Face occlusion detection algorithm based on yolov5

Yuanzhang Zhao\textsuperscript{1,3}, Shengling Geng\textsuperscript{1,2,3}\textsuperscript{*}

\textsuperscript{1}School of Computer Science, Qinghai Normal University, Xining, Qinghai, 810016, China
\textsuperscript{2}Academy of Plateau Science and Sustainability, Qinghai Normal University, Xining, Qinghai, 810016, China
\textsuperscript{3}A State Key Laboratory of Intelligent Information Processing and Application of Tibetan Language jointly established by the Ministry and the Province, Qinghai Normal University, Xining, Qinghai, 810016, China

\textsuperscript{*} Geng_sl@126.com

Abstract. The current face-mask recognition detection algorithm during the epidemic only distinguishes between wearing or not wearing a mask. Such detection often has certain loopholes, such as using other objects to cover their mouths and noses instead of masks to cheat the detection. To address such problems, this paper proposes a YOLOv5 based face occlusion detection algorithm, which is modified based on the YOLOv5 algorithm by improving the loss function as DIoU and increasing the experimental samples by introducing multiple data sets to improve the object detection effect. The experimental results show that the improved YOLOv5 algorithm has improved the object detection effect for different kinds of face occlusions, which verifies the method's effectiveness.

1. Introduction

Currently, China is in a new phase of normalization of epidemic prevention and control, and the main task is to prevent cases from being imported from abroad. Wearing masks correctly is an effective way to stop the spread of the virus. So many mask identification systems have been installed in airports, train stations, and other significant public places to detect whether people are wearing masks. However, when people cover their mouths and noses with scarves, collars, hands, etc. to pass through the detection devices, they can often fool the machines, thus creating a significant safety hazard and risk of outbreak transmission. Nowadays, deep learning\cite{1}-based object detection\cite{2} techniques are widely used in various fields, and with the increasing maturity of deep learning, many research scholars have accumulated many scientific achievements. This has led to significant breakthroughs in the field of computer vision\cite{3}.

The algorithms represented by CNN\cite{4} in object recognition have improved tremendously in terms of detection speed and accuracy. At present, the mainstream target detection algorithms are mainly divided into two categories, and one is the two-stage algorithm based on detection frame and classifier, such as RCNN\cite{5}, Fast RCNN\cite{6}, and Faster RCNN with RPN\cite{7}, their main problem is the complex network structure and slow detection speed. Another kind of the one-stage algorithm based on regression, such as SSD\cite{8}, YOLO\cite{9}, YOLOv2\cite{10}, YOLOv3\cite{11}, and YOLOv4\cite{12}, has more practicality because of their accurate localization and faster detection speed. The algorithm used in this paper is the improved YOLOv5 algorithm proposed on top of these algorithms.
This paper addresses the shortcomings of existing face mask detection devices. Existing detection algorithms only distinguish between wearing a mask and not wearing a mask. Therefore, the detection proposed in this paper needs to identify the face target and make an accurate distinction between cases where the target is wearing a mask correctly and incorrectly, as well as precisely what is being worn while wearing a non-mask.

2. YOLOv5 algorithm

YOLOv5 is a single-stage target detection algorithm, which adds the focus structure to YOLOv4 and constructs two CSP structures. YOLOv5 network contains four generic modules, as in figure 1, and six basic components, as in figure 2.

Figure 1. YOLOv5's four generic modules.

Figure 2. YOLOv5's six basic components.

Among them, four general modules specifically include (1) Input side: input image, the size of the input image is 608*608, and there is an image pre-processing stage in this phase. As shown in figure 1a. (2) Benchmark network: This module is used to extract some generic feature representations, and it is a classifier network with excellent performance. As shown in figure 1b. (3) Neck network: the neck network is located in the middle of the benchmark network and the head network, and it can be used to improve further the diversity of features and robustness. As shown in figure 1c. (4) Head output: the head is used to complete the output of the target detection results. As shown in figure 1c.

The six basic components include (1) CBL: CBL module consists of conv+bn+leaky_relu activation function, as in figure 2a. (2) Res unit: borrowed from the residual structure in the Res network, used to build a deep network, CBM is a sub-module in the residual module, as in figure 2b. (3) CSP1_x: borrowed from CSP network structure, this module consists of CBL module, Res unit module, and convolutional layer, concate, as in figure 2c. (4) CSP2_x: borrowed from CSP network structure, this module consists of convolutional layer and x res unit modules and concate, as in figure 2d. (5) Focus: This structure first connects multiple slice results and then sends them to the CBL
module, as in figure 2e. (6) SPP[13]: the maximum pooling of 1*1, 5*5, 9*9 and 13*13 is used for multi-scale feature fusion, as in figure 2f.

3. Algorithm improvement

3.1. Data processing improvement

Obtaining more datasets and performing appropriate data pre-processing to meet the desired training requirements, this paper performs two parts: data augmentation and smoothing label processing.

The main purpose of data augmentation is to increase the amount of base data for training to improve the generalization ability of the model and to increase the noise to improve the robustness of the model in two aspects. In this paper, we combine the open-source dataset of Wuhan University and the MAFA dataset of Shi-Ming Ge from the Institute of Information Engineering, Chinese Academy of Sciences, and then obtain a part of the dataset through the network, combine these datasets, and expand the original dataset through operations such as data flipping, image scaling, enhancing contrast, saturation, and brightness, which effectively improve the detection accuracy.

The label smoothing method randomly increases the error annotations in training set during the experiments in this paper. It is made to have a negative learning rate during the training process, thus promoting the model to approach the correct result. Label smoothing is a regularization process that reduces overfitting training. It makes the probability distribution predicted by the model for the test set close to the actual distribution, improving the classifier performance[14].

3.2. Loss function improvement

The GIoU loss function is used in the original YOLOv5 model. GIoU is proposed to alleviate the gradient problem when the IoU loss function does not overlap the detection frame, and the penalty term is added to the original IoU.

\[
L_{\text{GIoU}} = 1 - \text{IoU}(A, B) + \frac{|C - A \cup B|}{|C|}
\]  \hspace{1cm} (1)

Where \( A \) is the prediction frame, \( B \) is the real frame, and \( C \) is the minimum enclosing frame of \( AB \). Here GIoU first tries to increase the size of the prediction frame so that it can overlap with the real frame and then performs IoU computation, inventing that this will consume a lot of time in trying to contact the prediction frame with the real frame, affecting the convergence speed. To solve this problem and speed up the convergence, the DIoU loss function is introduced.

\[
L_{\text{DIoU}} = 1 - \text{IoU}(A, B) + \frac{\rho^2(A_1, B_1)}{C^2}
\]  \hspace{1cm} (2)

Where \( A \) is the prediction frame, \( B \) is the real frame, \( A_1 \) is the center point coordinate of the prediction frame, \( B_1 \) is the center point coordinate of the real frame, \( \rho \) is the Euclidean distance, and \( c \) is the real diagonal length of the ab minimum enclosing frame. Since DIoU loss can directly minimize the distance between two target boxes, the convergence speed is faster. In addition, DIoU can also replace the common IoU evaluation strategy and be applied to NMS, making the results obtained by NMS more reasonable and efficient.

4. Dataset

The production of the dataset in this paper mainly includes three steps: data collection and organization, data pre-processing, and dataset annotation. According to the requirements of the YOLO algorithm, the data used in this experiment are processed in the standard format of the PASCAL VOC dataset[15] and manually labeled with Labeling labeling software to generate XML files for training.

4.1. Data collection and organization

The data required in this paper are mainly from the open-source dataset of Wuhan University, the MAFA dataset of the Institute of Information Engineering of the Chinese Academy of Sciences, and
the images collected from web resources. They are divided into six categories, which are: right (correctly wearing mask), false: nothing (not wearing mask), false: collar (covered by the collar), false: hand (covered by hand), false: scarf (covered by scarf), false: other (other covering), and some images are shown in figure 3.

![Figure 3. Part of the classification picture.](image1)

![Figure 4. The labeling example.](image2)

4.2. Data pre-processing
To ensure more accurate and fast training, data pre-processing is required. Here the data is expanded with operations such as horizontal flipping, image scaling, brightness boosting, saturation boosting, and contrast boosting on the experimental data. Label smoothing is used to increase the mislabeling in the dataset and promote the model to move closer to the correct result.

4.3. Annotation of the dataset
From the integrated 20,000 images, 3,120 images that meet the requirements are selected for labeling, and the Labeling tool is used to label and classify them into six categories. The delineated training set is 2082 images and the test set is 1038 images. The labeling example is shown in figure 4.

5. Experimental setup

5.1. Experimental environment
The experimental environment of this paper is CPU Intel(r) Core(TM) i9-10850k@3.60ghz, GPU NVIDIA GeForce RTX 2080ti 11GB, RAM 64GB, image processing tool is matplotlib, OPEN CV, tensorboard. The overall framework of PyTorch. As table 1.

| Platform | Name                                      |
|----------|-------------------------------------------|
| CPU      | Intel(r) Core(TM) i9-10850k@3.60ghz       |
| GPU      | NVIDIA GeForce RTX 2080ti 11GB            |
| Tool     | matplotlib, OPEN CV, tensorboard         |
| Frame    | PyTorch                                   |

5.2. Experimental parameters
Before model training, the basic parameters are set as follows: momentum is set to 0.937, weight decay coefficient is set to 0.0005, and the initial learning rate is 0.001, as shown in table 2.

| Parameters | Parameters Setting |
|------------|--------------------|
| Learning Rate | 0.001              |
| Decay      | 0.0005             |
| Momentum   | 0.937              |
6. Experimental results and analysis

6.1. Evaluation metrics
In this paper, the evaluation metrics commonly used in target detection tasks are used to test the overall performance: 1. Recall, the probability that the correct category in the sample is correctly predicted. In Equation (3), \( TP \) denotes the number of correct categories predicted correctly, and \( FN \) denotes the number of correct categories predicted as negative categories. 2. Precision, the percentage of the number of correctly predicted samples in the actual model in the prediction dataset, and \( FP \) denotes the number of correct categories predicted to be correct. As in Equation 4

\[
\text{recall} = \frac{TP}{TP + FN} \quad (3)
\]

\[
\text{precision} = \frac{TP}{TP + FP} \quad (4)
\]

The number of convergence curves of the overall loss function during the training of the improved YOLOv5 network is shown in figure 5, and the accuracy curve is shown in figure 6. It can be seen that the loss value tends to stabilize at about 0.03 when the iterations reach 100, and the mAP accuracy stabilizes at about 0.72. From this analysis, it can be concluded that the network works well in the training phase.

6.2. Results and Analysis
After training, the obtained model was tested on the test set using the accepted model. Some of the detection results are shown in figure 7, from which it can be seen that the overall localization is accurate and the recognition effect is ideal. After training and testing, the average detection accuracy for each category was derived, as shown in table 3.

![Figure 5. Loss function.](image)

![Figure 6. Overall precision curves.](image)

![Figure 7. Some of the detection results.](image)

| Category | Right | False: nothing | False: collar | False: hand | False: scarf | False: other | All |
|----------|-------|----------------|--------------|-------------|--------------|--------------|-----|
| Original | 60.3% | 78.1%          | 50%          | 79.4%       | 47.3%        | 69.1%        | 64.1%|
| Improved | 64.3% | 84.1%          | 59.4%        | 84.4%       | 56.3%        | 73.1%        | 70.3%|
7. Conclusion

The improved model has improved the average recognition rate for most categories of images, the average accuracy rate has been enhanced by about 6%, the average recall rate has been enhanced by 2%, and the average training loss value has been reduced by about 0.01. It can be seen that the improved model can effectively improve the detection performance. This provides some technical support to enhance the performance of the existing mask recognition system. Continuing to strengthen the detection of other categories of masks is the focus of future work.

Acknowledgement

This work was supported partially by the National Science Foundation of China (Grant No. 61862055), the Major research and Development Project of China (Grant No. 2020 YFC 1523305), and the Major Research and Development Project of Qinghai (Grant No. 2019-GX-162).

References

[1] Hao X, Zhang G, Ma S. Deep Learning[J]. International Journal of Semantic Computing, 2016, 10(03):417-439.

[2] GIRSHICK R, DONAHUE J, DARRELL T. et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014:580-587.

[3] Heng W, Y Pan et al. "Computer vision research based on Open CV". Proceedings of 2013 International Conference on Communication Technology (ICCT 2013 VII). Ed.. WIT PRESS, 2013, 431-438.

[4] A. S. Razavian, H. Azizpour, J. Sullivan and S. Carlsson, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition," 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 512-519.

[5] Chen C, Liu M Y, Tuzel O, et al. R-CNN for Small Object Detection[J]. Springer, Cham, 2016:144.

[6] Girshick R. Fast R-CNN[J]. Computer Science, 2015:1440-1448.

[7] A. Fattal, M. Karg, C. Scharfenberger and J. Adamy, "Saliency-guided region proposal network for CNN based object detection," 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 2017, pp. 1-8.

[8] Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector[J]. Springer, Cham, 2016:21-37.

[9] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]// Computer Vision & Pattern Recognition. IEEE, 2016:779-788.

[10] Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger[J]. IEEE Conference on Computer Vision & Pattern Recognition, 2017:6517-6525.

[11] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement[J]. arXiv e-prints, 2018.

[12] an Li, Lv Xinyue, Lian Xiaofeng, Wang Ge. YOLOv4_Drone: UAV image target detection based on an improved YOLOv4 algorithm[J]. Computers and Electrical Engineering, 2021, 93(5).

[13] He K, Zhang X, Ren S, et al. Spatial Pyramid Pooling in Deep Convolutional Net-works for Visual Recognition[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 37(9):1904-16.

[14] Malta Ana, Mendes Mateus, Farinha Torres. Augmented Reality Maintenance Assistant Using YOLOv5[J]. Applied Sciences,2021,11(11):4758.

[15] Zhang Xiaoping, Ji Jiahui, Wang Li, He Zhonghe, Liu Shida. People’s Fast Moving Detection Method in Buses Based on YOLOv5[J]. International Journal of Sensors and Sensor Networks,2021,9(1):30-37.