International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 11, Issue 3 [May.-June. 2022] PP: 754-765

Face Mask Recognition System Based on YOLOv5 for Prevention and Control COVID-19

Hangao ZHANG, Shuaikang YAN, Yan ZHANG, Chengke LV, Qiliang TONG, Kai ZHANG, Kangdi CUI, Huaping SONG,Feilong CHEN,Pingchuan ZHANG

School of Information Engineering,Henan Institute of Science and Technology, Henan, CHINA
Corresponding Author: Corresponding Author: Pingchuan ZHANG

ABSTRACT:

The rapid development of computer vision makes human-computer interaction possible and has a wide application prospect. Since the discovery of the first case of COVID-19, the global fight against the epidemic has begun. In addition to various studies and findings by medical and health care experts, people's daily behaviors have also become key to combating the epidemic. In China, the government has taken active and effective measures of isolation and closure, as well as the active cooperation of the general public, such as it is unnecessary to stay indoors and wear masks. China, as the country with the first outbreak of the epidemic, has now become the benchmark country of epidemic prevention in the world. Of course, it is not enough for people to wear masks consciously. Wearing masks in all kinds of public places still needs supervision. In this process, this paper proposes to replace manual inspection with a deep learning method and use YOLOv5, the most powerful objection detection algorithm at present, to better apply it in the actual environment, especially in the supervision of wearing masks in public places. The experimental results show that the algorithm proposed in this paper can effectively recognize face masks and realize the effective monitoring of personnel.

Date of Submission: 07-06-2022 Date of Acceptance: 22-06-2022

I. INTRODUCTION

Computer vision technology uses a variety of imaging systems instead of the visual organ as the input means, and uses the computer instead of the brain to complete the processing and interpretation of visual information. With the continuous development of computer vision technology, computers can recognize various faces and provide feedback. Now, mask face recognition is most widely used in public places. Since the first case of pneumonia of unknown cause appeared in Wuhan, China in late 2019, the world has been plagued by a new epidemic.

Wearing a mask has become the most effective way for people around the world to avoid infection. Globally, masks were in short supply for a time, and merchants even raised the price of masks dozens of times until the market supply could meet the needs of consumers. Therefore, masks have become a must for personal travel. As masks have become a necessity, many public places where people gather, such as malls and swimming pools, require wearing monitoring, but today most Chinese shopping centers still use manual and restricted access methods to monitor the flow of people wearing masks. Although manual inspection has certain defects, it can be avoided by increasing the manpower. Although China's epidemic prevention and control is very good, in shopping malls and other places with large and scattered populations, the occurrence of infection cannot be completely avoided, because sending people to supervise will not only waste a lot of resources and manpower. At every entrance of the mall. In addition, if there is a lot of traffic at one of the entrances, it may cause the staff to miss the inspection. In addition, manual supervision also wastes time, causing a large number of people to gather at the door of shops and supermarkets, causing infection risks.

After the outbreak of the COVID-19, higher requirements have been placed on the public health protection of the public. At present, people must wear masks when entering and leaving public places and taking public transportation. Mask wearing inspection has become an essential operation for epidemic prevention and control [1]. In recent years, deep learning technology has been widely used in the field of object recognition [2]. Feng Guochen et al. conducted in-depth research on the automatic identification of helmets by using deep learning related methods [3]. Li Meiling et al. used a convolutional neural network model to extract road information

from high-resolution remote sensing images [4]. Some scholars have also conducted research on the recognition of mask wearing. For example, Zhang Xiubao and others used the Fast-RCNN algorithm to conduct research on the recognition of face wearing masks in all-weather natural scenes [5].

In order to realize the effective recognition of mask wearing, this paper proposes a mask wearing recognition method using the improved YOLOv5 model. The development of image recognition technology has gone through three stages. ①character recognition stage; ②image processing and recognition stage; ③object recognition stage. At present, the key research direction in the field of image recognition is classification and recognition in object recognition, which has been widely used in the field of security, transportation and the Internet. Object classification and recognition are mainly based on feature learning. In 2016, RedmonJ et al. proposed the YOLO algorithm [6]. Using the YOLO algorithm to perform feature extraction, classification and recognition on the target in the image can realize the automation of image feature extraction and classification recognition, and abandon the method of manually labeling image features in the traditional image recognition process. Its network structure is built on the GoogleNet model. of. The YOLO detection framework regards the target detection problem as a regression problem, and regresses the position and category of the target by dividing the grid. YOLO divides the picture into 7x7, and then generates such a 7x7 output through the convolutional neural network. Each output in the 7x7 predicts the target whose center point falls on this grid. The predicted target parameters include the target category and target box location. The YOLO algorithm is mainly implemented in three steps. First, normalize the input image soft size; second, convolution network feature extraction, predict the confidence of the bounding box; finally, filter the bounding box through the non-maximum suppression algorithm to obtain the optimal result . Compared with the Faster R-CNN algorithm, this unified model realizes end-to-end training and prediction, with faster detection speed, lower background false positive rate, better generalization ability and robustness. However, since each cell only makes bounding box predictions for the same set of categories, the localization accuracy of the YOLO algorithm suffers. Due to the way YOLO divides the grid, it is impossible to obtain enough candidate grids to predict the target for relatively dense targets, resulting in too many missed detections. YOLO is also not good at detecting small targets, mainly because the grid division is relatively rough, and the characteristics of small targets cannot be well preserved. These reasons all cause the low detection accuracy of YOLO. After that, YOLOv2 was proposed again. After testing on the VOC 2007 test set, the mAP increased from 67.4% to 76.8%. Compared with the previous version, while maintaining the processing speed, V5 has improved in three aspects: more accurate prediction, faster speed, and more recognition of objects.

Through the improvement of the YOLO algorithm series, it can be seen that by continuously optimizing the algorithm, the detection speed of the YOLO algorithm can meet the requirements of real-time analysis, and meet people's needs for high-efficiency and high-precision target recognition technology. In view of the rapid development of image recognition technology, target detection algorithms such as YOLO have an extremely broad development space, which promotes the continuous development of image recognition technology. Due to the good detection performance and detection accuracy of the YOLOV5 algorithm, it has wider application significance than other versions.

II. Description of the principle of YOLO algorithm

2.1. Demand Analysis

Target recognition algorithm refers to a class of algorithms that mark the targets to be recognized from the scene and determine their positions and categories, mainly including two processes of recognition and classification. After research, the existing target recognition algorithms are mainly divided into two categories: one is the typical two-stage recognition algorithms such as R-CNN and Fast-RCNN. Based on feature extraction, this type of algorithm first generates a large number of candidate regions from independent network branches, and then performs classification and regression on them. The other is typical one-stage recognition algorithms such as SSD and YOLO [7-8]. This class of algorithms performs classification and regression while generating candidate regions.

The advantages of the two-stage recognition algorithm are mainly reflected in scalability and high accuracy, while the one-stage recognition algorithm has faster recognition speed and is more suitable for target recognition problems that require real-time detection. The YOLOV5 recognition model, which belongs to the one-stage recognition algorithm, takes into account a high accuracy rate on the basis of ensuring real-time detection. Figure 1 shows the YOLOv5s network structure.

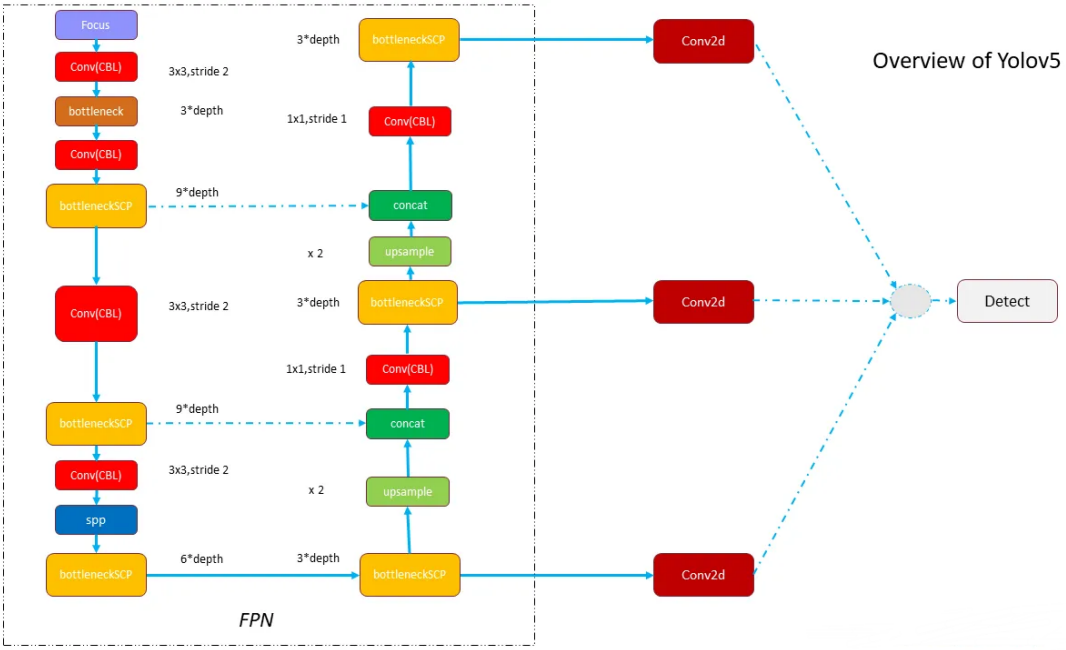

Fig. 1 YOLOv5s Network structure

YOLOv5 has the same overall architecture for networks of different sizes (n, s, m, l, x), except that different depths and widths are used in each sub-module to deal with the depth_multiple and width_multiple parameters in the yaml file respectively. It should also be noted that in addition to the n, s, m, l, and x versions, there are also n6, s6, m6, l6, x6, the difference is that the latter is for larger resolution pictures such as 1280x1280, of course, there are also some structural Difference, the latter will downsample 64 times and use 4 prediction feature layers, while the former will only downsample to 32 times and use 3 prediction feature layers.

In fact, YOLOv5 has not changed much in the Backbone part. However, YOLOv5 has a small change after the v6.0 version compared to the previous version, replacing the first layer of the network (originally the Focus module) with a 6x6 convolutional layer. The two are equivalent in theory, but for some existing GPU devices (and corresponding optimization algorithms) using a 6x6 convolutional layer is more efficient than using the Focus module. The following picture is the original Focus module (similar to Patch Merging in the previous Swin Transformer), which divides each 2x2 adjacent pixel into a patch, and then puts the pixels in the same position (same color) in each patch together. 4 feature maps are obtained, and then a 3x3 convolutional layer is connected. This is equivalent to using a 6x6 convolutional layer directly.

The changes in the Neck part are relatively large. First, the SPP is replaced by SPPF (designed by Glenn Jocher himself). I personally think this change is very interesting. The functions of the two are the same, but the efficiency of the latter higher. The SPP structure is shown in the figure below. The input is passed through multiple MaxPools of different sizes in parallel, and then further fusion is performed, which can solve the target multi-scale problem to a certain extent.

The SPPF structure is to serially pass the input through multiple 5x5 MaxPool layers. It should be noted here that the calculation result of two 5x5 MaxPool layers in series is the same as that of a 9x9 MaxPool layer. Three 5x5 layers are serialized. The size of the MaxPool layer is the same as that of a 13x13 MaxPool layer.

2.2. Algorithm flow chart of YOLOv5

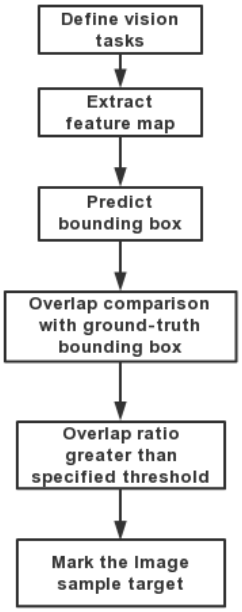

Fig. 2 Flow chart of YOLO algorithm

2.3. Detailed design of YOLOv5

2.3.1. Bounding Box Prediction

After YOLO9000, our system predicts bounding boxes using dimensional clusters as anchor boxes [9]. The network predicts 4 coordinates for each bounding box, tx, ty, tw, th. If the cell is offset from the upper left corner of the image by (cx, cy) and the bounding box prior has width and height pw, ph, the prediction corresponds to:

$$b_x = \delta(t_x) + c_x \quad (1)$$
$$b_y = \delta(t_y) + c_y \quad (2)$$
$$b_{\square} = p_h e^{t^h} \quad (3)$$
$$b_w = p_w e^{t^w} \quad (4)$$

During training, we use the sum of squared error losses. If the ground truth of some coordinate prediction is that our gradient is the ground truth (computed from the ground truth box) minus our prediction: . This ground truth value can easily be calculated by reversing the equation above. YOLOv3 predicts the target score for each boundary of the boxes using logistic regression. If the bounding box prior overlaps the ground truth object by more than 1, the value should be 1 for any other bounding box before. If the bounding box prior is not the best, but does overlap the ground truth object by more than a certain threshold, we ignore the prediction, following [10]. We use a threshold of 0.5. Unlike [10], our system only assigns a bounding box object to each ground truth. If the bounding box prior is not assigned to real objects, it does not result in loss of coordinates or class prediction, only abjectness is affected.

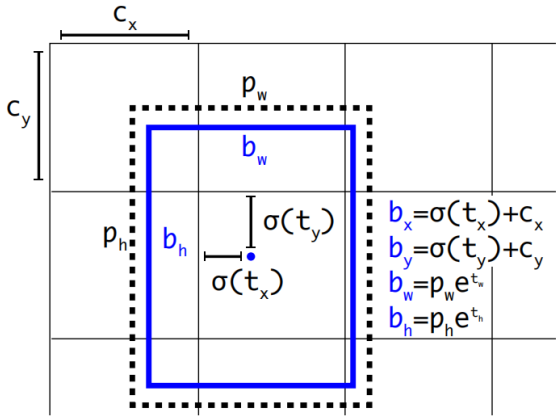

Fig. 3 Bounding box prediction with prior dimension and location

2.3.2. Classification prediction

Each box uses multi-label classification to predict the class the bounding box is likely to contain. We do not use softmax as we found that good performance is not necessary, instead we just use independent logistic classifiers. During training, we use binary cross-entropy loss prediction for classes. This formula helps when we move to more complex fields like Open Images Dataset [11]. There are many overlapping labels (i.e. woman and person) in this dataset. Using softmax assumes that each box has the correct class, which is often not the case. Multi-label methods better model the data.

2.3.3. Multiscale prediction

YOLOv3 predicts boxes at 3 different scales. Our system extracts feature from these scales using a concept similar to feature pyramid networks [12]. From our base feature extractor, we add several convolutional layers. Finally, where the prediction 3-d tensor encodes bounding box, object and class predictions. In our experiments COCO [13] we predict 3 boxes at each scale, so the tensor is $N \times N \times [3 * (4 + 1 + 80)]$ for 4 bounding box offsets, 1 Objectness prediction and 80 class predictions.
Next, we take the feature maps from the previous 2 layers and upsample them by a factor of 2. We also took the previous feature map in the network and merged it with our upsampled features using concatenation. This approach allows us to obtain more meaningful semantic information from upsampled features and a finer-grained information map from earlier features. We then add more convolutional layers to process this combined feature map and end up predicting a similar tensor, albeit now twice the size. We perform the same design again to predict the final scaled box. Therefore, our prediction scale for the 3rd time benefits from all previous computations as well as fine-grained features early in the network. We still use k-means clustering to determine our bounding box prior. We just chose 9 clusters and 3 arbitrarily scaled, then divided the clusters evenly across scales. On the COCO dataset, the 9 clusters are: (10×13), (16×30), (33×23), (30×61), (62×45), (59×119), (116× 90), (156 × 198), (373 × 326).

2.3.4. Feature extraction

We use a new network to perform feature extraction. Our new network is a hybrid approach between networks used in YOLOv2, Darknet-19 and the novelty residual network stuff. Our network uses consecutive 3×3 and 1×1 convolutional layers, but now has some shortcut connections and is significantly larger.

Table (1). Darknet53.

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

III. Mask recognition model based on YOLOv5

The YOLOv5 recognition model has the characteristics of fast recognition speed and good effect. However, in order to realize the specialized detection of small targets such as masks and improve the accuracy of mask wearing recognition, it is necessary to further optimize the network structure of the YOLOv5 model and adjust the parameters.

3.1. Mask wearing identification process

When performing mask wearing recognition, first, the face detection module performs face recognition and labeling on the input image, and then the mask wearing recognition module cuts out the sub-image of the face region from the detected face region, which is divided into masks wearing and not wearing masks. Masks are used as grouping basis for image classification, and finally the image recognition results of whether masks are worn or not are obtained. The flowchart of mask wearing recognition is shown in Figure 4.

Fig. 4 Flow chart of mask wearing identification

3.2. Adjust input size

There are many small targets in the practical application of mask wearing recognition, and the down sampling of YOLO algorithm is generally 32 times, so the width and height must be divisible by 32. Since it is generally selected as a multiple of 32 in multi-scale training, the minimum is 320 × 320, max. 608 × 608, so adjust the input size of the data to 608 × 608. To a certain extent, it can improve the accuracy of small target recognition. At the same time, due to the inconsistency of picture size, by increasing the input size, it also reduces the cropping of large pictures. In addition, the data set the letterbox function of the PY file is modified to enable the YOLOv5 model to adapt to image scaling, which solves the problem of information redundancy and reduced

reasoning speed caused by too many black edges at both ends after scaling and filling.

3.3. Improvement of initial candidate box

YOLOv5 model introduces an anchor box in the process of target recognition. The candidate region box is a set of initial regions with fixed size and aspect ratio. In model training, the closer the parameters of the initial candidate box are to the real boundary box, the easier the model will be trained, and the predicted boundary box will be more consistent with the real boundary box. In a word, the design of initial candidate frame parameters directly affects the speed of target recognition and the accuracy of target frame position. Therefore, the anchor parameter in the original YOLOv5 model needs to be adjusted according to the data set. In order to obtain the optimal anchor parameters, K-means clustering algorithm can be used to cluster the width and height of the marked target frame in the training set. According to the characteristics of the network structure of YOLOv5 model, it is necessary to find the width and height dimensions of 9 cluster centers and take them as the values of anchor parameters in the network configuration file. Therefore, the K-means clustering algorithm is used to set the parameters of 9 groups of initial candidate boxes.

3.4. Improvement of convolution layer

In the improvement of the parameters of the initial candidate frame, the width and height of the anchor can be changed, so that the corresponding relationship between the original anchor features with fixed width and height mapped to the same grid is no longer consistent. Therefore, the deformable convolution can be used to modify the original feature map. After the width and height of anchor are convoluted, a deformable convolution offset can be obtained. The features of anchor are fused into a grid, which solves the problem of feature mismatch after transformation. At the same time, since some regions have no anchor feature (such as background region), it is meaningless to perform convolution calculation on them. Therefore, the mask value of regions without anchor feature can be set to 0 to improve the calculation speed of convolution layer.

3.5. Choice of loss function

The loss function is the basis for the judgment of the false detection samples of the deep neural network. The choice of the loss function has a great influence on the convergence effect of the model, and a better recognition effect can be obtained by selecting an appropriate loss function. In the improved YOLOv5 model, GIOU_Loss is used as the loss function, and its formula is:

$$GIOU\_Loss = 1 - GIOU = 1 - (IOU - \frac{|\Delta|}{|C|}) \quad (5)$$

The GIOU_Loss function adds a measure of the intersection scale, which is beneficial to solve the problem that the bounding boxes sometimes do not overlap. At the same time, in the process of model training, with the increase of the number of sample training iterations, the learning rate is reduced from the initial 0.001 to 0.0005, which is conducive to the further convergence of the loss function and improves the recognition effect of the model.

IV. Program Design Instructions

4.1. Program code design structure

├── data: It is mainly a configuration file that stores some hyperparameters (these files (yaml files) are used to configure the paths of the training set, test set and validation set, including the number of target detection types and the name of the type); there are also some official test images. If you train your own dataset, you need to modify the yaml file. However, it is not recommended to put your own data set under this path, but it is recommended to put the data set under the same level directory of the YOLOv5 project.

├── models: There are mainly configuration files and functions for network construction, which contain four different versions of the project, namely s, m, l, and x. As can be seen from the name, the size of these versions. Their detection measures are from fast to slow, but the accuracy is from low to high. This is the so-called fish and bear's paw cannot have both. If you train your own data set, you need to modify the corresponding yaml file to train your own model.

├── utils: Stores the functions of the tool class, including the loss function, the metrics function, the plots' function and so on.

├── weights: Place the trained weight parameters.

├── detect.py: Use the trained weight parameters for target detection, which can detect images, videos and cameras.

├── train.py: A function to train your own dataset.

├── test.py: The function to test the training results.

├──requirements.txt: This is a text file that contains some versions of the environment dependency packages that use the YOLOv5 project. You can use this text to import the corresponding version of the package.

4.2. Preparation of pretrained weights

Generally, it is to shorten the training time of the network and achieve better accuracy. The 5.0 version of YOLOv5 provides us with several pre-trained weights, and we can choose different versions of the pre-trained weights according to our different needs. After obtaining the name and size information of the weight, it can be expected that the larger the pre-trained weight, the higher the training accuracy will be, but the slower the detection speed will be.

V. Program Design Instructions

5.1. Experimental dataset

We often obtain some target detection dataset resource labels from the Internet in the format of VOC (xml format), and the file format required for YOLOv5 training is YOLO (txt format), here we need to convert the label file in xml format. txt file. At the same time, when training your own YOLOv5 detection model, the data set needs to be divided into training set and validation set.

The dataset used in this experiment is a small-scale image dataset containing 1200 faces without masks and faces with masks. The annotation information of the dataset includes target type and location information. In order to ensure that the training data is as much as possible and the test set is universal, the training set and the test set are divided according to the ratio of 9:1.

5.2. Model Evaluation Metrics

The model evaluation indicators used in this paper mainly include: precision rate, recall rate, average precision and harmonic mean. The higher the precision rate and the recall rate, the better the recognition effect of mask wearing, but the two are negatively correlated. The mean precision and the harmonic mean are quantitative indicators that consider both the precision rate and the recall rate. The larger their values, the better the recognition effect of mask wearing.

Precision and recall are two metrics widely used in the fields of information retrieval and statistical classification to evaluate the quality of results. The precision is the ratio of the number of relevant documents retrieved to the total number of documents retrieved, which measures the precision rate of the retrieval system; the recall rate refers to the ratio of the number of relevant documents retrieved to the number of all relevant documents in the document library, which measures the is the recall rate of the retrieval system.

If an instance is a positive class, but is predicted to be a positive class, it is a true class (True Positive TP)

If an instance is a negative class, but is predicted to be a negative class, it is a true negative class (True Negative TN)

If an instance is a negative class, but is predicted to be a positive class, it is a false positive class (False Positive FP)

If an instance is a positive class, but is predicted to be a negative class, it is a false negative class (False Negative FN)

Then the accuracy is:

$$precision = \frac{TP}{TP+FP} \tag{6}$$

the recall is:

$$recall = \frac{TP}{TP+FN} \tag{7}$$

Both values are between 0 and 1, and the closer the value is to 1, the higher the precision or recall.

mAP (mean average precision) is an average value, which is often used as a detection accuracy indicator in target detection. The mAP indicator is calculated by detecting different AP (average precision) values

corresponding to multiple targets in a task for an average target. The value of AP is the area of a P-R curve that is accurately drawn by the precision and recall of the experimental results obtained through predictive analysis. Usually in deep learning, the F1 curve comprehensively considers the precision rate and the recall rate, which reflects the quality of the model to a certain extent.
The F1 curve is calculated as follows:

$$F1 = \frac{2*P*R}{P+R} \qquad (8)$$

5.3. Experimental results

After training 100 epochs, the improved YOLOv5 model for mask wearing recognition is evaluated according to the above model evaluation indicators. The evaluation results are shown in Figure 5.

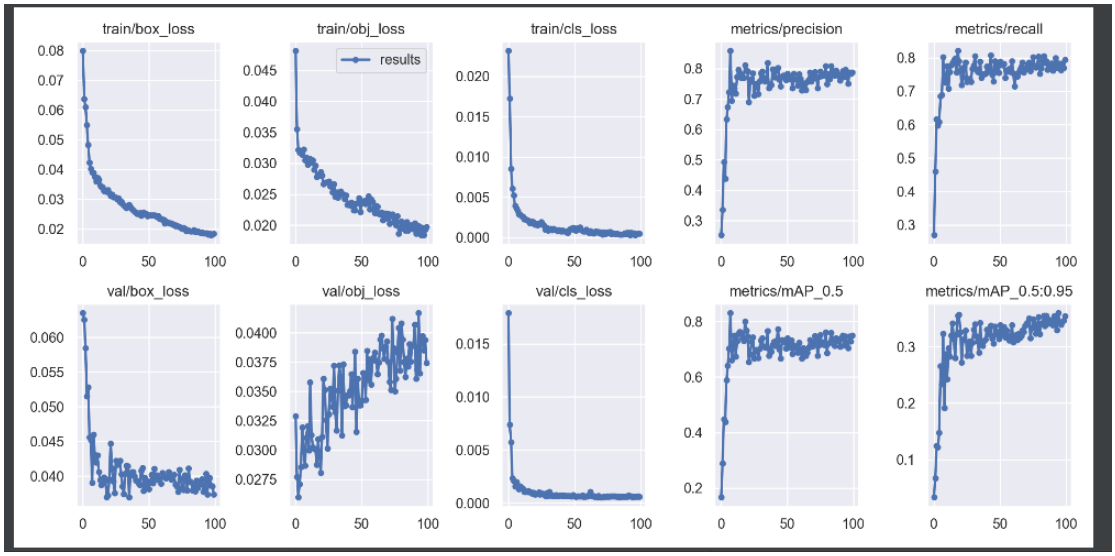

Fig. 5 Evaluation result

Where cls_loss represents the loss of confidence, box loss represents the loss of the predicted frame position, and obj loss represents the loss of the target.
The P, R, P-R, F1 curves for evaluating the model are shown in the following figure:

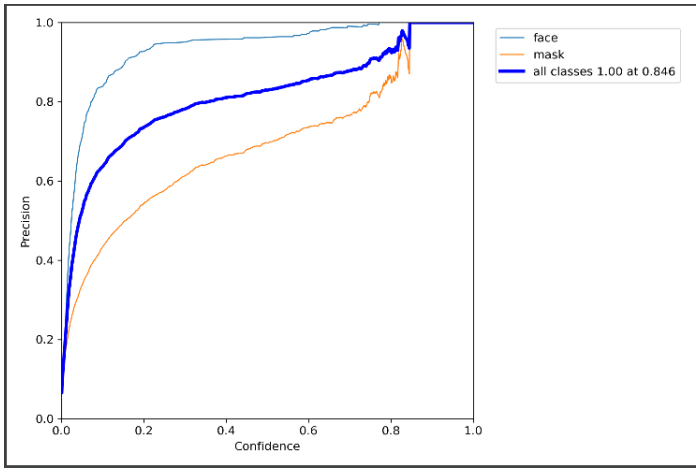

Fig. 6 P curve

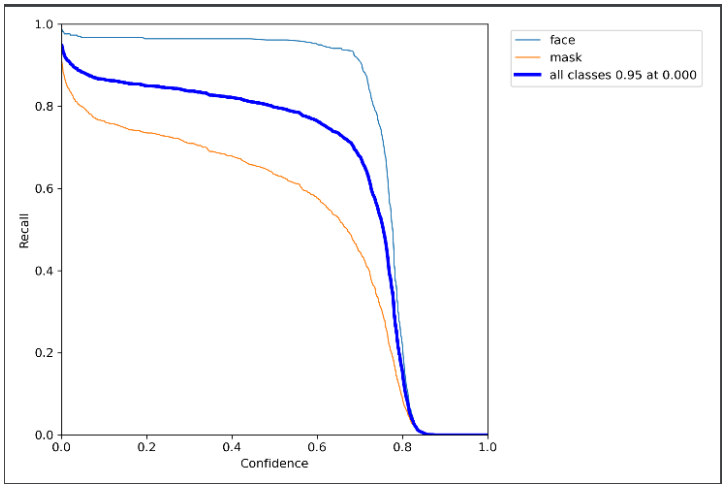

Fig. 7 R curve

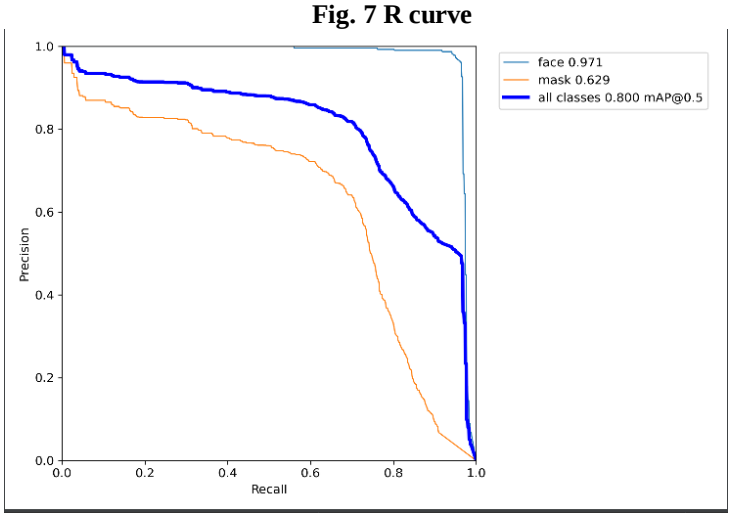

Fig. 8 P-R curve

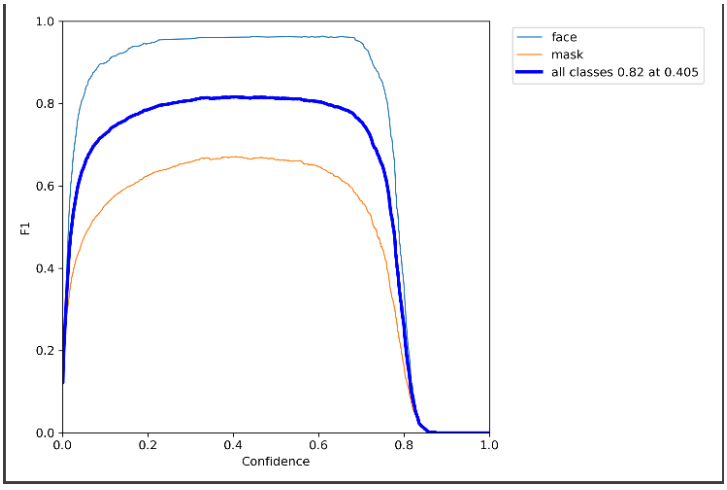

Fig. 9 F1 curve

5.4. Analysis of results

The evaluation results of the improved YOLOv5 model for mask wearing recognition are shown above. After 100 epochs, the model reaches a state of convergence. During the model training process, the improvement of its accuracy and recall is very stable. After the model reaches saturation, the accuracy rate is shown in Figure 6,

which can be kept above 80%; the recall rate is shown in Figure 7, which can be kept stable at around 78%. The average precision and the harmonic mean are also kept at a high level. The average precision is shown in Figure 9, which can be stably maintained at around 80%; the model has a faster recognition speed in practical applications. After testing, the GTX 1650 graphics card was used. , its recognition speed can reach 35FPS, which basically meets the speed requirements of real-time detection.

VI. Results presentation and conclusion

6.1. Results presentation

Fig. 10 Detection of a picture

Fig. 11 Detection of a video

6.2. Conclusion

I have developed a system that can monitor an area with a live camera without any additional equipment. The proposed system is a simple real-time video analyzer. It has the potential to check if people are wearing masks. It can be installed in any supermarket and public places. This helps us defeat the widespread COVID-19 virus. Because wearing a mask reduces community spread of the COVID-19 virus. We can use it for many other options, such as checking and verifying that all customers are wearing masks. The system thoroughly checks people entering through the main entrance. We can process the video recording and see if the person is wearing a mask. If the person is wearing his/her face covering, the door will open; otherwise, it may display some false commands such as "please wear your face mask". The developed model uses YOLOv5 and Pytorch technology to process images and real-time video. From the results, it can be said that the developed model is able to detect whether an individual is wearing a mask. The model learns parameters quickly. It captures video from the camera, processes the video, identifies objects, and finds out whether a person is wearing a mask or not. This system has some limitations. For example, it sometimes accurately detects whether a person is wearing or not wearing a mask only when the person is directly facing the camera. For example, it is quite useful in supermarkets and airports.

REFERENCES

[1]. Wu Zunyou. [2020] "The role of asymptomatic infection of novel coronavirus pneumonia in the spread of the epidemic and prevention and control strategies" Chinese Journal of Epidemiology, Vol. 41, Issue 6: pp.801-805.

[2]. Zhang Hui, Wang Kunfeng, Wang Feiyue. [2017] "The application progress and prospect of deep learning in target visual detection" Chinese Journal of Automation, Vol. 43, Issue 8: pp.1289-1305.

[3]. Feng Guochen, Chen Yanyan, Chen Ning, et al. [2015] "Research on automatic identification technology of helmets based on machine vision" Mechanical Design and Manufacturing Engineering, Vol. 44, Issue 10: pp.39-42.

[4]. Li Meiling, Fu Hui, Wang Xiaojing, et al. [2016] "Road extraction from high-resolution remote sensing images" Remote Sensing Information, Vol. 31, Issue 2: pp.64-68.

[5]. Zhang Xiubao, Lin Ziyuan, Tian Wanxin, et al. [2020] "Face mask recognition technology in all-weather natural scenes" Science in China: Information Science, Vol. 50, Issue 7: pp.1110-1120.

[6]. Redmon J, Divvala S, Girshick R, et al. [2016] "You only look once: Unified, real-time object detection" Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA. pp.779–788.

[7]. Qiao Yanting, Chen Wanpei, Zhang Tao. [2020] "SSD-based lightweight vehicle detection network" Radio Engineering, Vol. 50, Issue 11: pp.926-931.

[8]. Lin Jianwei. [2019] "A review of YOLO image detection technology" Fujian Computer, Vol.35, Issue 9: pp.80-83.

[9]. Lomte V,Shinde A.[2014] "Review of a New Distinguishing Attack Using Block Cipher with a Neural Network[J]" International Journal of Science and Research, Vol.3, Issue 8: pp.733-736.

[10]. S. Ren, K. He, R. Girshick, and J. Sun. [2015] "Faster r-cnn: Towards real-time object detection with region proposal networks" arXiv preprint arXiv:1506.01497.

[11]. I. Krasin, T. Duerig, N. Alldrin, et al. [2017] "Openimages: A public dataset for large-scale multi-label and multi-class image classification" Dataset available from https://github.com/openimages.

[12]. T.-Y. Lin, P. Dollar, R. Girshick, K. He, et al. [2017] "Feature pyramid networks for object detection" In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.2117–2125.

[13]. T.-Y. Lin, M. Maire, S. Belongie, et al. [2014] "Microsoft coco: Common objects in context" In European conference on computer vision, pp.740–755.