

Face Mask Recognition based on YOLOV3

Yansong Chen *

School of optoelectronic Engineering, Chongqing University of Posts and Telecommunications,
Chongqing, China

* Corresponding author: yansong@ldy.edu.rs

Abstract. Wearing masks is considered one of the effective ways to prevent epidemics and protect physical health. In densely populated public places, detecting whether the mask is correctly worn has important application value, which can help managers reduce the potential risk of disease transmission. Previous works mainly rely on manual features, which are limited to fully expressing complex scene information. Thanks to the rapid development of artificial intelligence technology, automatic mask detection methods based on deep learning have achieved breakthroughs in speed and accuracy, while mask detection in complex scenes is still an open issue. To alleviate this issue, this paper proposes a mask detection method based on YOLOV3. Specifically, this study introduces regularization methods such as Drop Block and Label Smoothing, as well as data augmentation strategies such as Gridmask and Mosaic. This study further explores the recognition performance for refined scenarios including no mask, standard mask-wearing and incorrect mask-wearing. Extensive experimental results show the effectiveness of our method. By introducing the improved YOLOV5 algorithm to realize the mask detection function, the data structure is optimized through pruning, the data volume is reduced, the calculation speed is accelerated, and the mask detection is well realized.

Keywords: Mask recognition, YOLOV3, data enhancement.

1. Introduction

The arrival of the COVID-19 epidemic has made people see the importance of masks. As a practical and effective protective measure, maintaining the correct wearing of masks can effectively isolate respiratory droplets and other transmission paths. However, mask-wearing still mainly relies on personal self-awareness or manual supervision by management personnel, which is time-consuming and laborious. Although the epidemic is slowly leaving us, we still cannot do without wearing masks in many places, such as construction sites with flying sand, hospitals filled with viruses, or during the influenza season. Therefore, it is particularly necessary to design an automatic detection method that can be applied to the wearing of facial masks [1]. In the literature [2], the YOLOV3 neural network is used to detect masks inside the laboratory when combined with hardware equipment. After improving the algorithm, the accuracy has been significantly improved, and an early warning system has been added to remind people who are not wearing it, which has good promotion value for epidemic prevention and control requirements in public places.

Previous mask detection follows the framework of machine learning, where various manual features are designed to describe the complex application scenes and then fed into the classification head for mask recognition. With the development of embedded and computer vision, the detection network based on deep learning was deployed on hardware equipment to achieve mask detection, which greatly improved the efficiency and reduced the cost. Much efforts have been devoted to improving the detection performance. In order to achieve accurate mask detection, literature [3] introduced feature fusion and channel attention learning in convolutional neural networks, and optimized the loss function. However, the results listed the data but did not show the identification results. By improving YOLOV3, literature [4] replaced the the original backbone network of Darknet-53 with EfficientNet-B1, so that the parameters and calculation can be reduced. Meanwhile, the feature pyramid based on the attention mechanism is optimized to enhance the mask information, allowing the network to focus on the effective area of the image instead of focusing on all pixels in the image. Compared with the original YOLOV3, literature [4] reduced the detection accuracy by

1.73% and the number of model parameters by 79%, while increased the detection speed by 3.93 times. Literature [5] used the YOLOV3 algorithm and the classic darknet-53 network, mainly optimizing the loss function, and introduced the GIoU function as a training model for the loss function. It has better performance in multi-target detection, with faster detection speed and lower cost and higher accuracy.

Literature [6] In this review document: a variety of feasible methods for mask detection are introduced, and the advantages and disadvantages of each method are discussed, including the classic Multi-Stage Detectors and the classic convolutional neural network CNN, and are continuously improved on this basis. It also introduces the feasible application scenarios of each method, which has practical significance for promoting the application of the method in the future. [7] uses the MobileNetV2 machine learning algorithm that is different from the traditional yolo. It also uses the CNN convolutional neural network, but uses the kaggle data set for the data set, which will have a lower performance for masked face recognition. Big improvement. Though previous works have significantly advanced the mask detection accuracy, we argue that most of them focus on the scene whether the mask is worn rather than whether the mask is correctly worn, such as in some cases the mask did not completely cover the mouth and nose. To alleviate the above issue, based on the YOLOV3 detection framework, this paper transforms the task of facial mask wearing detection into a composite problem of facial detection and mask detection, and excavates and models the implicit relationship between the two end-to-end, thereby achieving accurate recognition of detailed scenes such as mask not wearing, incorrect mask wearing, and standard mask wearing. Based on data-driven strategies, we first conduct in-depth analysis of real-world scenarios and introduce regularization methods such as Drop Block and Label Smoothing, as well as data augmentation strategies from Gridmask and Mosaic, to enhance the diversity of the modeling process. Extensive experimental results show the effectiveness of our method.

2. Method

2.1. Revisiting YOLOV3 Network

2.1.1. Backbone

YOLOV3 uses Darknet-53. In the YOLOv2 framework, the previous generation of YOLOV3, the darknet-19 network is used. Compared with YOLOv2, YOLOv3 has a much larger number of parameters and uses 53 convolutional layers. YOLOV3 uses the backbone of Darknet-53 for feature extraction, where effective feature with three different scales can be extracted for following operation. The different scales of feature then are fused based on a pyramid structure, which are finally used multi-scale object prediction. Darknet-53 is not only more powerful than darknet-19, but also more efficient than resnet-101 and resnet-153.

As shown in Fig 1, the Darknet-53 network is composed of several 1×1 and 3×3 convolutional layers, in front of the BN layer and ReLU function. The suffix 53 means that there are 53 convolution layers to form the darknet network. There are no pooling layers and fully connected layers in the YOLOV3 network structure, so the transformation of the tensor size depends on the convolution kernel step size. Compared to YOLOV2, the scale transformation of YOLOV2 tensor is completed through maximum pooling.

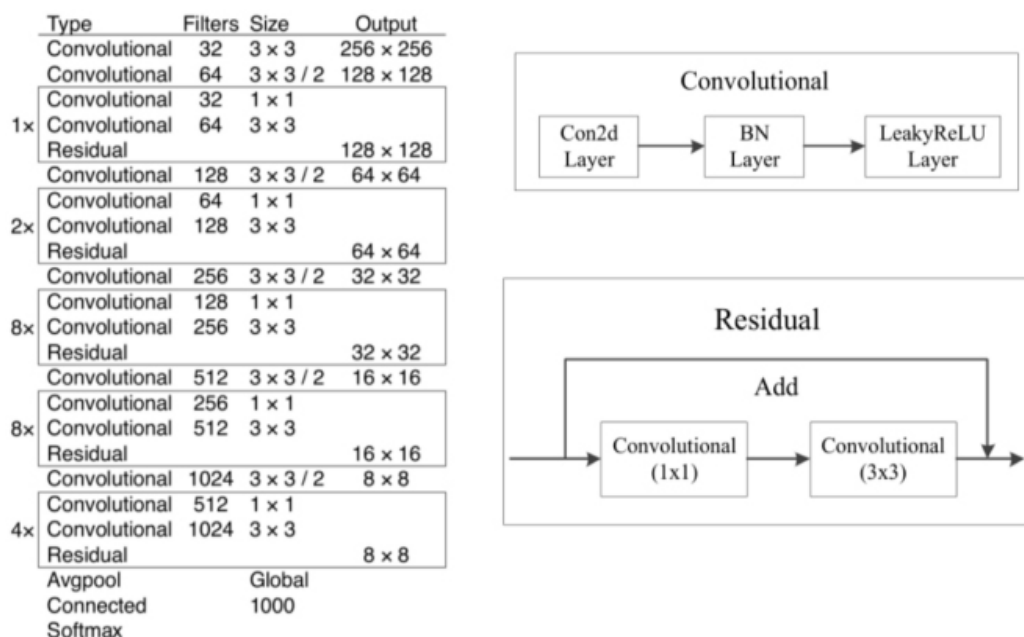


Figure 1. The network structure of Darknet-53

2.1.2. Key Steps

Compared with the previous yolo algorithm, YOLOV3 has improved the following points: (1) Achieve multi-scale prediction by introducing the Feature Pyramid Networks (FPN). (2) Optimize the backbone and add the residual structure. (3) Softmax is no longer used as a classifier, and the loss function is switched to binary cross-entropy loss. YOLOV3 follows the previous anchor and uses a set of borders of different sizes and aspect ratios preset by gridcell to cover different positions of the image. Fig 2 shows the three components of YOLOV3. (1) CBL: The smallest component in YOLOV3 is composed of convolution layer, BN layer, and Relu function. (2) Res unit: Drawing on the residual structure in Resnet, the network construction is more in-depth. (3) ResX: CBL is composed of X residual components and is a large component in YOLOV3. CBL plays a pooling role behind the Res module and reduce the resolution of the feature map.

By changing the original network connection and without increasing the calculation amount of the previous model, the performance of small object detection is greatly improved, and the multi-scale problem in object detection is well solved. FPN calculation process: (1) Use the output of each stage in the backbone network as a layer, and form the final output into a feature pyramid. In order to save memory, the output of the first residual is omitted. (2) Perform a 1×1 convolution on the feature pyramid to reduce the channel dimension, and then use the deep conv layer with strong semantic information to output 2 times upsampling. The principle of this algorithm is to use an interpolation algorithm between image pixels to insert new pixels. (3) Use horizontal connections to merge the results of the first and second steps. Since the feature map size x sampled by the previous layer is the same as the size y of this layer, plus the 2-fold relationship between the feature maps output by each stage. (4) The merged result uses 3×3 convolution to slow down the aliasing effect of upsampling, effectively avoiding jagged edges when grayscale changes. YOLOv3 is a good detector with a great performance of 0.5 IOU average AP on the COCO dataset [8].

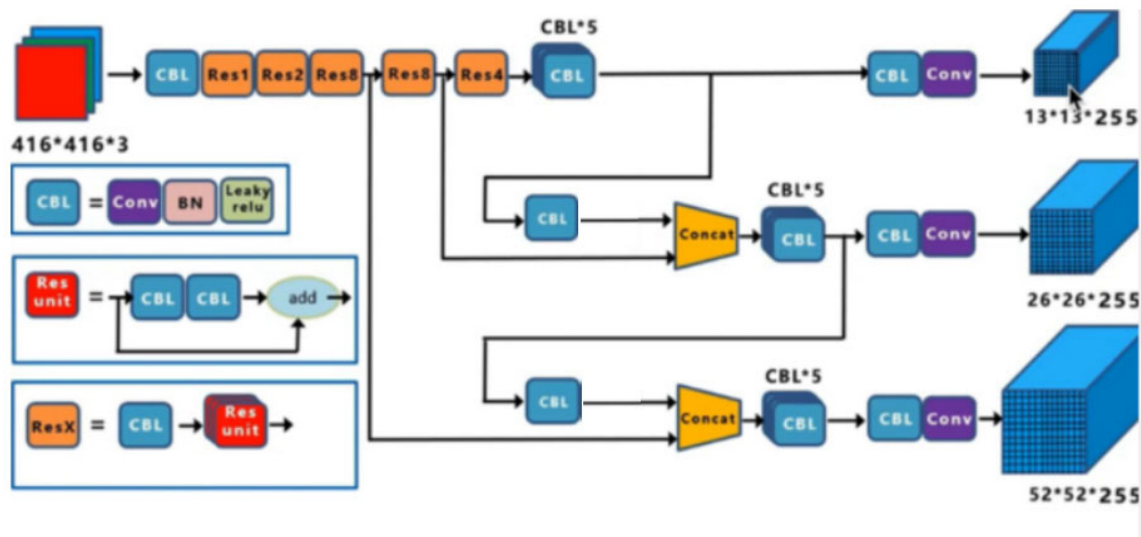


Figure 2. The framework of YOLOV3

2.1.3. Loss Functions

Since logistic is used instead of Softmax in YOLOV3, YOLOV3 replaces the classification loss function of YOLOV2 with bipartite cross entropy loss, as:

$$\begin{aligned} \text{Loss} = & \lambda_{\text{coord}} \sum_{i=0}^S \sum_{j=0}^B I_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i^j)^2 + (y_i - \hat{y}_i^j)^2 \right] + \lambda_{\text{coord}} \sum_{i=0}^S \sum_{j=0}^B I_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i^j} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i^j} \right)^2 \right] - \\ & \sum_{i=0}^S \sum_{j=0}^B I_{ij}^{\text{obj}} [\hat{c}_i^j \log(c_i^j) + (1 - \hat{c}_i^j) \log(1 - c_i^j)] - \lambda_{\text{noobj}} \sum_{i=0}^S \sum_{j=0}^B I_{ij}^{\text{noobj}} [\hat{c}_i^j \log(c_i^j) + (1 - \hat{c}_i^j) \log(1 - c_i^j)] - \\ & \sum_{i=0}^S I_{ij}^{\text{obj}} \sum_{c \in \text{classes}} ([\hat{p}_i^j \log(p_i^j) + (1 - \hat{p}_i^j) \log(1 - p_i^j)]) \end{aligned} \quad (1)$$

Center coordinate loss:

$$\sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i^j)^2 + (y_i - \hat{y}_i^j)^2 \right] \quad (2)$$

In the formula, the network output should be t_x and t_y , which are multiplied by the function and stride respectively, and will finally be mapped to the target on the 416×416 graph. When using this method to calculate the error, you essentially use $\times$ stride and the real target to resize to a target of 416×416 to calculate the error. When an anchor box in the grid is responsible for a real target, the bounding box generated by this anchor box should be compared with the box of the real target to calculate the center coordinate error. The width and height coordinate loss can be calculated as:

$$\sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{obj}} \left[\left(\sqrt{w_i^j} - \sqrt{\hat{w}_i^j} \right)^2 + \left(\sqrt{h_i^j} - \sqrt{\hat{h}_i^j} \right)^2 \right] \quad (3)$$

The network output in the formula should be t_x and t_y . When an anchor box in the grid is responsible for a real target, the bounding box generated by this anchor box should be compared with the box of the real target to calculate the width and height error. Similarly, the confidence loss can be calculated by:

$$\sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} [\hat{C}_i^j \log(C_i^j) + (1 - \hat{C}_i^j) \log(1 - C_i^j)] - \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{noobj} [\hat{C}_i^j \log(C_i^j) + (1 - \hat{C}_i^j) \log(1 - C_i^j)] \quad (4)$$

Usually, we use cross entropy to represent the confidence of the loss function. The first thing to understand is that regardless of whether the anchor box is associated with a target, the confidence

error needs to be calculated. The definition of confidence indicates the degree of confidence that there is indeed an object in the framed box and the degree of confidence that the framed box includes all the features of the entire object. The loss functions are composed of objects with and without objects respectively, and an additional loss weight coefficient is added to the object loss part. First, for an image, more content does not contain the object to be detected, and only a small part is used for detection, which results in the contribution calculation part with objects being much smaller than the contribution calculation part without objects, thus making the network. There is a tendency to identify cells that do not contain objects, so what we have to do is reduce this weight. The Classification loss is:

$$-\sum_{i=0}^{S^2} I_{ij}^{obj} \sum_{c \in classes} ([\hat{P}_i^j \log(P_i^j) + (1 - \hat{P}_i^j) \log(1 - P_i^j)]) \quad (5)$$

The classification error also uses cross entropy as the loss function. when an anchor box in the grid is responsible for a real target, then the bounding box generated by this anchor box will calculate the classification loss function.

2.2. DropBlock

Dropblock is a regularization method similar to dropout. It is different from other methods in that it removes adjacent regions from a layer's feature map instead of removing independent random units.[9] Dropout is a regularization technology widely used in fully connected layers, but its role in convolutional layers is minimal. The main reason is that adjacent elements in the feature map of the convolutional layer will share semantics. If an entire unit is dropped, Some of the elements do not have much impact on the overall situation and will still maintain the original semantic information. Considering problems such as this, a structural form of dropout is used for regularization, that is, block discarding.

Dropblock is a more structured form of Dropout, which is the use of block discarding. In addition to the convolutional layer, using Dropblock in the connection layer can improve accuracy, and the model's robustness to hyperparameter selection will also be improved during the training process.

block_size: Set a constant block size for all feature maps, regardless of feature map resolution. When block_size=1, DropBlock is similar to dropout.

γ represents the Bernoulli distribution probability in the drop process, which is the probability of the center point of a random drop. Considering that the elements in block_size*block_size are all set to 0, γ can be calculated as:

$$\gamma = \frac{(1 - keep_prob) * feat_size^2}{block_size^2 * (feat_size - block_size + 1)^2} \quad (6)$$

The number of elements in a traditional dropout drop is the drop probability multiplied by the total number of elements ((1-keep_prob) represents the proportion of drop drops). In the implementation, in order to ensure that the drop block will not exceed the original image, it is necessary to set an effective area for drop first.

2.3. Label Smoothing

Label smoothing is a regularization method that prevents the model from being overconfident in prediction and can effectively improve the problem of under-generalization. Using hard labels to train a model can lead to overconfident predictions by the model. A solution to this problem can be to make the network smoother by converting hard labels into soft labels. [10] Its working principle is mainly the conversion from hard label to soft label, which optimizes the network more smoothly. From a large perspective, label smoothing is actually a regularization tool that can be used on DNN. By using a weighted average between a uniform distribution and hard labels to generate soft labels, it is also used to deal with over-fitting problems in training and improve training efficiency. Label

smoothing not only makes the model's predictions more reasonable, but also gives it a certain ability to resist noise. It also makes up for a series of problems caused by insufficient signal volume, increases the amount of information, and uses a limited amount of data to make training more complete. The feature norm is reduced to gather the samples of each category, so that the unseen data can finally be predicted.

In the label network learning of classic one-hot encoding, if the probability of the model predicting the target category approaches 1, and the non-target category approaches 0, the value of z_i in the final logits vector will tend to infinity. Thereby allowing the model to learn in the direction where the difference between correctness and error is large. However, if the value of logit is too large, the adaptability will be reduced, and the model will overly trust its own prediction results. Of course, when the amount of training data is insufficient or the annotation is inaccurate, using the cross-entropy loss function may not be the best solution.

$$y_i = \begin{cases} 1, & i = target \\ 0, & i \neq target \end{cases} \quad (7)$$

Use the minimum loss function $H(y, p) = -\sum_i^K y_i \log p_i$ when training the network, the logits vector z output by the penultimate layer is calculated using the Softmax function:

$$p_i = \frac{\exp(z_i)}{\sum_j^K \exp(z_j)} \quad (8)$$

Label smoothing uses vectors to replace one-hot encoded label vectors y_{hot} :

$$\hat{y}_i = y_{hot}(1 - \alpha) + \alpha/K \quad (9)$$

Where k is the total number of categories in multi-classification, and α is a small hyperparameter (usually 0.1), that is:

$$\hat{y}_i = \begin{cases} 1 - \alpha, & i = target \\ \alpha/K, & i \neq target \end{cases} \quad (10)$$

This method is equivalent to adding noise components to the real distribution, thereby controlling the prediction results of the model and avoiding more arbitrary predictions. At the same time, the output difference between positive and negative samples will not be too large to avoid overfitting.

2.4. Mosaic

Mosaic is a data enhancement algorithm, whose main principle is to splice four pictures together to form a new picture, then obtain the anchor box of the picture, and finally return the picture to the model for learning. The model will perform data augmentation operations on the four images respectively, and finally output the corresponding position of the equal-sized mask under the image. The operations include: (1) Invert the original image left and right; (2) Scale the original image; (3) Adjust the hue, saturation, and brightness of the original image.

According to the random dividing line set in advance, $\min_offset_x$ and $\min_offset_y$ are generated to the center coordinates of $cutx$ and $cuty$ in the puzzle. For the final image, the frame is processed, then the undersized objects are discarded, and the previous image is covered in pie chart order.

3. Experiment and result analysis

3.1. Evaluation metric

The average precision AP (Average Precision) and average precision MAP (Mean Average Precision) are adopted as the evaluation indicators of the model. The corresponding formulas for AP and MAP are:

$$AP = \int_0^1 P(R) dR \quad (11)$$

$$Map = \frac{\sum_{i=1}^N AP_i}{N} \quad (12)$$

Where N represents the number of detection target categories.

3.2. Performance analysis

To evaluate the effectiveness of our method, an additional experiment is conducted. In the experiment, the training and validation sets were divided in a 1:9 ratio during network training. The Adam optimizer was used for network optimization, with a total of 200 epochs and a model training time of 50 hours. The AP values and mAP values of these four categories can be observed in Table 1, where Class0 is face recognition, Class1 is when there is a mask but not worn, Class2 is when the mask is not worn according to the regulations, and Class3 is when the mask is worn according to the regulations.

Table 1. AP and mAP for four different categories

	Class0	Class1	Class2	Class3
AP	0.61	0.95	0.87	0.89
mAP	0.833			
Parameter quantity	235M			

The final AP values of the four categories of output are 0.61, 0.95, 0.87, 0.89, and the mAP is 0.83. Since the first category of facial recognition is mostly labeled as masks in the database, the final output accuracy will be greater than that of other categories. Further analyzing the data in Table 1, we found that the addition of data augmentation modules and feature augmentation modules did not significantly improve the accuracy of the scene without wearing a mask. We believe that the possible reasons include: (1) the data augmentation module retained the original facial image as the scene without wearing a mask, but mainly enhanced the training sample size for mask wearing and incorrect wearing scenarios, Therefore, the model tends to improve the accuracy of correctly and incorrectly wearing masks during training; (2) For the scenario of not wearing a mask, it is degraded into a simple face detection task during model training. However, our design of the feature enhancement module is based on the purpose of better capturing the feature details of the mask. Therefore, theoretically, the feature enhancement module is more sensitive to the presence or absence of masks in the test sample.

3.3. Visualization

We also visualize the detection results of four different categories in Figure 3. Based on these visualized results, it can be found that the proposed method can accurately locate masks in the testing scene and capture the relative position relationship between masks and faces, accurately determining the specific scene in which masks are worn.

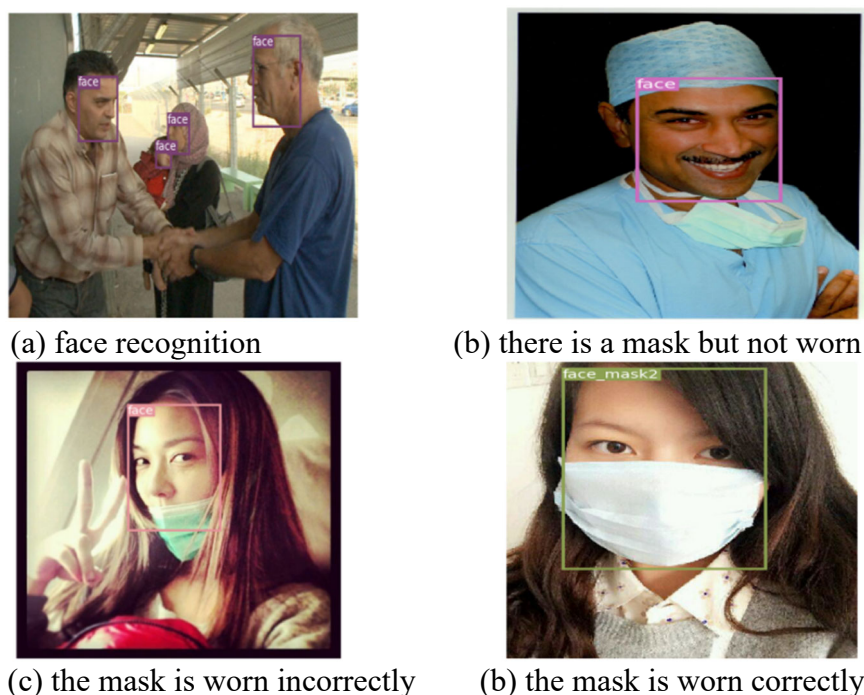


Figure 3. The visualization results of four different categories

4. Discussion and Conclusion

The application model still has room for improvement: the parameter amount of the classic YOLOV3 network is too large, resulting in too long training time. It is necessary to optimize the parameter amount through pruning and other methods, or use a lightweight YOLOV3 model for training. The dataset adds more datasets about faces for training to improve the accuracy of face recognition. The YOLOV3 model can be replaced by the newer YOLOV5 model in subsequent studies. Occlusion is a major difficulty in target detection. In the future, the model may be improved in terms of mask occlusion or face occlusion, and the model may be optimized by modifying the loss function or using online difficult sample mining technology.

The mask testing project proposed in this article is mainly based on the purpose of technology originating from life and then being applied to life. Although the COVID-19 epidemic is gradually moving away from us, there are countless new problems in life caused by masks. This article uses darknet-53 as the network framework and uses the weight.py file to pre-train the model. The data set uses Labeling software to label the images. DropBlock, Label Smoothing, Gridmask, mosaic and other algorithms were used to regularize and enhance the data. Subsequent improvements were made around the design of the anchor box, forming a classic anchor-free framework. The final output includes four types of target recognition, including face recognition, masks but not worn, masks not worn according to specifications, and masks worn according to specifications. The final mAP of the four types of target detection output reached 0.833. Since the data set is mainly because all the scenes are marked with masks and there are few parts about faces, there will be a large deviation in the recognition accuracy of the final output.

References

- [1] Wang Ke, Zhao Hui, Zhang Cheng, Wei Zihan. "Based on the improved YOLOv5 face mask recognition algorithm." *Informatization Research*, 2022, 48 (6), 38 - 45.
- [2] Ji Ruisong, Zhang Jianliang, Wu Yunjie. "Research on the application of image recognition technology in laboratory epidemic prevention." *LABORATORY SCIENCE*, 2022, 25 (6).

- [3] Zhang Shouming, Liu Kai, et al. "Research on lightweight mask detection algorithm based on YOLOv3." *Computer Science and Application*, 2022, 12 (6), 1700 - 1709.
- [4] Zhang Xiubao, Lin Ziyuan, TianWanxin, Wang Yan, Shen Haifeng, YeJieping , et al. "Face mask recognition technology in all-weather natural scenes. "SCIENCE CHINA PRESS, 2020, 50 (7), 1110 - 1120.
- [5] Chen Kexin, Wang Yude, et al. "Mask wearing for people in natural scenes based on improved YOLOv3 Detection algorithm." *Computer Systems & Applications*, 2021, 30 (2): 231 – 236
- [6] Mohammed Ali, Firas Amer, and Mohammed SH Al-Tamimi. "Face mask detection methods and techniques: A review." *International Journal of Nonlinear Analysis and Applications* 13.1 (2022): 3811 - 3823.
- [7] Sanjaya, Samuel Ady, and Suryo Adi Rakhmawan. "Face mask detection using MobileNetV2 in the era of COVID-19 pandemic." *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*. IEEE, 2020.
- [8] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." *arxiv preprint arxiv.*: 1804. 02767 (2018).
- [9] Ghiasi, Golnaz, Tsung-Yi Lin, and Quoc V. Le. "Dropblock: A regularization method for convolutional networks." *Advances in neural information processing systems* 31 (2018).
- [10] Zhang, Chang-Bin, et al. "Delving deep into label smoothing." *IEEE Transactions on Image Processing* 30 (2021): 5984 - 5996.