Research and Application of Face Mask Detection Algorithm Based on YOLOV4-Tiny

Ruisha Zhu*, Linjun Zhao

School of Computer Science and Information Engineering, Hubei University, Wuhan, Hubei, 430062, China
**Corresponding author*

Abstract: *This paper mainly studies the target detection of face masks. Based on yOLOV4-TINY algorithm, multi-scale feature fusion of FPN structure and feature enhancement network are used to optimize the algorithm structure. First of all, this paper uses the xml.etree.ElementTree (ET) module in python to read the xml file to obtain the key information such as < object > face parameters, < name > face mask wearing status, < bndbox > face bounding box diagonal two vertex coordinates and so on. According to the vertex coordinates (bndbox) in the xml file, the target box is marked on the original image, and the color state of the target box is distinguished according to the wearing state of < name > face mask. Finally, two pictures of "250.png" and "477.png" are selected to be displayed in the body of this text. Then, a face mask detection algorithm based on YOLOv4-Tiny is established. when training the model, the multi-task joint loss of mask detection task is optimized by combining CIOU loss function and label smoothing strategy, and the Mosaic data enhancement method and learning rate cosine annealing attenuation strategy are used to improve the convergence speed and robustness of the model. After many times of detection and verification, the experimental results show that the detection accuracy of the proposed algorithm model is 94.84% (with_mask), 96.04% (without_mask) and 94.86% (mask_weared_incorrect) respectively in the three categories of human face mask detection: with_mask, without_mask and mask_weared_incorrect. Its quantitative indicator < number of faces correctly classified > / < number of all faces included in the tag file >, mAP is as high as 95.25%, and the accuracy is greatly improved.*

Keywords: *Yolov4-Tiny, Face Mask Detection Algorithm, Iou*

1. Introduction

With the outbreak and spread of COVID-19 across the globe, the epidemic has posed a major threat to human life, health and safety, and exerted a significant impact on global economic development. Although the domestic epidemic has been basically effectively contained, the epidemic prevention and daily monitoring throughout the country remain unslackened. Wearing masks is the most convenient and effective prevention and control measure and method to prevent the spread of COVID-19 in our daily life and travel. For the problem of wearing masks, automatic face mask recognition can effectively detect the wearing of masks among people or crowds. This technology has become an important technical means to restrain the rapid spread of disease in public places with large human flow and protect health.

In the face mask automatic recognition technology, it can quickly and effectively detect the situation of people wearing masks in the picture.As shown in Figure 1-2, facial mask wearing automatic recognition technology can detect the facial features of each person. Each box (bounding box) can select a face, and different box colors are used to distinguish whether a mask is worn or not.

2. YOLOv4-Tiny structures

2.1. Backbone feature extraction network Backbone

Yolov4-tiny [1] is a lightweight version of YOLOv4 with simple network structure and effective balance of accuracy and speed. It uses CSPdarknet53_tiny as the backbone feature extraction network. Compared to CSPdarknet53, the activation function has been remodified to LeakyReLU for faster use. CSPdarknet53_tiny has two features [2]:

(1) CSPnet structure

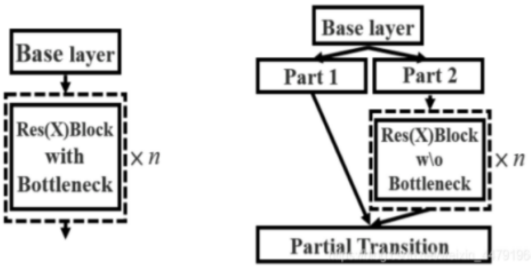

Figure 1: CSPnet structure

(2) Divide the channel

In the main part of CSPnet, CSPdarknet53_tiny will divide the feature layer after 3*3 convolution into two parts, and take the second part. Split in TensorFlow using tf.split. The main feature extraction network is used to obtain the effective feature layers of two shapes, that is, the last two shapes of CSPdarknet53_tiny, which are passed into the enhanced feature extraction network to construct FPN.

2.2. Feature pyramid

The FPN structure is used in YOLOV4-TINY, mainly for feature fusion of the two effective feature layers obtained in the first step [3].

2.3. YOLOHead uses these features to make predictions

In the feature utilization part, yolov4-tiny extracts multiple feature layers for target detection, two feature layers are extracted in total, and the shape of the two feature layers are (26, 26, 128) and (13, 13, 512) respectively. Shape of output layer is (13, 13, 75), (26, 26, 75).

3. Training model of YOLOV4-Tiny

3.1. Mosaic data enhancement

The Mosaic data enhancement of OLOv4 referred to CutMix data enhancement method [4], which has certain theoretical similarity. CutMix data enhancement uses two images for stitching. But Mosaic uses four images, which have the advantage of enriching the background of the objects it detects and directly calculating the data from the four images when it calculates BN.

3.2. Label Smoothing

The idea of label smoothing is as follows:

$$new\_onehot\_labels = onehot\_labels * \left(1 - label\_smoothing\right) + label\_smoothing \;/\; num\_classes \tag{1}$$

When a Label_smoothing value is 0.01, the formula becomes something like this:

$$new\_onehot\_labels = y * \left(1 - 0.01\right) + 0.01 \;/\; num\_classes \tag{2}$$

3.3. CIOU

CIOU takes into account the distance, overlap rate, scale and penalty term between target and anchor, making target box regression more stable and avoiding divergence in the training process like IoU and GIoU. The penalty factor takes into account the length-width ratio of the prediction frame and the length-width ratio of the target frame.

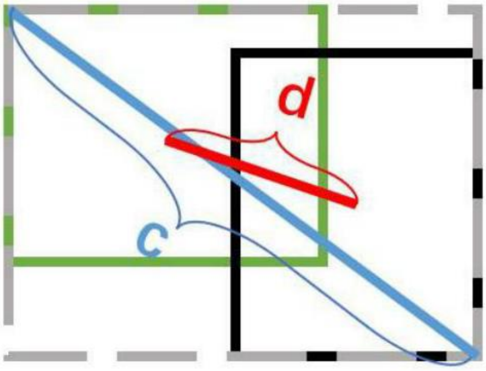

Figure 2: Principle of CIOU algorithm

$$CIOU = IoU - \frac{\rho^2(b,b^{gt})}{c^2} \alpha v \tag{3}$$

$\rho^2(b,b^{gt})$ respectively represents the Euclidean distance of the center point of the prediction box and the real box. C represents the diagonal distance of the smallest closure region that can contain both the prediction box and the real box.

$$\alpha = \frac{v}{1-IoU} + v \tag{4}$$

$$v = \frac{4}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})^2 \tag{5}$$

$$LOSS_{CIOU} = 1 - IoU + \frac{\rho^2(b,b^{gt})}{c^2} + \alpha v \tag{6}$$

4. Loss component

4.1. Calculate parameters required for Loss

Y_pre is the output of an image through the network, which contains two feature layers. It needs to be decoded to be able to draw on a picture

Y_true is the offset position, length, width and type on the grid of (13, 13), (26, 26) corresponding to each real box in a real image. It still needs to be coded to be consistent with y_pred's structure.

Actually y_pre and y_true are both shapes

(batch_size, 13, 13, 3, 85), (batch_size,26, 26, 3, 85)

The output y1, y2, y3, [... :2] refers to the offset relative to each grid point, [...,2:4] refers to the width and height, [...,4:5] refers to the confidence of the box, [...,5:] refers to the predicted probability of each category.

Y_true is the offset position, length, width and type on the grid of (13, 13), (26, 26) corresponding to each real box in a real image. It still needs to be coded to be consistent with y_pred's structure.

4.2. Calculation process of Loss

(a) Use y_true to extract the locations of real target points (m, 13, 13, 3, 1) and corresponding species (m, 13, 13, 3, 80) in this feature layer.

(b) The predicted output of yolo_outputs is 0 0 the predicted value y_pre after 0 0 is 0 0 shape (m, 13, 13, 3, 85)

(c) For each picture, calculate the IoU of all real boxes and prediction boxes. If the coincidence degree between some prediction boxes and real boxes is greater than 0.5, ignore it

(d) Ciou was calculated as regression loss, and only regression loss of positive samples was calculated here

(e) Calculate the loss of confidence.

(f) When calculating the loss of prediction type, it calculates the difference between prediction type and real type with actual target

5. Model solution

5.1. Comparison results before and after tagging

This article uses the xml.etree.elementTree (ET) module in Python to parse XML data.ET provides two objects: ElementTree transforms the entire XML document into a tree, and Element represents a single node in the tree. The interaction of the entire XML document (reading, writing, and finding the desired elements) is typically at the ElementTree level. For a single XML Element and its children, it is done at the Element level.

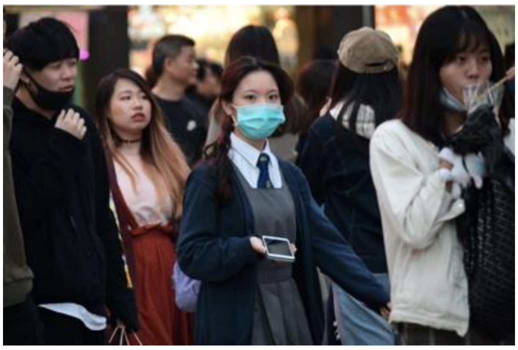

Figure 3: 250. PNG before the tag

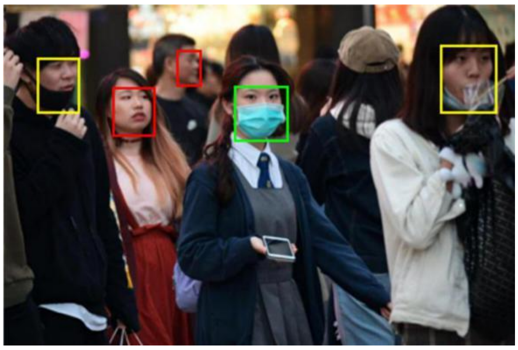

Figure 4: 250. After PNG tag

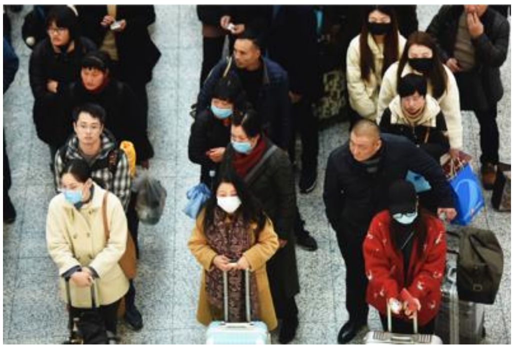

Figure 5: 477. PNG before the tag

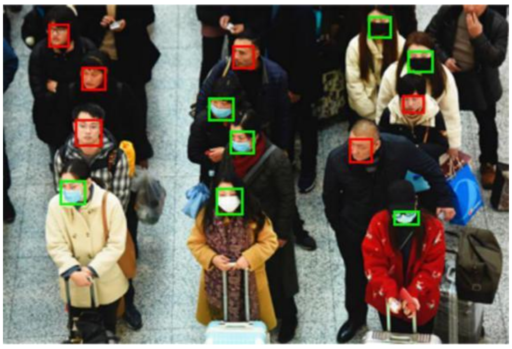

Figure 6: 477. PNG after the tag

5.2. Network training of face mask detection algorithm based on YoloV4-Tiny

Generate 2007_train. TXT and 2007_val.txt with voc_annotation.py. Run the train.py file to start the training.

1) Classes_path = 'model_data/voc_classes.txt', which corresponds to the data set in the experiment.

2) anchors_path= 'model_data/yolo_anchors. TXT ', which stands for the corresponding TXT file of the priori box.

3) anchors_mask = [[6, 7, 8], [3, 4, 5], [0, 1, 2]], used to help the code find the corresponding prior box.

4) model_path= 'model_data/yolo4_weight.h5', weight file downloaded from the web disk.

5) input_shape= [416, 416], the input shape must be a multiple of 32.5.

6) Mosaic = False, cosine_smoothing = False, Label_SMOOTHING = 0.

7) The training is divided into two stages, namely the freezing stage and the thawing stage.

5.3. The result of training model is displayed after verification

After many times of detection and verification [5], our team's face mask detection algorithm model based on YoloV4-Tiny achieved 0.953 mAP on the verification set.

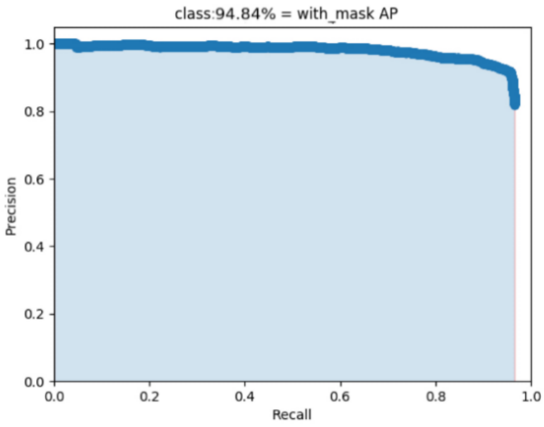

Figure 7: Test the total accuracy of the mask worn by the target

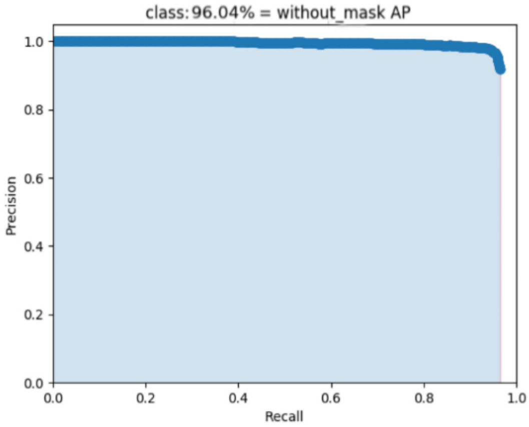

Figure 8: The total accuracy of detecting the target without a mask

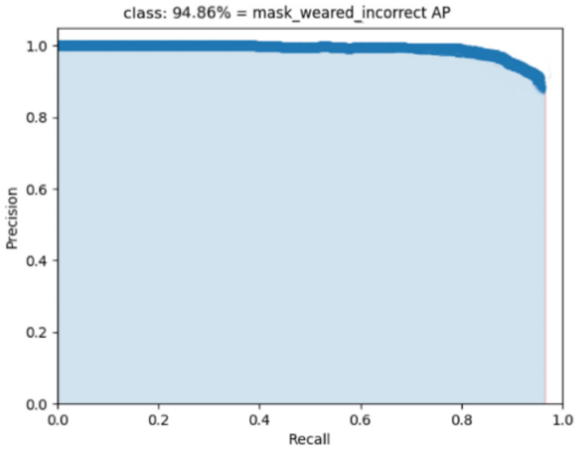

Figure 9: Detect the total accuracy of the target wearing a mask incorrectly

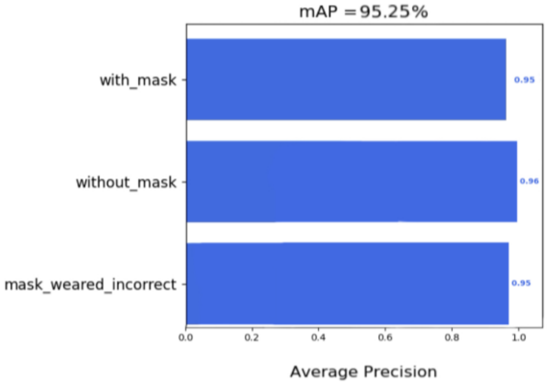

Figure 10: Detect the average loss rate of three categories of targets

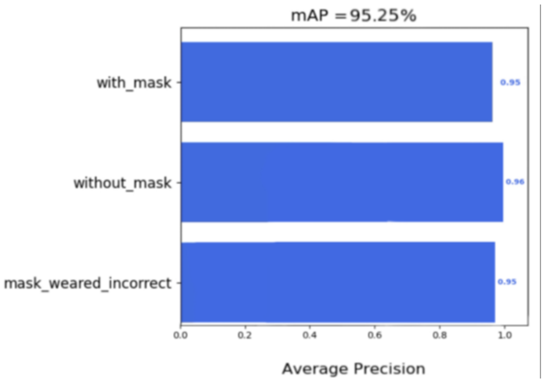

Figure 11: The total average accuracy of the three categories of detection targets

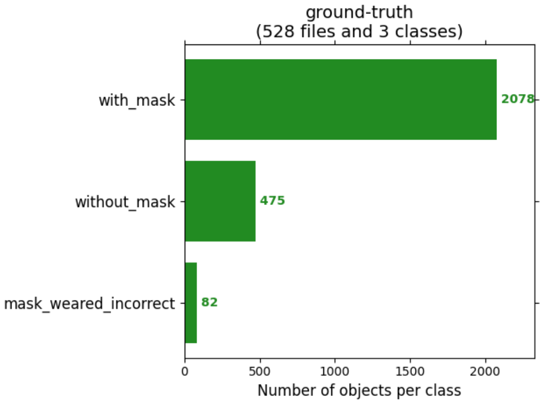

Figure 12: Detects the actual values of the target's three categories

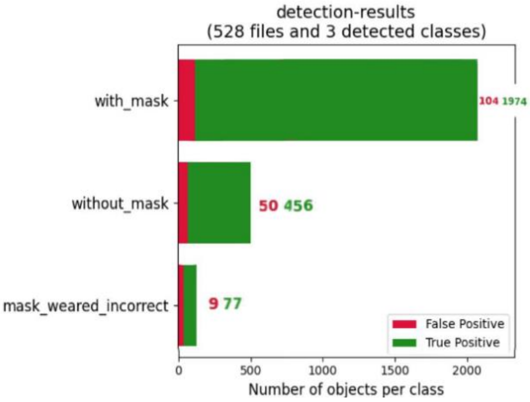

Figure 13: Detect the detection values of the three categories of the target

6. Conclusion

In order to extract the face information recorded in the tag file, this paper uses the xml. etree. ElementTree (ET) module in Python to read the XML file, and then distinguishes the color state of the target box according to the wearing status of face mask. Secondly, in order to meet the detection accuracy and real-time requirements of mask detection tasks in various scenarios, this paper established a face mask detection algorithm based on YoloV4-Tiny, trying to explore the accuracy of targets in dark environment, fuzzy targets, small size targets and occlusion targets. After several tests, the experimental results show that, the accuracy of the algorithm model in this paper reached 94.84% (with mask), 96.04% (without mask) and 94.86% (Mask weared incorrect). Its quantitative index < the number of correctly classified faces >/< the number of all faces contained in the tag file >, mAP is up to 95.25%, and the accuracy is greatly improved.

References

[1] Ye Mao, Ma Jie, Wang Qian, Wu Lin. Lightweight mask wear detection algorithm based on multi-scale feature fusion [J]. Computer Engineering 2021 01000 (3428): 1-13.

[2] Wang C Y, Liao H Y M, Wu Y H, et al. CSPNet: A new backbone that can enhance learning capability of CNN [C] / / Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 14-19, 2020, Virtual, Online, United States: IEEE, 2020: 1571-1580.

[3] Wang Yihao, Ding Hongwei, Li Bo, Yang Zhijun, Yang Jundong. Mask wear detection algorithm based on improved YOLOv3 in complex scenes [J]. Computer Engineering, 2020 and 46 (11): 12-22.

[4] Ye Zixun, Zhang Hongying. Lightweight improvement of YOLOv4 mask detection algorithm [J]. Computer Engineering and applications, 2021 57 (17): 157-168.

[5] Rui Geng, Yixuan Ma, Wanhong Huang. An improved helmet detection method for YOLOv3 on an unbalanced dataset[C]. Proceedings of 2021 3rd International Conference on Advances in Computer Technology, Information Science and Communications (CTISC 2021). IEEE, 2021.