster centers. When the line distance between a point x_i and a cluster center is the shortest, assign the point x_i to the class corresponding to the cluster center. For each class ci, recalculate its cluster center, the calculation formula is as follows:

$$c_i = \frac{\sum_{x_i \in c_i} x_i}{|c_i|} \tag{21}$$

Step 5. Repeat Step 3 and Step 4 until the location of the cluster center remains stable.

Using K-Means ++ algorithm can reduce the loss value at the initial training stage, reduce the number of iterations required for convergence, and accelerate the convergence speed.

F. Detection Process of YOLOv3-M3

MobileNetv3 network contains two network structures, large and small. Considering the precision problem, MobileNetv3-Large network is the main model in this paper. The network structure is shown in Table I. The YOLOv3-M3 algorithm overall structure is as follows: the backbone network is MobileNetv3-Large instead of DarkNet-53. We adjusted the network structure of the original MobileNetv3-Large to delete layers 17-20 in Table I. In addition, in the detection header part of YOLOv3, we continue to use FPN. Firstly, in each feature fusion branches after YOLOv3 convolutional set, we continue to use the 1×1

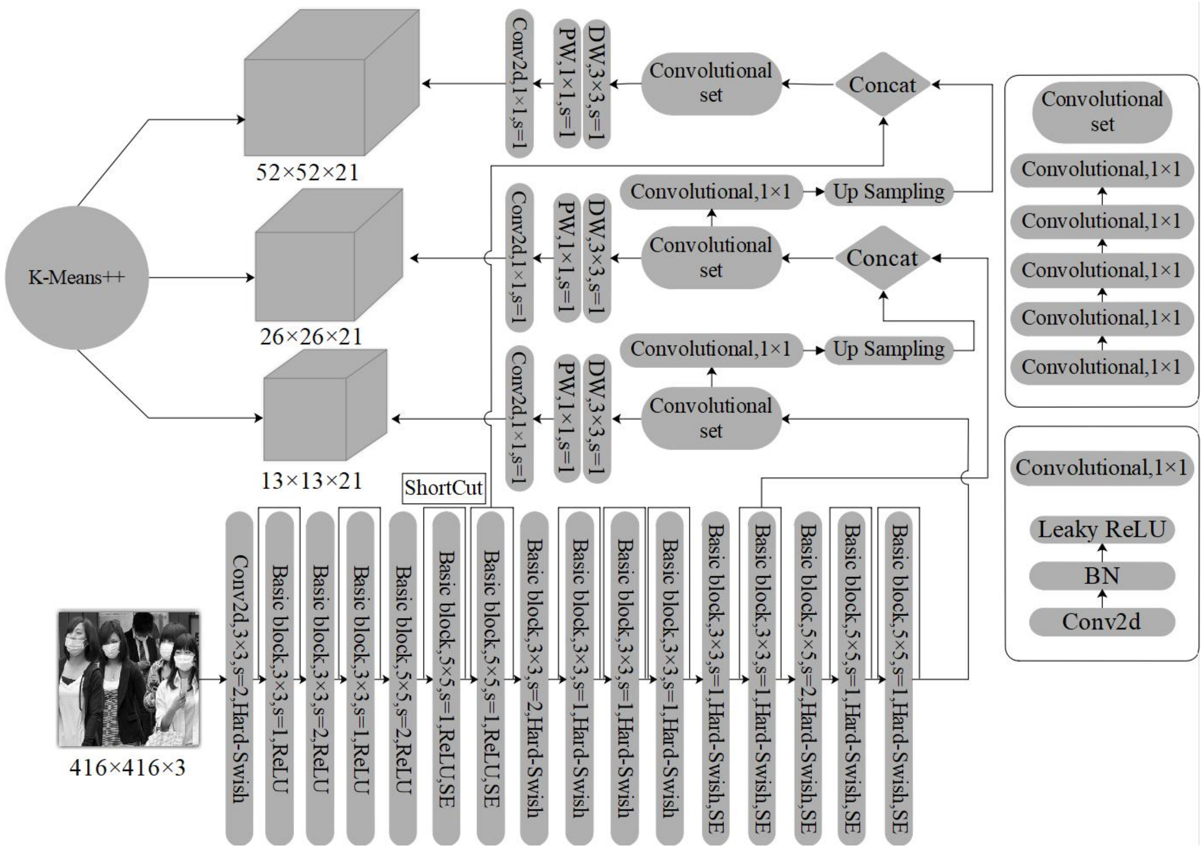


Fig. 8. The structure of YOLOv3-M3.

convolutional block, and up sampling the feature map. Secondly, the features of layer 16, layer 13, and layer 7 in the MobileNetv3 network are sequentially concatenated, the concatenating process is in step 2 of the detection process. Then the 3×3 ordinary convolution in each prediction branches are changed to the depthwise separable convolution, and reduce the number of convolution layers. Finally, the anchor box is allocated to the corresponding prediction layer using the clustering results. The structure of YOLOv3-M3 is show in Fig.8.

The detection process of YOLOv3-M3 object detection algorithm includes training process and prediction process. The training process is mainly composed of backbone network MobileNetv3 and multi-scale prediction layer. The prediction process mainly includes calculating the information of bounding box and non-maximum suppression. The detection process of YOLOv3-M3 is as follows:

Step 1. Firstly, images with dimensions of 416×416×3 are input. After calculating through conv2d and basic block structures of MobileNetv3 network, output 7, 13 and 16 layers of the network, the feature map dimensions are 52×52×40, 26×26×112 and 13×13×160 respectively.

Step 2. In the multi-scale prediction layer, the first layer outputs a 13×13 feature map. 2x up sampling the resulting 13×13 feature map, and the channel concatenated and feature fusion are carried out with the feature map of 26×26 to form the output feature map at the second layer. After splicing the feature map of the second layer, continue 2x up sampling is performed, and the same operation is performed with the feature map of 52×52 of the previous layer, and the output the feature map of the third layer.

Step 3. The nine anchor boxes with different sizes obtained by the improved clustering algorithm are allocated to three feature maps with different scales according to their sizes. After this, the model will predict the location information for the bounding box, the offset of the horizontal length and vertical length, the probability of containing the object, and the class score based on the information of the 9 anchor boxes. The model will fit the predicted value with the real label value, train with the loss function for back propagation and gradient descent, iterate and update the parameters of the network, and get the final weight value. At the same time, the regression loss of the bounding box is modified to *CIoU*.

Step 4. In the prediction process, the image is firstly input into YOLOv3-M3 model, the model calculates and outputs the predicted values (t_x, t_y, t_w, t_h, obj, cls) for each bounding boxes. Secondly, the final confidence scores need to be calculated. The confidence score is the conditional class probability for each class. The calculation formula is as follows:

$$c\_scores = obj \times \begin{bmatrix} c(face) \\ c(face\_mask) \end{bmatrix} \tag{22}$$

The *c_scores* is the final confidence score, *obj* is the probability of the bounding box have object, *c*(*face*) and *c*(*face_mask*) represent the probability that the object is face or wearing masks. We set the threshold is 0.6 to delete the bounding box with low scores. Finally, remove redundant bounding boxes of the same object by non-maximum suppression, and the bounding box with the highest *c_scores* is retained as the predicted box.

IV. Experiment

In this section, we evaluate the YOLOv3-M3 algorithm. Firstly, we introduce the selection of datasets and the setting of experimental parameters. Secondly, the effects of clustering results and loss function on the model results are compared, and the improved model is evaluated. Eventually, we conducted a comparative experiment of different algorithms on face mask datasets.

A. Experiment Environment and Data Set

The data set for this experiment consists of the publicly available MAFA dataset and the WIDER FACE dataset [14], the training set consists of 2193 images. Among them, 1078 images are of people without masks and the label of images are "face", 1115 images are of people wearing masks and the label of images are "face_mask". In addition, the test data set are 1,059 images. In Fig.9 (A) and (B), images are from the train data, in Fig.9 (C) and (D), images are from test data. The ground truth is shown in Table II.

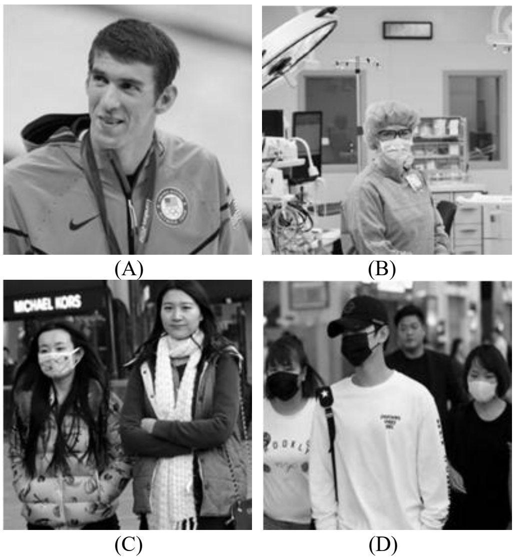


Fig. 9. Images in the dataset.

TABLE II
THE INFORMATION OF GROUND TRUTH

Label Name	Information
face(A)	(337,220,642,586,0)
face_mask(B)	(138,129,184,202,1)
face_mask(C)	(200,53,250,112,1)
face(C)	(347,10,398,68,0)
face_mask(D)	(153,91,225,191,1)
face_mask(D)	(285,55,367,138,1)
face(D)	(387,50,438,112,0)
face_mask(D)	(509,117,576,195,1)

The rationality of the size of the anchor box has a very important influence on the convergence rate. If the size of the anchor box is inconsistent with the scale of the detected object, it will lead to a large number of missed and false detections. In order to avoid this problem, an improved clustering algorithm is used to cluster the artificial annotated boxes. The comparison results of the two clustering algorithms are shown in Table III.

TABLE III
COMPARISONS OF CLUSTERING RESULTS

Algorithm	Measure	The size of anchor box		
		Small	Medium	Large
K-Means	13×13	(114,152)	(174,226)	(316,390)
	26×26	(42,54)	(58,75)	(78,105)
	52×52	(12,15)	(20,27)	(31,41)
K-Means++	13×13	(121,124)	(133,206)	(234,270)
	26×26	(41,58)	(60,88)	(79,113)
	52×52	(8,13)	(17,29)	(33,40)

The parameters set in this experiment are as follows: we set epoch to 100, the first 50 iterations were frozen training. The last 50 iterations were thawing training. In addition, we set batch size to 32, the learning rate being set to 0.0001, the training data and validation data were divided in a ratio of 9:1.

B. Model Evaluation and Comparative Experiment

In the model evaluation stage. Firstly, we compare the loss values of the three models. Secondly, in order to verify the advantages of the K-Means++ clustering algorithm and the *CIoU* loss function, compare the loss value image and *mAP* 0.5 value image of the unimproved and improved YOLOv3-M3. Thirdly, the validation set evaluation model is used to draw the P-R curve and obtain the average precision of each class, and calculate the final Mean Average precision. Finally, the YOLOv3-M3 object detection algorithm is compared with other object detection algorithms, and the test sets are input into YOLOv3 and YOLOv3-M3 model, and the test results are compared.

The loss value is an important condition to measure the quality of the model. We compare the train loss values and validation loss values during training of the YOLOv3 model, YOLOv3-M3(N) without improved clustering algorithm and loss function, and the improved YOLOv3-M3 model. The comparison of loss values of three modules are shown in Table IV.

TABLE IV
COMPARISON OF LOSS VALUES OF THREE MODELS

Model	Train loss		Validation loss	
	Initial loss	Final loss	Initial loss	Final loss
YOLOv3	133.14	10.35	69.2	10.61
YOLOv3-M3(N)	202.52	7.80	159.18	13.66
YOLOv3-M3	142.57	5.76	95.04	6.61

Based on YOLOv3-M3, we improved the clustering algorithm and the loss function to obtain the training loss value image. As shown in Fig.10, the loss value of *IoU* and *CIoU* loss function is close at the beginning of iteration, but the convergence speed of *CIoU* loss function is significantly faster than that of *IoU* loss function. After the clustering algorithm is improved to K-Means++, the loss value is significantly reduced in the initial stage, and the convergence speed is faster, the model is more stable, and the final loss value is the lowest.

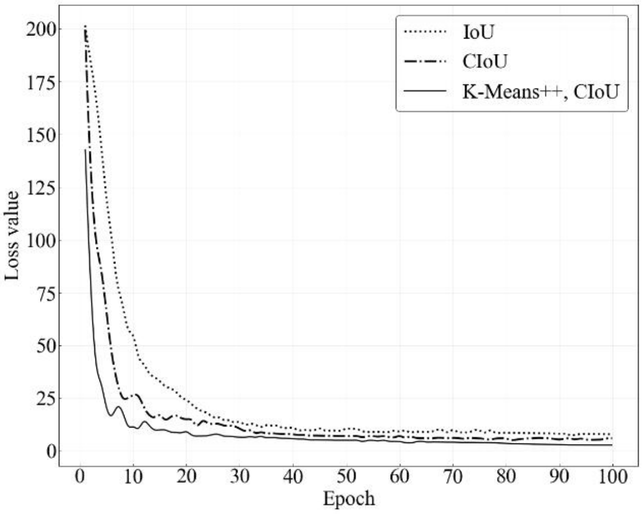


Fig. 10. Comparison of loss value for *IoU*, *CIoU* and K-Means++, *CIoU*.

We adopted some metrics as the evaluation indexes of this experiment. These formulas are as follows:

$$precision = \frac{TP}{TP + FP} \tag{23}$$

$$recall = \frac{TP}{TP + FN} \tag{24}$$

$$AP = \int_0^1 P(R)\, dR \tag{25}$$

$$mAP = \frac{1}{C} \sum_{c \in C} AP(c) \tag{26}$$

In this experiment, TP means the quantity of samples with face masks accurately detected. FP means the quantity of samples with face masks incorrectly detected. FN means the quantity of samples with face masks but not detected by the algorithm. *AP* and *mAP* are the most commonly used evaluation indexes of algorithm detection precision. The *AP* value means average precision, and the area under the precision (P) - recall (R) curve formed by precision value and recall value is *AP* value. The *mAP* value is the average of the sum of *AP* precision in all classes (C), the *mAP 0.5* is the AP value when *IoU*=0.5.

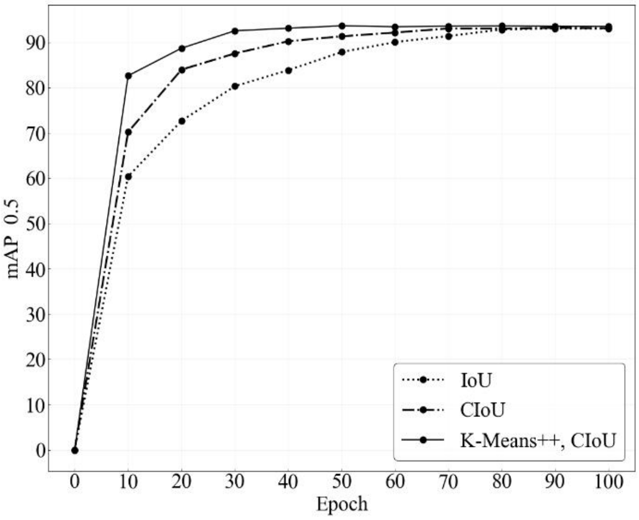


Fig. 11. Comparison of *mAP 0.5* for *IoU*, *CIoU* and K-Means++, *CIoU*.

Fig.11 shows the *mAP 0.5* value of the YOLOv3-M3 model in three different cases. According to the change trend of *mAP 0.5* value, the improved YOLOv3-M3 has absolute advantages in both initial value and convergence speed, which reflects the advantages and characteristics of *CIoU* loss function and K-Means++ clustering algorithm.

Fig.12 and Fig.13 are P-R curves of two classes (face and face_mask), and their *AP 0.5* values are 91.12% and 95.57%, respectively. The *mAP* value of the YOLOv3-M3 algorithm is 93.51%.

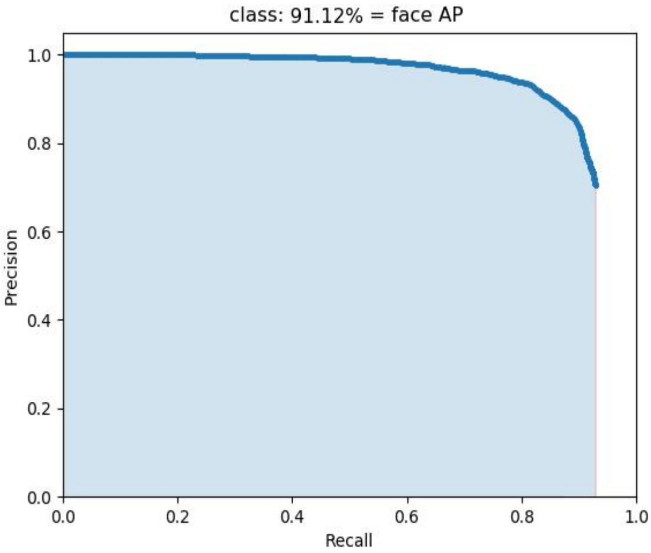


Fig. 12. The P-R curve of face.

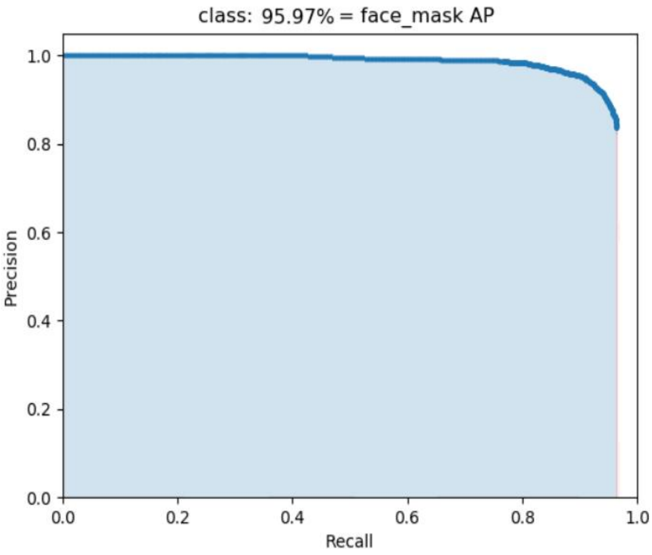


Fig. 13. The P-R curve of face_mask.

In this paper, the main indicators of the comparison experiment are the size of the model, the detection speed of the algorithm and the *mAP*. The size of the model refers to the size of the model's weight file after training, the detection speed refers to the number of images that can be detected by the model in 1 second; *mAP 0.5* and *mAP 0.75* are mean Average precision when *IoU=0.5* and *0.75*. In addition, the same detection algorithm used different backbone networks for experimentation. The comparisons of different detection algorithms are shown in Table V.

TABLE V
COMPARISONS OF DIFFERENT DETECTION ALGORITHMS BASED ON FACE-MASK DATASETS

Algorithm		Model Size /MB	Detection Speed /(frame·s^{-1})	mAP 0.5/%	mAP 0.75/%
Object Detection Algorithm	Backbone Network				
Faster R-CNN	VGG16	521.5	0.7	94.2	72.2
	ResNet50-FPN	110.2	0.8	93.6	71.6
SSD	VGG16	100.8	4.7	87.3	66.3
SSDLite	MobileNetv3-Small	7.2	20.5	82.7	61.5
	MobileNetv3-Large	14.9	15.7	85.8	65.4
YOLOv3	DarkNet53	235.2	2.1	90.4	69.3
	MobileNetv3-Small	3.8	17.2	87.9	64.1
	MobileNetv3-Large	26.5	13.6	93.0	71.5
YOLOv3-Tiny	DarkNet-Tiny	35.3	8.9	85.4	65.2
YOLOv3-M3	MobileNetv3-Large	26.5	13.6	93.5	71.9

Compared with the original YOLOv3 algorithm, the *mAP* value and detection speed of the YOLOv3-M3 algorithm are greatly improved. The *mAP 0.5* and *mAP 0.75* are 93.5% and 71.9%, respectively, which are 3.1% and 1.6% higher than the original YOLOv3 algorithm. The detection speed of YOLOv3-M3 is 13.6 frame/s, which is about 7 times that of the original algorithm, and the model size is only 26.5M, which is nearly 10 times smaller than the original algorithm. Although the precision of Faster R-CNN algorithm exceeds 93%, the detection speed is only 0.7-0.8 frame/s. As a lightweight detection algorithm, SSDLite has a high detection speed, but the detection precision is not as high as the algorithm in this paper.

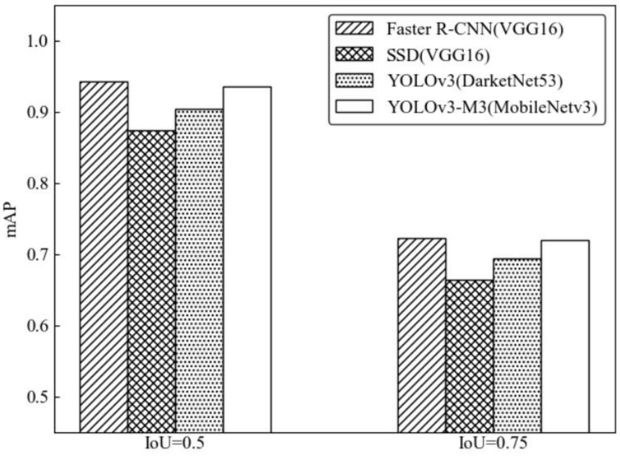


Fig. 14. The *mAP* values of the four algorithms when *IoU*=0.5 and *IoU*=0.75.

Fig. 14 shows the bar chart of *mAP 0.5* and *mAP 0.75* values for the four algorithms. The values of YOLOv3-M3 and Faster-RCNN are close, and higher than SSD and YOLOv3.

TABLE VI
CROSS VALIDATION YOLOv3-M3 ALGORITHM ON FACE-MASK DATASETS

Index	Test1	Test2	Test3	Test4	Test5	Average value
mAP 0.5(%)	91.86	93.80	92.82	92.06	91.53	92.41
mAP 0.75(%)	69.22	71.22	71.53	70.03	70.34	70.47

The data set is divided into 5 parts, 1 part is selected as the test set, and the other is used as the training set, which is repeated 5 times. As shown in Table VI, Test1-Test5 are five different test results. On *mAP 0.5* and *mAP 0.75*, the average values are 92.41% and 70.47%, respectively, proving that YOLOv3-M3 is a reliable and stable algorithm.

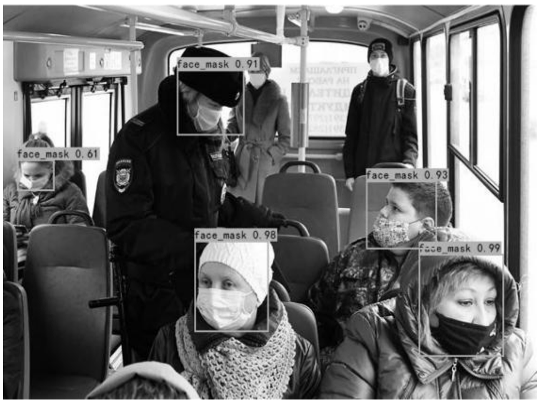


Fig. 15. The detection result of YOLOv3 algorithm.

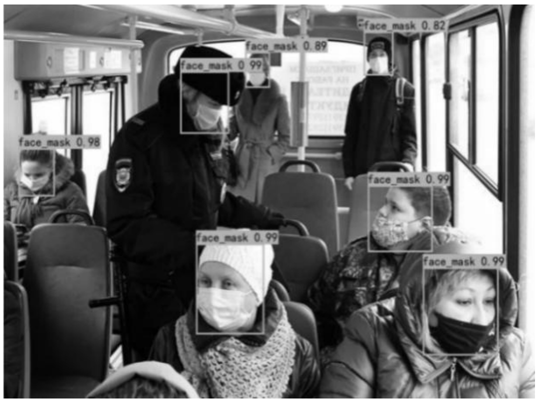


Fig. 16. The detection result of YOLOv3-M3 algorithm.

Fig.15 and Fig.16 show the detection results of two algorithms in the test dataset. The YOLOv3 algorithm has a high precision for close-range objects in complex environments, but the detection effect of the distant objects is not ideal. In contrast, the YOLOv3-M3 algorithm in this paper also has good detection effect for long-range objects.

V. CONCLUSIONS

This research proposes an object detection algorithm, YOLOv3-M3, which replaces the YOLOv3 backbone network with a lightweight MobileNetv3-Large network. Meanwhile, the initial clustering algorithm of anchor box is updated as K-Means++ and the loss function is modified as *CIoU* on this basis. The algorithm is trained on the public mask data set. By adjusting the experimental parameters, the algorithm is trained on the public mask dataset, and the experimental results are compared with other algorithms. The detection precision of YOLOv3-M3 has been greatly improved comparing with SSD, SSDLite, YOLOv3 and YOLOv3-Tiny. The precision is very close to the two-stage detection algorithm, and the detection speed can reach 13.6 frame/s. It can be used for video detection and camera real-time detection, which is an efficient object detection model.

REFERENCES

[1] S. Shiehzadegan, N. Alaghemand, M. Fox and V. Venketaraman, "Analysis of the Delta Variant B.1.617.2 COVID-19," *Clinics and Practice*, vol. 11, no. 4, pp. 778-784, 2021.

[2] S. Singh, U. Ahuja, M. Kumar, K. Kumar and M. Sachdeva, "Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment," *Multimedia Tools and Applications*, vol. 80, no. 13, pp. 11-16, 2021.

[3] J. Wu, B. Peng, Z. Huang and J. Xie, "Research on Computer Vision-Based Object Detection and Classification," *Computer and Computing Technologies in Agriculture VI*, vol. 392, pp. 183-188, 2012.

[4] J. X. Wan, W. Ding, H. L. Zhu, X. Ming, Z. K. Huang, T. Li, Y. X. Zhu and H. Wang. "An Efficient Small Traffic Sign Detection Method Based on YOLOv3," *Journal of Signal Processing Systems*, vol. 93, pp. 899-911, 2021.

[5] T. M. Deng and Y. J. Wu, "Simultaneous vehicle and lane detection via MobileNetV3 in car following scene," *Public Library of Science*, vol. 17, no. 3, pp. 551-569, 2022.

[6] S. Sethi, M. Kathuria and T. Kaushik, "Face Mask Detection using Deep Learning: An Approach to Reduce Risk of Coronavirus Spread," *Journal of Biomedical Informatics*, vol. 120, pp. 1-12, 2021.

[7] J. M. Yu and W. Zhang, "Face Mask Wearing Detection Algorithm Based on Improved YOLO-v4," *Sensors*, vol. 21, no. 9, pp. 263-284, 2021.

[8] F. Mercaldo and A. Santone, "Transfer Learning for Mobile Real-Time Face Mask Detection and Localisation," *Journal of the American Medical Informatics*, vol. 28, no. 7, pp. 1548-1554, 2021.

[9] Z. G. Li, J. H. Zhang, B. Li, X. Y. Gu and X. D. Luo, "COVID-19 Diagnosis on CT Scan Images Using a Generative Adversarial Network and Concatenated Feature Pyramid Network with an Attention Mechanism,"*Medical Physics*, vol. 48, no. 8, pp. 4334-4349, 2021.

[10] E. M. Abd, A.Dahou, N. A. Alsaleh, A. H. Elsheikh, A. I. Saba and M. Ahmadein, "Boosting COVID-19 Image Classification Using MobileNetV3 and Aquila Optimizer Algorithm," *Entropy*, vol. 23, no. 11, pp. 1383-1403, 2021.

[11] Y. K. Liu, G. P. Yang, Y. W. Huang and Y. L. Yin, "SE-Mask R-CNN: An improved Mask R-CNN for apple detection and segmentation," *Journal of Intelligent & Fuzzy Systems*, vol. 41, no. 6, pp. 6715-6725, 2021.

[12] J. J. Gao and T. Yang, "Face detection algorithm based on improved TinyYOLOv3 and attention mechanism," *Computer Communications*, vol. 181, pp. 329-337, 2022.

[13] D. Y. Jia, Z. H. He, C. W. Zhang, W. T. Yin, N. K. Wu and Z. Q. Li, "Detection of cervical cancer cells in complex situation based on improved YOLOv3 network," *Multimedia Tools and Applications*, vol. 81, no. 6, pp. 8939-8961, 2022.

[14] J. L. Zhang, X. W. Wu, C. H. H. Steven and J. K. Zhu, "Feature agglomeration networks for single stage face detection," *Neurocomputing*, vol. 380, no. c, pp. 180-189, 2019.