2022 International Conference on Machine Learning, Control, and Robotics (MLCR) | 978-1-6654-5459-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/MLCR57210.2022.00019

A Face Mask Detection Algorithm Based on YOLOv5

Jiahui Yin
School of Electrical and Electronic Engineering
University of Manchester
Manchester, United Kingdom
School of Computer Science and Technology
Nanjing Tech University
Nanjing, China
yinjiahui0104@163.com

Jing Jin*
School of Computer Science and Technology
Nanjing Tech University
Nanjing, China
jing_jin@njtech.edu.cn

***Abstract*—As a new machine learning method, deep learning has been widely used in computer vision. YOLOv5, a target detection algorithm based on deep learning, has a good detection effect. In the case of COVID-19, masks should be worn correctly in public places. Therefore, it is urgent to design an accurate and effective face mask detection algorithm. To solve the problem of mask-wearing detection, a face mask detection algorithm based on YOLOv5 is proposed. The main research contents include training of the YOLOv5 model, verification of face mask detection function, and analysis and comparison of detection effects of three different sizes of detection models: YOLOv5s, YOLOv5m and YOLOv5l. The proposed model realizes the mask detection function and obtains the advantages and disadvantages of different scale models through performance evaluation. The maximum mAP of the model reached 88.1%, with good detection accuracy.**

***Index Terms*—Face Detection, Mask Recognition, Deep Learning**

I. Introduction

Under the current epidemic situation, wearing a mask has become an effective way to prevent and cut off the transmission of the epidemic[1]. Therefore, the research on face mask detection algorithms is particularly important.

For the traditional target detection algorithm, the performance is not stable enough. In the process of extracting target features, the detection results are easily affected by the target shape, environment and other factors, and the detection accuracy and speed cannot meet the high requirements[2]. Compared with the traditional target detection algorithm, the algorithm based on deep learning has a significant improvement in performance. Its basic principle is to extract the characteristics of the target by establishing a model, which is less affected by human factors. It improves detection accuracy, reduces detection time, and is more efficient.

Target detection technology based on deep learning can be divided into two categories, two-stage and one-stage[3]. For the two-stage target detection algorithm, the first step is to extract features from the input image, generate the region proposal, and predict the approximate position of the target. In the second stage, the main task is to finish the refinement of the target position and the judgment of the category. In contrast, the one-stage target detection algorithm based on boundary box regression can automatically generate both the classification probability and position coordinates of the detected target.

However, in realistic scenarios, masks have a smaller area and single features compared to large detection targets such as cars and pedestrians, and there are problems such as irregular mask-wearing, occlusion by extraneous factors and unclear features, which increases the difficulty of detection. At the same time, in airports, schools, shopping malls and other crowded and densely populated environments, there is more demand for face mask detection and higher accuracy requirements[4]. According to the above, the accuracy and speed of face mask detection models need to be improved, and the study of face mask detection algorithms based on deep learning has great practical application significance.

II. YOLOv5 Target Detection

A. Introduction to YOLO Family of Models

YOLO (You Only Look Once) algorithm is a target detection method proposed by Joseph Redmon et al. in 2016[5]. YOLO creatively proposes to treat target detection as a regression problem, with only one stage of detection, using a single network to output both target location and category. Redmon improved on the original algorithm with the release of YOLOv2 and YOLOv3, followed by Alexey Bochkovskiy with YOLOv4. With the continuous updating and iteration of the YOLO version, its algorithm becomes faster and more accurate. Glenn Jocher launched YOLOv5 based on YOLOv4 in 2020. YOLOv5 is the latest version of YOLO, which uses the Pytorch framework to replace the Darknet framework of YOLOv4. YOLOv5 adjusts and optimizes the parameters based on retaining the original network part, and the detection speed is improved. The fastest detection speed reaches 140fps[6].

There are four network models in YOLOv5, the smallest of which is YOLOv5s. The YOLOv5s model can exhibit a fast prediction speed, but the prediction accuracy is not as

978-1-6654-5459-9/22/$31.00 ©2022 IEEE
DOI 10.1109/MLCR57210.2022.00019

good as it could be. In addition, YOLOv5m, YOLOv5l and YOLOv5x are three other models that continue to deepen and expand on YOLOv5s, with some improvement in accuracy but slower detection speed. The network architecture of YOLOv5s is shown in Figure 1[7].

The overall structure of the YOLO model is that the input image is transformed by a neural network to produce an output tensor, consisting of 18 convolutional layers, 6 pooling layers and 2 fully connected layers.

B. How YOLOv5 Algorithm Works

The basic idea of YOLO is that a picture is first segmented into several equally sized grids, and if the centre coordinates of a target fall on a particular grid, then the task of identifying that target is given to the corresponding one. The detection process of the algorithm is roughly divided into three steps: first pre-processing, where the image is uniformly resized and segmented, then feature extraction using a convolutional neural network, and finally a non-maximal suppression (NMS) operation. The flow of the YOLO algorithm is shown in Figure 2.

The detailed detection steps of the YOLOv5 algorithm are first, the input image is resized and divided into S×S grids, with each grid being responsible for identifying only the objects in its corresponding region, greatly reducing the probability of repeated recognition and the amount of computation. Each grid predicts the presence or absence of a target inside it, and if a target is present, the grid also predicts the probability of target coordinates, bounding boxes and categories. Each bounding box has five parameters, x, y, w, h, and confidence, where the centre of the predicted box is (x,y), the relative width of the predicted box is w, and the relative height is h. Confidence reflects the accuracy of the predicted target location when the target is contained in the bounding box, and can be calculated as follow:

$$Conf(object) = P_r(object) \cdot IoU_{tru}^{pre} \tag{1}$$

Where, $P_r(object)$indicates whether the grid of the prediction frame contains an object. When the object is included, $P_r(object) = 1$, conversely, $P_r(object) = 0$. The intersection over union (IoU) represents the coincidence ratio between a predicted frame and the actual position of the object, and the higher the value, the greater the coincidence degree. Ideally, the predicted bounding box is the same as the IoU of the real object. The conditional probability value predicted by each grid is $P_r(Class_i|object)$, multiply the classification probability of each border by the confidence level of the border to obtain the confidence score of a certain category:

$$\begin{aligned} Conf &= P_r(Class_i|object) \cdot Conf(object) \\ &= P_r(Class_i) \cdot IoU_{tru}^{pre} \end{aligned} \tag{2}$$

NMS means non-maximal suppression. In the field of target detection, for a single image, the model ends up predicting a number of boxes and the prediction score for that box. When there are too many overlapping boxes (i.e., the IoUs between the boxes are too large), the prediction is optimized using the NMS algorithm, which is based on the idea that the boxes with the largest intersection ratio in the overlap are kept and the boxes with a lower than maximum intersection ratio are removed.

C. Loss Function of YOLOv5 Algorithm

The loss function is used to measure the extent to which the model's predictions differ from the actual results and greatly determines the performance of the model. In the YOLOv5 detection method, the complete loss function consists of three components: the bounding box regression loss $(Loss_{box})$, the confidence prediction loss $(Loss_{obj})$ and the category prediction loss $(Loss_{cls})$. It uses $GloU\_loss$ (Generalized Intersection over Union) to calculate the mean value of the bounding box regression loss function, the smaller the box value, the more accurate the prediction box regression; $BCELogits\_loss$ to calculate the mean value of the confidence prediction loss function, the smaller the objectness value, the more accurate the target detection; using BCE_loss is used to calculate the mean value of the category prediction loss function, and the smaller the value of classification, the more accurate the target classification.

III. Model Training

A. Dataset Construction

The dataset is a key part of the model training process, and the performance of model detection is largely dependent on the quantity and quality of the dataset. The more comprehensive the data set, the closer it is to the real application scenario, and the better the performance of the trained model in practice. The image labelling tool used in this paper is LabelImg, and the labelling interface is shown in Figure 3.

The dataset images for this experiment consist of 1200 images from the training set and 400 images from the verification set. The images contain two target categories, faces wearing masks and faces without masks, and the image annotation process is as follows.

- Open the image set path. Find the folder where the image set is located and import the images into LabelImg.
- Label the image. Pull out the rectangular box and completely enclose the face within the box and set the category label. Images with masks are labelled as ”mask”; images without masks are labelled as ”face”.
- Save the data set. Click on the ”Save” button to save the data, and then click on ”Next Image” to label the next image. Once all images have been annotated and saved, the dataset is complete.

YOLO labels are stored in the format (class, x, y, w, h) in a text document. Class ”0” is with a mask (mask) and class ”1” is without a mask (face), (x, y) represents the centre coordinates, w represents the width and h represents the height, all normalized data. Ultimately, the information for each image will be saved in a separate text document.

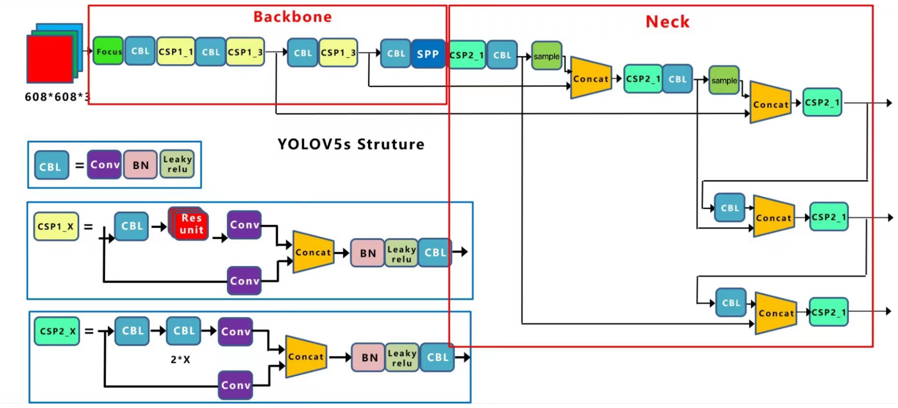


Fig. 1. YOLOv5s network architecture.

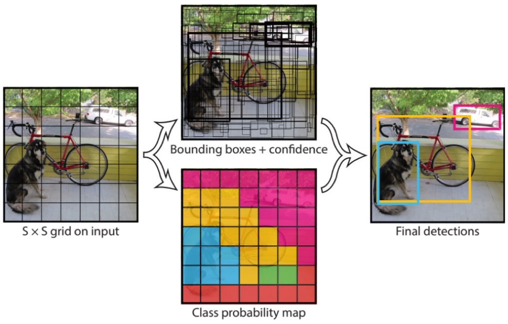
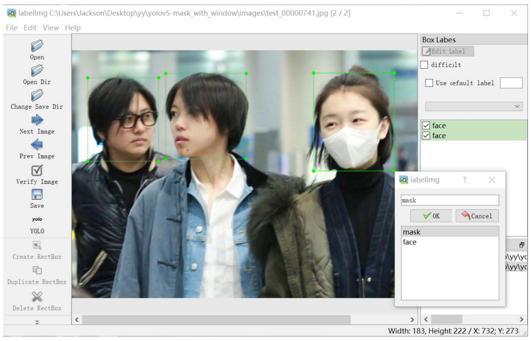


Fig. 2. YOLO algorithm process.


labelImg C:\Users\Jackson\Desktop\yy\yolov5-mask_with_window\images\test_00000741.jpg [2 / 2]

Width: 183, Height: 222 / X: 732; Y: 273

Fig. 3. LabelImg interface example.

B. Training Environment

To ensure the same training environment for each model and reliable experimental results, all model training in this paper was run in the environment configuration in TABLE I.

TABLE I
ENVIRONMENT CONFIGURATION

Name	Configuration
Operating System	Windows 10
GPU	NVIDIA GeForce GTX 1080 Ti
CUDA	10.2
cuDNN	7.6.5
Python	3.8.5
Pytorch	1.8.0

C. Training Process

First set the model configuration file. Take the small model yolov5s.yaml as an example, the batch size is set to 4, which means 4 images are processed at the same time each time; the number of training rounds is set to 100, which means 100 iterations; the number of categories is set to 2, and the two target categories are mask and face. Then set the configuration file mask_data.yaml for the dataset and set the paths for the training images and the validation images respectively. The parameters of the training model in this paper are set as shown in TABLE II.

TABLE II
MODEL TRAINING PARAMETERS SETTING

Parameter	Configuration
network model	yolov5s.yaml
dataset	mask_data.yaml
pretraining model	yolov5s.pt
epochs	100
batch_size	4
image_size	640

After configuration, train.py was run to start the training, which took about 3 hours in total. After training, two weight files were obtained: best.pt and last.pt. best.pt was selected as the model training result and used for subsequent performance analysis. In addition, the loss function used to record the training process, as well as the data and images used for the performance analysis, are also output after training.

IV. Experimental Results

A. Performance Evaluation Metrics

In the field of target detection, the following metrics are commonly evaluated: Precision (P), Recall (R), Precision-Recall (P-R) curve, F-measure (F), Average Precision (AP) for a single class, mean Average Precision (mAP) for multiple classes and Frames per second (FPS) for detection speed.

In a binary classification problem, samples can be classified into four categories according to the combination of the actual and predicted categories. The confusion matrix for the classification of samples is shown in TABLE III.

TABLE III
Confusion Matrix

	Positive sample	Negative sample
Predicted to be positive	True Positive(TP)	False Positive(FP)
Predicted to be negative	False Negative(FN)	True Negative(TN)

Precision represents the proportion of all results with a positive prediction that are also positive. Precision is useful for highlighting the relevance of results and assessing the accuracy of predictions, and can be calculated as follow:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

Recall represents the percentage of all positive samples that are successfully predicted to be positive. It can be used to assess the comprehensiveness of the prediction, and can be calculated as follow:

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

F-measure means the harmonic mean of P and R, and can be calculated as follow:

$$F = \frac{1}{\frac{\lambda}{P} + \frac{1-\lambda}{R}} \tag{5}$$

When $\lambda = 0.5$, it simplifies to:

$$F1 = \frac{2P \cdot R}{P + R} \tag{6}$$

The P-R curve provides a visual representation of the relationship between precision and recall of the model in the overall sample, and we can compare the P-R curves to determine the different model performances. The two evaluation metrics, precision and recall, are in conflict with each other. If the recall rate increases while the precision remains flat or increases, the model's performance improves.

AP represents the area of the graph enclosed by the P-R curve and the coordinate axes for a single category. In general, the size of the area is proportional to the performance of the classifier. mAP is the average of the multi-category AP values, and the higher the value of mAP, the better the performance of the model.

FPS is the number of images that can be detected per second, which is an important metric for evaluating speed. A higher value means a faster detection rate.

The performance evaluation metrics chosen for this paper are shown in TABLE IV, where mAP@0.5 represents the mAP for an IoU threshold of 0.5.

TABLE IV
Performance Evaluation Metrics

Metrics	Explanation
P	Precision
R	Recall
F1	harmonic averaging of P and R
P-R	balance curve of P and R
AP	average precision for individual categories
mAP@0.5	average precision for all categories for IoU threshold 0.5
FPS	number of images detected per second

B. Performance Analysis of Face Mask Detection

The precision and recall of the model gradually increased and stabilized with time and the number of training sessions. The changes in precision and recall during training are shown in Figures 4 and 5 respectively.

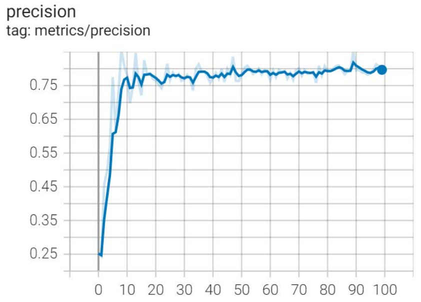


Fig. 4. Precision-Training count function.

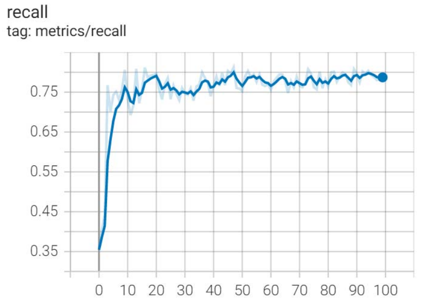


Fig. 5. Recall-Training count function.

Meanwhile, the loss functions are shown in Figures 6, 7 and 8. Figure 6 shows the bounding box regression loss function, Figure 7 shows the confidence prediction loss function and Figure 8 shows the classification prediction loss function. It is clear that all of the 3 losses decrease and stabilize with slight fluctuations as the number of training sessions increases.

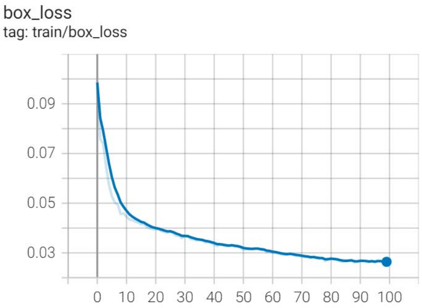


Fig. 6. Bounding box regression loss.

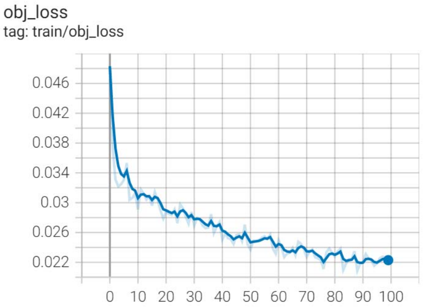


Fig. 7. Confidence prediction loss.

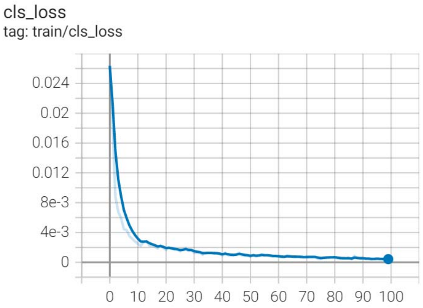


Fig. 8. Category prediction loss.

Through training, the detection effect of this model is shown in Figure 9. As can be seen in the figure, the detection system identifies five faces wearing masks and one face without a mask, which are boxed out with red and pink rectangles respectively. The category face or mask is labelled above the rectangles and the data to the right of the category indicates the confidence level. For example, ”mask 0.89” means that the confidence level that this is a face with a mask is 0.89; ”face 0.90” means that the confidence level that this is a face without a mask is 0.90.

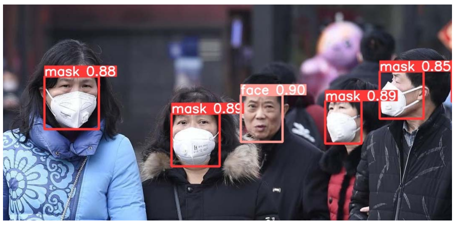


Fig. 9. Example of detection effect.

After training, the confusion matrix of the YOLOv5s model was obtained as shown in Figure 10. In this image, all data are normalized according to the column orientation.

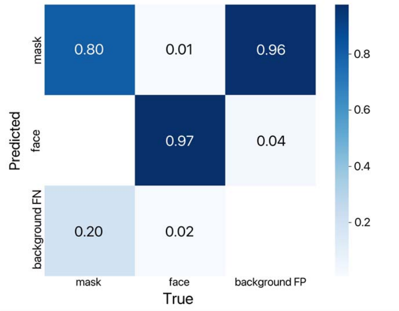


Fig. 10. Yolov5s model confusion matrix.

According to the formulae for precision and recall, the data in the diagonal cells of the confusion matrix is R, while P can be calculated horizontally. From the graph it follows that:

$$P_{face} = \frac{TP}{TP+FP} = \frac{0.97}{0.97+0.04} = 0.960$$

$$P_{mask} = \frac{TP}{TP+FP} = \frac{0.80}{0.80+0.01+0.96} = 0.452$$

$$R_{face} = \frac{TP}{TP+FN} = 0.970$$

$$R_{mask} = \frac{TP}{TP+FN} = 0.800$$

Thus, after calculation it follows that:

$$P = \frac{0.960+0.452}{2} = 0.706$$

$$R = \frac{0.970+0.800}{2} = 0.885$$

$$F1 = \frac{2P \cdot R}{P+R} = 0.785$$

The functions of Precision, Recall and Confidence are shown in Figures 11 and 12 respectively. From the figure, it can be obtained that as the confidence level increases, the precision gradually increases and the recall gradually decreases, and the precision and recall are negatively correlated.

F1 and P-R curves are shown in Figures 13 and 14 respectively. Both metrics reflect the balance between precision

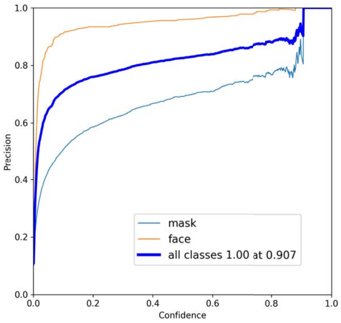


Fig. 11. Precision-Confidence function.

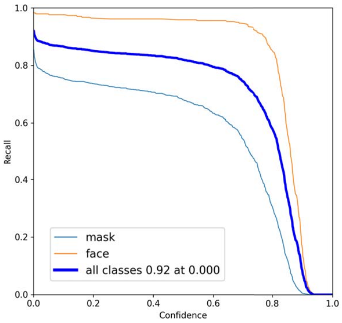


Fig. 12. Recall-Confidence function.

and recall and are important functions for measuring the performance of the target detection model.

From the F1-Confidence function, the F1 reaches a maximum of 0.82 at a confidence level of 0.476, which indicates a certain balance between precision and recall and maximum performance. From the P-R curve, it is obtained that the model has a detection accuracy of 0.976 for the category of unmasked faces and 0.606 for the category of masked faces. At an IoU threshold of 0.5, mAP@0.5 is 0.791.

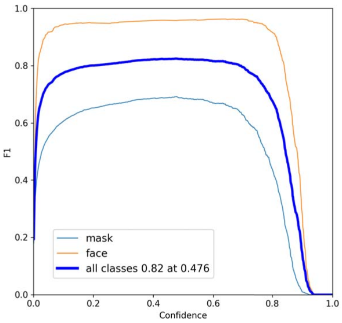


Fig. 13. F1-Confidence function.

In addition, a speed test yielded that the model could achieve a detection speed of 0.006 seconds per image, i.e., the number of images detected per second was approximately 167.

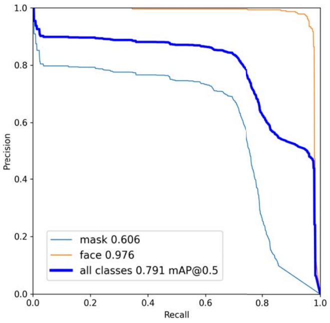


Fig. 14. P-R function.

Similarly, two other models of YOLOv5, YOLOv5m and YOLOv5l, were trained with the same parameter configuration and the performance evaluation data obtained are shown in TABLE V.

TABLE V
YOLOv5s, YOLOv5m, YOLOv5l Performance Comparison

	YOLOv5s	YOLOv5m	YOLOv5l
P	0.706	0.705	0.691
R	0.885	0.890	0.910
F1	0.785	0.787	786
AP(face)	0.976	0.962	944
AP(mask)	0.606	0.746	818
mAP@0.5	0.791	0.854	881
FPS	167	91	53

From the data in the table, the performance of the models varies for all different sizes. As the model increases, the average accuracy mAP for all categories at an IoU threshold of 0.5 gradually increases, indicating that the larger the model size, the higher the accuracy of detection. Conversely, the number of images detected per second FPS gradually decreases as the model increases, indicating that the smaller the model, the faster the detection.

To verify the detection effectiveness of the three trained models, YOLOv5s, YOLOv5m and YOLOv5l, the same image containing dense faces was tested using each of the three models. The original image is shown in Figure 15, which contains 11 faces wearing masks and 2 faces not wearing masks, with 5 of the faces being more heavily obscured. Three models were used to detect face mask wear on this image, and the detection results are shown in Figures 16, 17 and 18 respectively. Among them, Figure 16 shows the detection results using the YOLOv5s model, Figure 17 shows the detection results using the YOLOv5m model and Figure 18 shows the detection results using the YOLOv5l model.

The YOLOv5s model detected 6 faces with masks and 1 face without a mask in 0.009 seconds, the YOLOv5m model detected 7 faces with masks and 1 face without a mask in 0.012 seconds, and the YOLOv5l model detected 7 faces with masks and 2 faces without masks in 0.019 seconds. In conclusion, the detection accuracy increases, and the detection speed decreases with the increased size of the model. However, all 3 models did not fully detect all faces due to severe occlusion of some faces. For a face wearing a black mask in the centre, all 3 models failed to detect the target, and the detection performance needs to be further optimized.

Fig. 15. Original image.

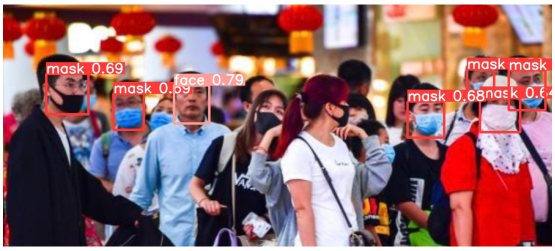


Fig. 16. Detection results using YOLOv5s model.

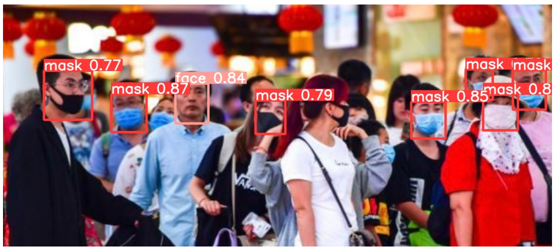


Fig. 17. Detection results using YOLOv5m model.

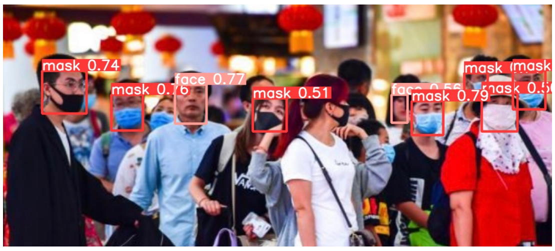


Fig. 18. Detection results using YOLOv5l model.

V. Conclusion

This paper provides a detailed introduction to the study of the face mask detection algorithm based on YOLOv5. Firstly, the YOLOv5 algorithm is described and illustrated, including the algorithm principle and loss function. Then the construction method of the face mask dataset and the parameters of the experimental platform are explained, and the three models YOLOv5s, YOLOv5m and YOLOv5l are trained one by one. After the training is completed, the detection effects of each model are analyzed, and conclusions are drawn based on the performance evaluation metrics selected in this paper. From the experiments, it can be obtained that the detection accuracy increases, and the detection speed decreases as the model increases.

VI. Acknowledgments

This work was supported by Natural Science Foundation of Jiangsu Higher Education Institutions of China (21KJB20008).

References

[1] S. E. Eikenberry, M. Mancuso, E. Iboi et al., "To mask or not to mask: Modeling the potential for face mask use by the general public to curtail the COVID-19 pandemic," Infectious Disease Modelling, vol. 5, 2020, pp. 293–308.

[2] N. K. Ratha, V. M. Patel, R. Chellappa, Deep Learning-Based Face Analytics. Cham, CA: Springer, 2021.

[3] Zhao, Zhong-Qiu et al., "Object Detection With Deep Learning: A Review," IEEE transaction on neural networks and learning systems, vol. 30.11, 2019, pp. 3212–3232.

[4] Zhao, Yuanzhang, and Shengling Geng, "Face Occlusion Detection Algorithm Based on Yolov5," Journal of Physics: Conference Series, vol. 2031.1, 2021, pp. 12053.

[5] Redmon, Joseph et al., "You Only Look Once: Unified, Real-Time Object Detection," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[6] Chen, Weijun et al., "YOLO-Face: a Real-Time Face Detector," The Visual computer, vol. 37.4, 2021, pp. 805–813.

[7] S. Tan, X. Bie, G. Lu and X. Tan, "Real-time detection for mask-wearing of personnel based on YOLOv5 network model," J. Laser J., vol. 42.2, 2021, pp. 147–150.