



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Research on mask wearing detection algorithm based on YOLOv3

by

Ruoyu Mu

Supervisor: **Michael Dacey**

Project submitted as part of the requirements
for the award of MSc Master of Science

September 2022

Abstract

Droplets and aerosols, which are airborne carriers, can spread the novel coronavirus, which is extremely contagious. The only way to stop the outbreak from spreading again is to wear masks in public. Additionally, some peculiar businesses (such grinding mills and dust-free industries) mandate that employees always wear masks while performing tasks. Manual patrol-style investigation is time-consuming and labor-intensive due to the wearing of thousands of masks. As a result, combining cameras for advanced PC-side monitoring has increased in popularity. The detection efficiency on low-performance PCs is subpar due to the fact that the bulk of mask wearing detection algorithms now place a strong emphasis on detection accuracy. The purpose of this topic is to investigate an efficient and precise light-mask wearing detection technique. It first checks to see if the face is covered using the YOLOv3 algorithm.

In order to meet the demands of low weight and quick detection of mask wear, MobileNetV2, a feature extraction network based on depthwise separable convolution and inverted residual structure, is first introduced beneath the core network of YOLOv3. By dividing the classic convolution into channel-by-channel convolution and point-by-point convolution, the network's parameter quantity and computational complexity are significantly reduced, improving the algorithm's detection speed. The channel's dimensions are initially increased. Dimensionality reduction improves the feature information that depthwise separable convolution recovers.

After the feature extraction network, a pyramid pooling structure is added to enable the fusion of global and local features, which improves the expressiveness of the final feature map and increases the model's detection accuracy without increasing the computational cost of the network.

The network training procedure is made more efficient by increasing the loss of edge box regression and include data. Using the L norm during training and the intersection ratio during assessment, YOLOv3 determines the loss of edge box regression and determines whether the target has been located. The issue the two are facing is unfair. Therefore, as a loss function linked to the intersection ratio, the loss of edge box regression is optimized in this study. The use of the Mosaic data augmentation technique in this research also improves the network's generalization and anti-interference capabilities.

Cross-validation on a dataset that was created by the author is used to assess the method in this research. The experimental results demonstrate that the proposed method significantly reduces network complexity while maintaining detection accuracy. On the experimental platform, detection speed reaches 125 frames per second, which represents a significant speed advantage over other superior detection algorithms. The approach described in this work also offers some adaptation to real-time detection in complicated environments while maintaining strong robustness under changes in brightness scale and occlusion.

Key words: loss of edge box regression; mask wearing detection; YOLOv3; MobileNetV2; spatial pyramid pooling;

Abstract.....	2
1 Introduction.....	4
1.1 Research background and significance.....	4
1.2 Research status at home and abroad.....	5
1.2.1 Research status of mask wearing classification algorithm.....	5
1.2.2 Research status of mask wearing detection algorithm.....	6
1.3 The main research content and chapter arrangement of this paper.....	8
1.3.1 Main research content.....	8
1.3.2 Chapter arrangement.....	8
2 Theory of Target Detection Algorithm Based on Deep Learning.....	9
2.1 Convolutional Neural Network Theory.....	9
2.1.1 Local connection and weight sharing.....	9
2.1.2 Composition of Convolutional Neural Networks.....	11
2.1.3 Forward Propagation and Back Propagation.....	14
2.2 Theory and development of target detection algorithm.....	14
2.2.1 Anchor-based detection algorithm.....	15
2.2.2 Detection algorithm based on anchor-free.....	15
2.2.3 Fusion of anchor-based and anchor-free detection algorithms.....	16
2.3 Summary of this chapter.....	17
3 Mask wearing detection method based on YOLOv3.....	17
3.1 The overall network structure of YOLOv3.....	18
3.1.1 Feature extraction network Darknet-53.....	18
3.1.2 Feature pyramid structure.....	19
3.1.3 Multi-scale detection layer.....	20
3.2 Anchor frame mechanism.....	20
3.2.1 Position prediction based on anchor box.....	20
3.2.2 Size priors for anchor boxes.....	21
3.3 Design of Loss Function.....	22
3.4 Dataset Construction and Labeling.....	22
3.4.1 Annotation files.....	23
3.5 Experimental results.....	23
3.5.1 Experimental platform and parameter settings.....	23
3.5.2 Evaluation Index of Detection Algorithm.....	24
3.5.3 K-means clustering results.....	24
3.5.4 YOLOv3 Model Evaluation Results.....	25
3.6 Chapter Summary.....	26
4 Optimization of Network Training Process.....	26
4.1 Optimization method for loss of edge box regression.....	26
4.1.1 Loss function GIoU.....	27
4.1.2 Loss functions DIoU and CIoU.....	28
4.2 Data Augmentation.....	30
4.2.1 Mixup data enhancement.....	30
4.2.2 Mosaic Data Augmentation.....	31
4.3 Experimental results.....	31
4.3.1 Loss function comparison experimental results.....	31
4.3.2 Data enhancement comparison test results.....	32
4.4 Chapter Summary.....	32
5 Test Results and Evaluation of Mask Wearing Test.....	33
5.1 Performance evaluation of the model.....	33
5.2 Chapter Summary.....	37
6 Summary and Outlook.....	37
6.1 Summary.....	37
6.2 Outlook.....	38

1 Introduction

1.1 Research background and significance

The latest coronavirus pneumonia outbreak, which began in Wuhan, Hubei Province at the end of December 2019, is still going strong and is costing a lot of money in many different countries across the world. According to statistics, the new coronavirus had caused 1.15 million deaths, 43.21 million infections, and 31.27 million cases of cure as of October 26, 2020. Thanks to coordinated efforts and effective prevention by all levels of government and the entire community, the domestic epidemic has been contained and has maintained a zero-growth trend for many consecutive months, but it has not yet been totally under control outside. As more and more national leaders and medical professionals become aware of China's proficiency in effectively halting the virus' transmission when it comes to epidemic prevention and control, several nations have since imitated this practice and requested China's help. Thanks in part to effective party and government leadership, a strong healthcare system, and the careful usage of masks by our people, China was able to contain the outbreak in just two months.

Despite the steady containment of the internal epidemic, we cannot let up on our guard. The new coronavirus is extremely contagious, so an outbreak of the epidemic might happen at any time. As a result, people must intentionally wear masks when entering and leaving busy public spaces. Relying solely on public awareness is insufficient. There are still a lot of people that take chances and disregard mask wearing. Additionally, several unique workplaces, such dust-free workshops, grinding workshops, etc., mandate that employees put on masks at all times while performing tasks. In order to prevent this, the government, schools, and factories must step in and monitor the use of masks whenever people enter and exit public spaces. It goes without saying that depending solely on human resources to carry out post-post screening has drawbacks including high labor intensity, low efficiency, limited coverage, and slow timeliness. As a result, intelligent and automatic mask detection is required.

The current coronavirus [1] pneumonia outbreak that began in Wuhan, Hubei Province at the end of December 2019 has persisted and has been very expensive for several countries around the world. Statistics show that as of October 26, 2020, the new coronavirus had caused 1.15 million fatalities, 43.21 million infections, and 31.27 million cures. The internal epidemic has been contained and has maintained a zero-growth trend for many consecutive months as a result of the concerted efforts and efficient prevention by governments at all levels and the entire population, but it has not been totally controlled outside. As national authorities and medical professionals become more aware of China's proficiency in effectively controlling the spread of the virus when it comes to epidemic prevention and control, several countries have since imitated this practice and requested China's assistance. Strong party and government leadership, an effective healthcare system, and the careful usage of masks by our people were some of the reasons why China was able to manage the outbreak in just two months [2].

We still can't let up on our vigilance despite the home plague being progressively under control. The epidemic could recur at any time because of the new coronavirus's high level of contagiousness. Citizens must consciously wear masks when entering and leaving public areas with heavy foot traffic. Relying on the public's consciousness is insufficient. People still take chances and disregard the use of masks in large numbers. Additionally, several unique workplaces [3], such as dust-free workshops, grinding workshops, etc., mandate that staff members always wear masks while performing their duties. As a result, whenever people enter and exit public spaces [4], the government, schools, and factories must step in and monitor the wearing of masks. Obviously, conducting post-post screening alone with human resources has drawbacks including high labor intensity, low efficiency, limited coverage, and

slow timeliness. Therefore, it is necessary for the mask wear detection to be automated and intelligent:

(1) Find solutions for manual post-post investigation's shortcomings. Given the aforementioned issues with manual inspection, a mask wearing detection approach based on a target detection algorithm is created. This method can be utilized to identify mask wear fast and correctly in order to supplement or completely replace manual inspection [5].

(2) Encourage the development of detection technology for people wearing masks. The existing detecting techniques still have a lot of issues. The majority of algorithms have poor applicability and are unable to perform real-time detection on devices with low processing speeds. The purpose of this topic is to investigate a quick and simple detection technique [6].

1.2 Research status at home and abroad

The implementation of research findings into the real manufacturing process is the ultimate goal of engineering research. The level of intelligence in security, inspection, and other industries has increased because to developments in technology like computer vision and electronic hardware. The target detection technology has advanced from the earliest off-line detection system tests to the current effective and quick real-time detection system, and it has started to be marketed. Various industries are trending toward progress in the direction of automation and intelligence at the same time as science and technology are developing and human demands are changing. Many academics, professionals, and the R&D divisions of various corporations have conducted research and continuously presented new and great algorithms in the area of intelligent mask wear detection. These algorithms primarily revolve around two concepts: the first is a single-target classification method, which only evaluates whether a single individual is wearing a mask; the second is a multi-target detection algorithm, which seeks to identify and locate all potential targets. This involves locating each face in the photo or video, determining if it is wearing a mask, and marking the location of each face with a rectangle frame, as illustrated in Figure 1-1. (face-mask in the figure is wearing a mask).



Figure 1-1 Application of target detection algorithm in mask wearing detection

1.2.1 Research status of mask wearing classification algorithm

Classification algorithms are a crucial subset of supervised learning that are typically applied to discrete prediction problems with non-continuous values, while they can also be applied to continuous prediction problems. To put it simply, a decision function that can map input and output is acquired by learning the sample data. A classifier is the name of this decision-making process. Classification is the process of a classifier predicting an output from an unidentified input. Binary classification and multi-classification are terms used when the output follows the binomial distribution and there are multiple categories. The classification of mask use in this study falls into two categories, meaning that there are

two possible outcomes: either the target is wearing a mask or is not.

The classification algorithm, a key component of supervised learning, is frequently used for discrete prediction problems with non-continuous inputs, while it can also take inputs in the form of discrete or continuous values. In other words, learning the sample data results in a decision function that maps input and output. A decision-making procedure like this is called a classifier. When a classifier predicts an outcome based on an unclear input, this is referred to as classification. The words "binary classification" and "multi-classification" are used to refer to two distinct methods where the outcome contains several categories and follows the binomial distribution. In this paper's classification of mask wear, there are thus two conceivable outcomes: either the target is wearing a mask or they are not.

Because it can only identify a single target, the mask wearing classification approach is more suitable for access control systems that only permit one individual to pass by at a time. In comparison to the thousand detection algorithm, the utilization is restricted. There aren't many studies on the classification method for mask wearers as a result, whether domestically nor internationally, including in China. Research.

1.2.2 Research status of mask wearing detection algorithm

The goals of the object detection approach include finding all the interesting objects in the image, pinpointing them, and determining their category. The target detection technique has two branches. A classification task on the first branch requires more foreground and background separation than a simple picture classification assignment; the second branch is a regression task that predicts where the target will be, that is, the edge box of the output target. All of the faces in the image must be detected, marked with edge boxes, and their mask-wearing status must be established in order to complete the mask wearing detection procedure. Consequently, a large number of studies have investigated tens of thousands of target detection algorithms, used them to complete the task of mask wearing detection, and are now focused on suggesting a generic algorithm that is efficient and precise and can be tailored to different scene detection scenarios.

According to the literature review, the current mask wearing detection strategies can be roughly divided into three groups: (1) approaches based on face detection and classification models; (2) techniques based on the YOLOv3 series; and (3) additional approaches [7].

(1) Face detection and classification model-based methods. Face detection and mask wear detection have a strong association. The foundation of mask wearing detection is face detection. The latter must differentiate between faces with and without masks and further categorize faces depending on the former. person's face [8].

Principal component analysis (PCA) was utilized by Md. SabbirEjaz [9] to express the face portion as a feature vector, and the closest neighbor approach was used to generate the feature vector in the target image and the template image. The recognition rate of faces with masks is weak, and this method is best for recognizing faces without them. A trigger alarm system was suggested by A. Nieto- Rodriguez[10] et al. to determine whether a medical mask is being worn. A face detector (Cascade LogitBoost) and a mask detector (Cascade GentleAdaBoost) produce a list of candidate detections when a new image enters the system. These candidate detections are represented by bounding boxes and confidence scores, and the detection stage is then eliminated by passing the list through an HSV color filter. The confidence score of the overlap between the two lists is used to determine the discovered category at the end. To enhance the expressive power of the feature map and optimize the loss function of the edge box regression, Niu Zuodong [11] and others proposed integrating the attention mechanism into the feature pyramid of the RetinaFace method. The ResNet-50 classification model, which integrates the attention mechanism and

the optimized loss function, is used by Didi Chuxing Company [12] to determine whether the face area wears a mask after first detecting the face based on the DFS algorithm, expanding the face area based on the face features, and finally, identify. According to the principle of "detection first, classification afterwards," Specta Technology IoT Co., Ltd. initially detects faces using the portable CenterFace [13] face detector before classifying mask wear using MobileNetV2. In order to determine whether a face is wearing a mask, Baidu suggested using the self-developed PyramidBox [14] algorithm for face detection. A new mask wearing state recognition technique was created by Qin[15] et al. by fusing a face detection network with an image super-resolution classification network. The algorithm flow includes picture preprocessing, face detection and cropping, and SRCNN network for image processing. MobileNetV2's super-resolution reconstruction can detect whether a mask is being worn.

(2) A strategy based on the YOLO series. The YOLO series is the most conventional subset of the universal target identification method. There are two of them that have effectively balanced high precision and high efficiency, YOLOv3 and YOLOv4 [16]. A network built on the two is directly used in many research projects. Accept the fact that mask usage has been discovered.

Wang Yihao and colleagues suggested combining the cross-stage local network (CSPNet), introducing the spatial pyramid pooling structure, combining bottom-up and top-down feature fusion strategies to achieve feature enhancement, and optimizing with CIoU as the loss function in order to enhance the YOLOv3 feature extraction network [17]. The area of the face that was wearing the mask was identified by Xiao Junjie [18] based on the detection findings after the mask wearing detection was realized using the YOLOv3 algorithm. Then, using the YCrCb elliptical skin color model, it was decided whether the mask wearing was normal. Cao Chengshuo [19] and colleagues implemented the channel attention mechanism in the YOLOv3 feature extraction network and used the bottom-up path aggregation technique in the feature pyramid structure in order to efficiently leverage the shallow fine-grained structural information. The target localization loss is optimized. Guan Junlin [20] and others proposed to rebuild and integrate the high-frequency and low-frequency components of the image, and to use the YOLOv4 algorithm to recognize mask wearing in order to reduce the impact of illumination on the detection findings.

(3) A technique that applies the YOLO series. The global target identification technique that has been around the longest is a subset of the YOLO series. Two of them, YOLOv3 and YOLOv4 [16], have successfully balanced high precision and high efficiency. A network based on the two is directly used in several investigations. It should be noted that a mask has been found.

To improve the YOLOv3 feature extraction network, Wang Yihao and colleagues suggested combining the cross-stage local network (CSPNet), adding the spatial pyramid pooling structure, combining bottom-up and top-down feature fusion strategies to achieve feature enhancement, and optimizing with CIoU as the loss function. Xiao Junjie [18] analyzed the area of the face that was wearing the mask based on the detection findings and then utilized the YCrCb elliptical skin color model to determine if the mask wearing was normal after the YOLOv3 algorithm successfully realized the mask wearing detection. To effectively utilize the shallow fine-grained structural information, Cao Chengshuo [19] and associates developed the channel attention mechanism in the YOLOv3 feature extraction network and used the bottom-up path aggregation technique in the feature pyramid structure. It lessens the target's localization loss. Guan Junlin [20] and others recommended merging the high-frequency and low-frequency components of the image, employing the YOLOv4 algorithm to detect mask wear, and decreasing the impact of illumination on the detection results.

1.3 The main research content and chapter arrangement of this paper

1.3.1 Main research content

The real-time detection effect requires a lot of PC hardware when the current mask wearing detection technique is in use because it is fairly complicated. In order to overcome this problem and further reduce the hardware cost required to implement the mask wearing detection method, the main research goal of this paper is to investigate a lightweight network with high accuracy and real-time detection on a low-performance PC [21]. We initially identified the application of YOLOv3 in mask wearing detection by developing a data set and an algorithm working environment. YOLOv3 was used as the main detection network to complete the network's lightweight design and enhance the network's structure.

For training and evaluation, first create the data set for mask wearing detection. The information for this topic is taken from three common public data sets, and the annotation files of the images are also changed based on the K-means method to make the annotation as accurate and consistent as possible. Following the clustering of the dataset's actual annotation boxes and the setting of the anchor box's size prior in line with the findings of the clustering, the network's training process is optimized. The YOLOv3 feature extraction network is then enhanced to MobileNetV2 in order to decrease the network's parameters and computational complexity. Boost the generalization efficiency of the network. The model is trained using the five-fold cross-validation method to somewhat solve the over-fitting issue. The model's detection capabilities are assessed using images and videos, and experiments with brightness, scale variations, and occlusion are also included. to evaluate the robustness of the model.

1.3.2 Chapter arrangement

The paper is divided into the following sections:

First Chapter Introduction. Before analyzing the current research status, primary research methodologies, and issues with mask wearing detection algorithms at home and abroad, the research background and significance of this topic are first clarified. Finally, the primary study topic of this essay as well as the chapter organization that follows are introduced.

Chapter 2: The associated theory of the deep learning-based target detection algorithm To describe the creation and major algorithms of the target detection algorithm, two branches—anchor-based and anchor-free—are first described, along with the basic principle and composition of the convolutional neural network.

YOLOv3-based mask wearing detection is covered in Chapter 3. First, the YOLOv3 algorithm's network structure, comprising the feature extraction network and the feature pyramid structure, is examined. Second, the calculation of the loss function is thoroughly discussed, as is the design basis of the anchor box in the prediction layer. The experimental setting, the data set, and the YOLOv3 algorithm are finally introduced. detection effectiveness.

Chapter 4: Improving how the network is trained. To improve the convergence impact of the network, first evaluate the issue of the loss of the edge box regression in the original loss function, and then optimize it as a loss function related to the intersection ratio. In addition, the two data augmentation techniques Mixup and Mosaic are compared. The network's capacity for generalization is further enhanced with the addition of data.

Chapter 5: Experimental findings and analysis of the detection of mask wear. First, a thorough examination of each index of the method used in this study is conducted, and it is then compared to YOLOv3. The model's resilience is then confirmed through a number of trials with brightness, scale

variation, and occlusion. The method described in this paper is compared to the two that were created using the current C++ environment. comparing and analyzing huge, great algorithms quantitatively.

Summary and outlook for Chapter 6. Summarize the job, point out the flaws, and mention what has to be done better in the following step.

2 Theory of Target Detection Algorithm Based on Deep Learning

Tensor computing for deep learning is made possible by the ongoing improvement in graphics processor speed. Neural network training now takes a fraction of the time it once did thanks to CUDA, an open source parallel computing framework, and CUDNN, a related computational acceleration package created by NVIDIA. The deep learning frameworks (Tensorflow, Pytorch, Caffe, etc.) developed by Google, Facebook, and other firms have improved the efficiency of researchers who are designing and implementing neural networks. By relying on the quickly evolving field of deep learning, the convolutional neural network, which has been transformed into an end-to-end detection algorithm, increasingly replaces the technique of manually building and extracting features from the target detection algorithm. In less than 8 years from the initial RCNN to the present YOLOv5, the target identification method based on deep learning has advanced from real-time detection on the PC side to the deployment application on the mobile phone side.

2.1 Convolutional Neural Network Theory

As the name implies, a convolutional neural network is a neural network that performs convolution operations. A convolutional neural network's fundamental building blocks are its convolutional, pooling, activation, and fully connected layers. It can understand the relationship between a vast number of input and output data and mine the internal features of the data by replicating the perception mechanism in the visual cortex of the brain. The convolutional neural network excels in the areas of picture classification, object identification, image segmentation, and image style transfer due to the intimate connections between its layers and the spatial organization of the network. Through weight sharing, local connections, and downsampling, it completely learns the intrinsic properties of the input. This considerably decreases network complexity. It is also extremely invariant to operations like translation and scaling, and it processes two-dimensional images with good robustness. Efficiency and robustness in computation.

2.1.1 Local connection and weight sharing

Each neuron in any layer of the convolutional neural network only responds to the neurons in the layer below it, which is known as local connection, local receptive field, or sparse connection, as shown in Figure 2-1. Only the input layer's neurons a1, c1, and d1 are connected to the response layer's neuron b2. It is a fully connected neural network if local connections are not used, meaning that every neuron in the response layer is connected to every other neuron in the input layer. Local connections are used depending on the qualities of the image. Due to the possibility of nearby pixels in a picture belonging to the same item, the spatial correlation of the image shows a relatively significant correlation between adjacent pixels whilst the correlation between distant pixels is relatively small. To retrieve the complete semantic information of the entire image, or high-level features, each neuron just needs to send back the pixel information in the local range of the image and then weight this local feature information in a higher dimension. The partial connection approach can significantly minimize the amount of network parameters and lower the danger of overfitting when the input pixel matrix is big since the full connection

method introduces a large number of parameters and makes it harder to train the network.

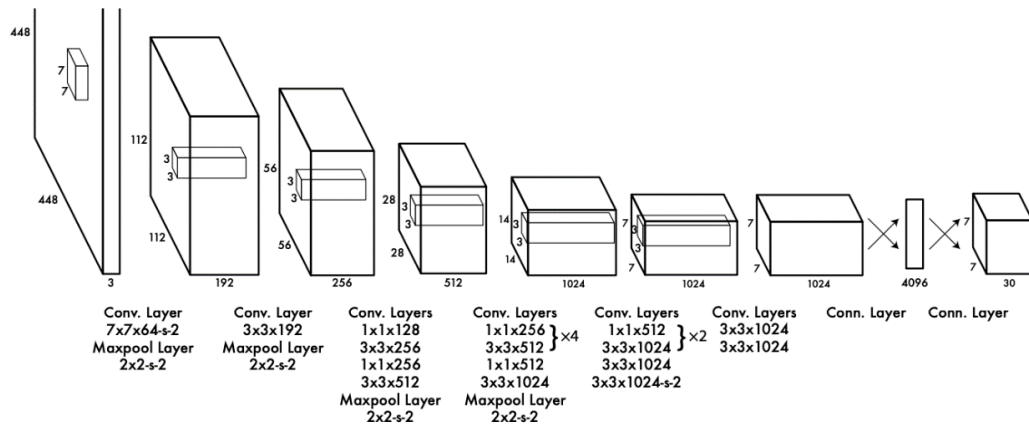


Figure 2.1 YOLO algorithm flow

Convolution operations are sequentially carried out on the feature map in the convolutional neural network in the form of weighted fusion in order to extract the input image. The convolution kernel in the convolutional layer slides on the feature map with a specific size of the pane at a specific step size. Potential local features in are depicted in Figure 2-2, where the stride is 2 and the convolution kernel is a 3*3 pane. Weight sharing is the process of applying convolution operations to the entire feature map using the same convolution kernel. Each convolution kernel parameter is determined by a particular point on the pixels in the feature map's local range. The same feature can be found in several parts of the image in this fashion. The convolutional neural network can therefore adapt to the small-scale translation of the objects in the image and has strong translation invariance.

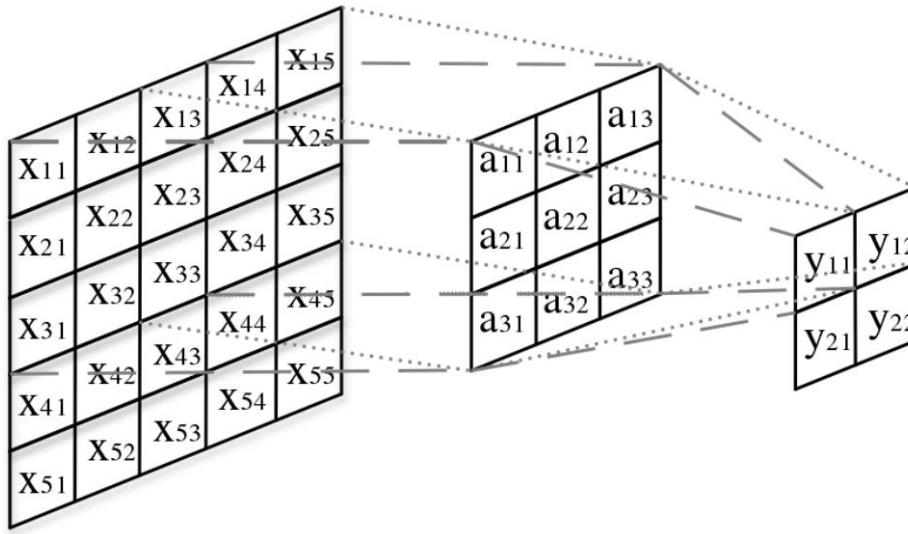


Figure 2-2 Weight sharing

Convolutional neural networks share weights, which minimizes the complexity of the network and the risk of overfitting and, on the other hand, set numerous convolution kernels in the convolutional layer to extract more information, which reduces the number of parameters for network training. The feature map is more expressive because to the feature information.

2.1.2 Composition of Convolutional Neural Networks

Although the network structure of convolutional neural networks is indefinitely changeable, it is ultimately always composed of the most basic convolution layer, the normalization layer, the activation layer, the pooling layer, and the fully connected layer.

A variety of convolution kernels are used in the convolution layer to extract visual features. Figure 2-3 depicts a schematic illustration of a fundamental convolution operation. Convolution is often defined as the process of locally summing and weighting the image's pixel information using the translation-invariant parameters of the convolution kernel. The following is a clearer definition of convolution:

$$a_j^l = f\left(\sum_{i \in M_j^l} a_i^{l-1} \times k_{ij}^l + b_j^l\right)$$

where a_j^l is the convolutional layer l First j the output value of the unit, f is the activation function of the convolutional layer, M_j^l represents the set of local input feature maps, and k represents the convolution kernel. As shown in Figure 2-3, the convolution kernel is similar to a sliding pane. It slides right or down on the input image according to a fixed step size (the step size is 1 in the figure), and

obtains the output feature map through linear weighting. The formula for calculating the size of the output feature map is:

$$S_{out} = \frac{S_{in} + 2 \times P - S_k}{2} + 1$$

In the formula, S_{out} is the width (height) of the output feature map, S_{in} is the width (height) of the input feature map, S_k is the width (height) of the convolution kernel, P stands for border padding (the edges are filled with 0). The image needs to be edge-filled for two reasons:

(1) Each convolution will shrink the size of the feature map, as seen in Figure 2-3. The final feature map will be small if the input image is small and there are many convolutions. There might only be one pixel left;

(2) the edge pixels of the input image only take part in one convolution operation, whereas the middle pixels take part in multiple convolution operations, which will obviously suppress the edge information of the image. This results in the incompleteness or loss of image edge features during the convolution process. The two most popular filling techniques are valid filling and same filling. The latter is filled, while the former is left empty, maintaining the feature map's size.

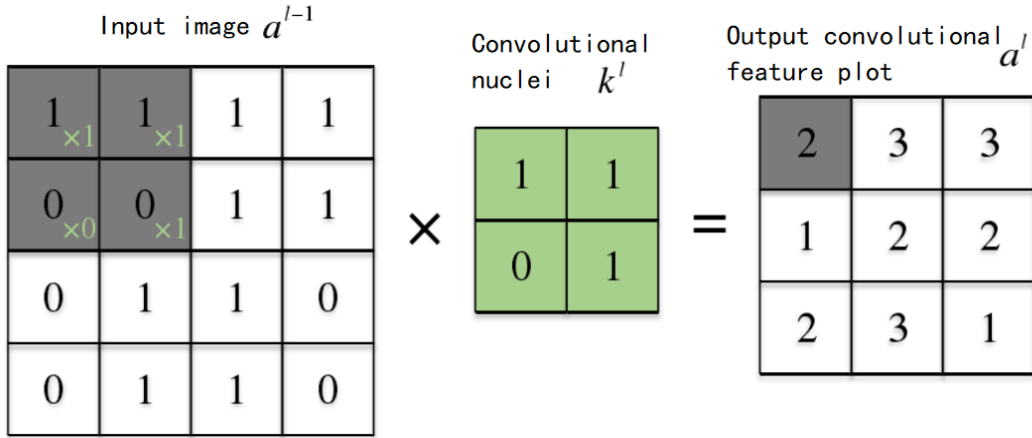


Figure 2-3 Schematic diagram of convolution calculation

The primary purpose of the normalization layer is to apply extra restrictions on each layer's data distribution during training to avoid gradient dispersion. Since the output of each layer in a convolutional neural network is calculated in separate layers, it is evident that the distribution of the different layers differs from the distribution of the input layers, and this discrepancy will grow as the network deepens. The network therefore needs to Converge is tough since each iteration involves learning new data distributions. The purpose of the normalization layer is to adjust each layer's input distribution so that its mean and standard deviation follow a normal distribution. The key procedure is: (1) Calculate the mean of each batch of input data; (2) Calculate μ_Ω the variance of the σ_Ω^2 input data for each batch; (3) Normalize the data for each batch; (4) Training parameters γ , β . The calculation process is as follows:

$$\mu_\Omega = \frac{1}{m} \sum_{i=1}^m x_i$$

$$\sigma_\Omega^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_\Omega)^2$$

$$\hat{x}_i = \frac{x_i - \mu_\Omega}{\sqrt{\sigma_\Omega^2 + \varepsilon}}$$

$$y_i = \gamma \hat{x}_i + \beta$$

where Ω is the m input samples of each batch, $\Omega = \{x_1, x_2 \dots x_m\}$ ie .

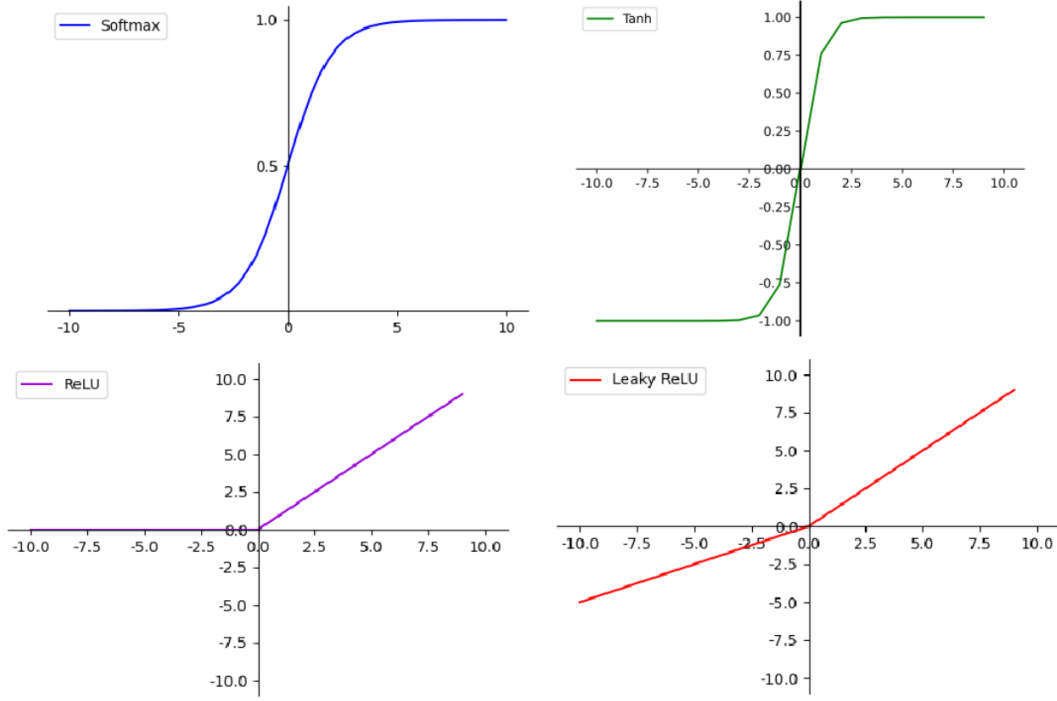


Figure 2-4 Several commonly used activation functions

The neural network's activation layer's job is to apply nonlinear changes to the features. The final expression of the neural network is still a linear function if there is no nonlinear modification. The neural network may potentially approximate any function when nonlinear elements are included through the activation layer, giving it a better capacity for classification. Several frequently used activation functions are displayed in Figure 2-4.

In order to reduce the number of parameters in the network, prevent overfitting to some extent, and lessen sensitivity to feature position, the pooling layer's function is to reduce the dimension of the feature. Maximum pooling and average pooling are the two primary categories of pooling functions. The computation process is comparable to convolution. As illustrated in Figure 2-5, the sliding window method is used to acquire the average or maximum value in a region of a certain size of the input feature map. Pooling computations are straightforward. In order to highlight the background features in the image, the average pooling can keep the variance of the feature points in the neighborhood within a manageable range, while the maximum pooling takes the largest feature points in the neighborhood in order to minimize the error brought on by the convolution layer's parameter settings. It is best for the preservation of texture features in the image to use the approximated mean shift of.

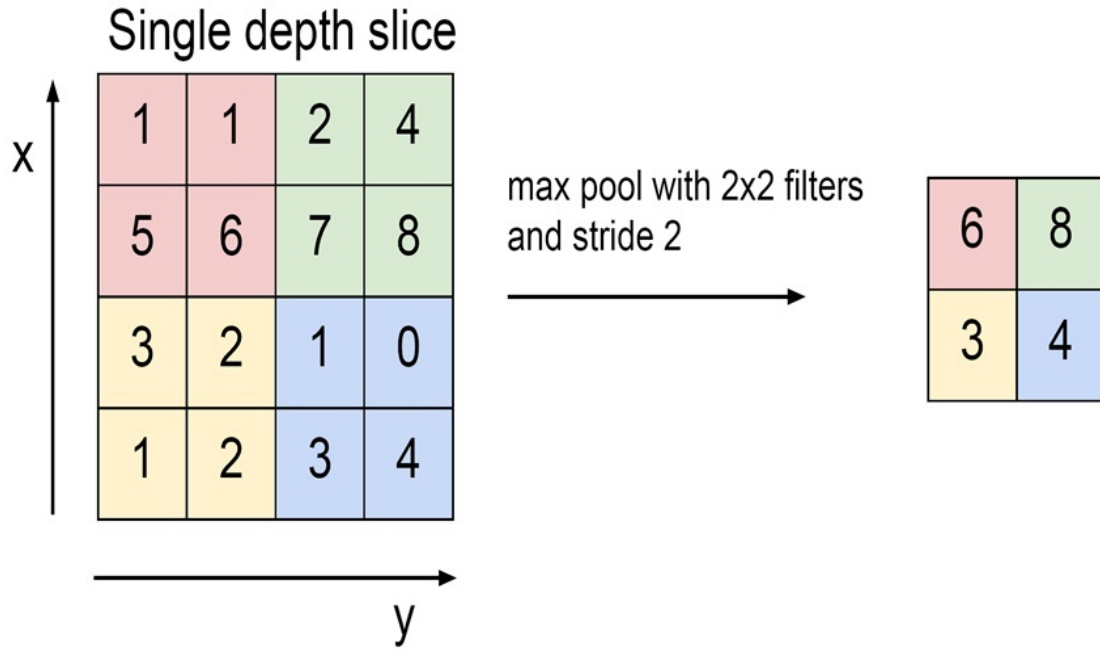


Figure 2.5 YOLO Convolution calculation

Formula illustrates how the fully connected layer, which typically appears at the end of the classification network, multiplies matrix vectors to produce the one-dimensional vector that is output as a three-dimensional feature vector by the convolutional layer [22].

$$y = Wx + b$$

W stands for weight, and b for bias. The fully connected layer's core function is a dimensional change of the feature, and the convolutional layer's local features that were extracted are weighted. The fully connected layer serves two main purposes: first, it extracts and incorporates useful features in the feature dimension transformation process; second, it conceals the semantic expression and maps samples from the feature space to real labels, which actually produces a classification effect. The fully connected layer is gradually replaced by 11 convolution or global average pooling because its operating parameters are too numerous in the actual network topology and may be realized by the convolution layer [23]. However, research continues to support the fully connected layer's benefits in transfer learning [24].

2.1.3 Forward Propagation and Back Propagation

Convolutional neural networks use back propagation and forward propagation for parameter updates [25]. According to the network's weight, bias, and activation function, the forward propagation process determines the difference between the predicted value and the true value; the back propagation process determines the gradient of the parameters in accordance with the chain rule, and then combines the learning rate with the network parameters. the procedure for updating systems to reduce this mistake.

2.2 Theory and development of target detection algorithm

Nearly all of the common target detection algorithms used today are deep learning-based. The algorithms have several branches that are determined by various principles. It can be classified into three categories: anchor-based (with anchor frame) detection algorithms, anchor-free (without anchor frame), and fusion anchor-based and anchor-free detection algorithms, depending on whether the anchor frame is used to extract candidate regions. This work divides the anchor-based detection algorithm into one-stage and two-stage iterations for development.

2.2.1 Anchor-based detection algorithm

The purpose of the anchor-based detection technique, which can be divided into single-stage and two-stage methods depending on how the candidate areas were formed, is often described as the classification and regression of a large number of candidate regions (RegionProposals). The two-stage method completes the detecting process in two parts [26]. The first stage creates a massive amount of candidate regions in order to compile an exhaustive list of all locations where objects could possibly occur. In the subsequent stage, these candidate regions are categorized and their placements are modified. The single-stage strategy replaces the time-consuming candidate region extraction process with the anchor box mechanism, which successfully completes the classification and positioning of the target based on the regression issue in one step. Due to two position regressions, the two-stage method marginally improves positioning accuracy, but the single-stage methodology is overall significantly faster.

The two-stage approach was primarily inspired by the Faster-RCNN series of algorithms, which were evolved from the RCNN[27] algorithm. For each input image, RCNN locates about 2000 potential regions using selective search (SelectiveSearch). The convolutional neural network uses these candidate regions, which have been scaled to a size of 227×227 , to extract feature vectors from each input image. One class trains an SVM classifier and a regressor to categorize the feature vector and enhance the candidate frame [28].

There are still several obvious limitations despite the fact that the RCNN algorithm's performance has greatly increased when compared to the traditional target detection technique [29–31]. For instance, the number of candidate regions that overlap and have significant redundant feature calculations leads to extremely sluggish detection speeds and significant disk usage. After that, the suggestion made by He Kaiming et al SPPNet. was successful in resolving this issue. A spatial pyramid pooling structure that can accept input of any size and output a fixed-size feature vector without the need for additional processing on candidate regions is the main innovation included in the system. zoom. Because it only needs to calculate the feature map of the entire picture once and avoids duplicate feature calculation, the technique is more than 20 times faster than RCNN without sacrificing detection accuracy. SPPNet does improve detection speed, but it still suffers from multi-stage training and simply fine-tunes the entirely linked layers without taking the preceding layers into account [32].

2.2.2 Detection algorithm based on anchor-free

Anchor-free detection, in contrast to anchor-based detection, completes detection by foreseeing the target center and four borders. Either guessing the pixel's distance from the target center or predicting a centerness score can be used to predict the target center. By estimating the distance between each pixel and the four limits of the real frame, the target borders are predicted. Depending on how objects are represented, different anchor-free approaches can be divided into algorithms based on joint detection of several key points and algorithms based on single center point regression.

The refining method used by this algorithm, called key point identification, improves model correctness and provides ideas for a later anchor-free technique. The YOLOv1, which was subsequently proposed, is also a pioneering study of anchor-free methodologies. In UnitBox [33], a DenseBox-like architecture, the L2 loss in DenseBox is optimized as an IoU loss. These three provide the foundation for the later success of the anchor-free method.

A popular method based on multi-keypoint joint detection is CornerNet [34], which uses two serially coupled Hourglass modules to extract features and two branches to forecast heatmaps, embedding vectors, and offsets. The upper left and lower right corners of the same target are paired using

the embedding vector, and the corners' precise locations are adjusted using the offset. The corner information of the positioning box is output using the heat map.

The main innovation of this method is the creative use of two edge box corners to detect objects. To more precisely localize the corners, a novel pooling layer is suggested. Additionally, the complete network is taught from scratch, liberating it from the limitations of earlier trained models. The subsequently proposed CornerNet -Lite[35] combines CornerNet-Saccade and CornerNet-Squeeze. The attention map is initially used by the former once it has predicted one using the reduced input image and a rough bounding box. Look for a suitable-sized foreground area to remove for the second stage of refinement, similar to the RPN concept. To create a simple feature extraction network, RPN essentially incorporates the ideas of SqueezeNet and Mobilenet into the Hourglass design.

Another branch of the Anchor-free approach is built on single center point regression. More representative is the one offered by Berkeley University, CenterNet [36] (same name as before). They claim that the bulk of target detection systems are built around anchor box principles. It is incredibly inefficient and requires extra post-processing to count potential locations for the target before classifying and regressing those locations (NMS suppression). CenterNet advises using the heat map to predict the target's center point and using that point as a benchmark to regress the width and height of the edge box as well as the offset of the point due to downsampling. This is shown in Figure 2-6. This network, which is very scalable, can identify 3D targets and estimate human poses by regressing additional target variables including 3D location, orientation, and attitude.

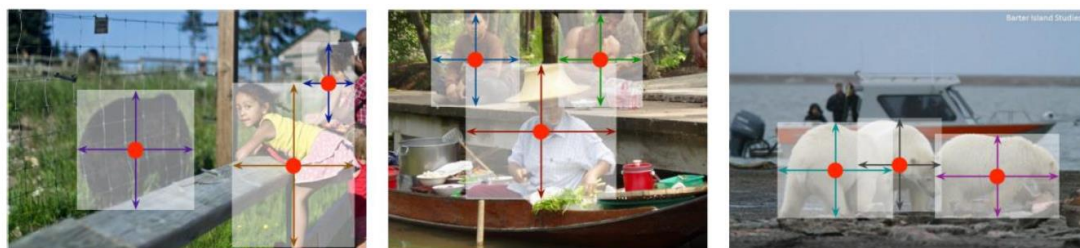


Figure 2-6 Returning to other attributes of the target with the center point

2.2.3 Fusion of anchor-based and anchor-free detection algorithms

The anchor-based technique has some drawbacks, including issues with the hyperparameter setup of the IoU threshold, the scale and aspect ratio of the anchor frame, which both require strong priors, and others. The Anchor-free method saves numerous hyperparameters as compared to the anchor-based method, and its advantage is a broader, more flexible solution space, which moves the detection algorithm closer to real-time high precision. The anchor-based method includes artificial prior distribution, and its network is simpler to train and more stable. However, because to its large solution space, it will also have a detrimental impact on unstable detection results, i.e., too many false positives (FalsePositive). As a result, some studies have suggested brand-new ideas for detection algorithms that incorporate these two approaches.

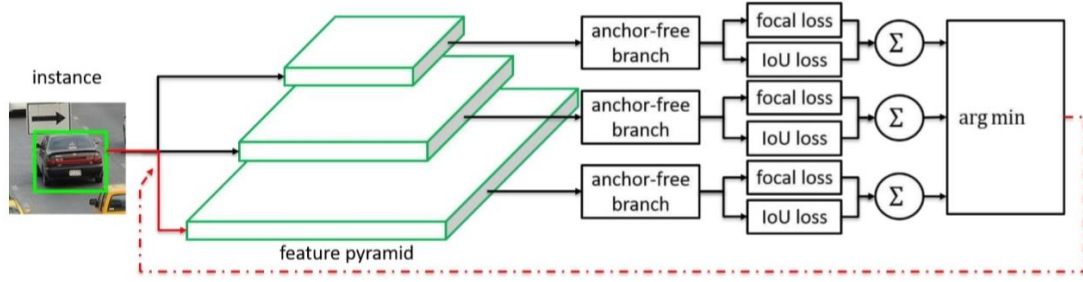


Figure 2-7 FSAF network structure

The anchor-based method, according to FSAF [37], has two drawbacks: guided feature selection and overlapping anchor box sampling. It was suggested to add an easy-to-use module for anchor-free feature selection (the FSAF module) at each stage of the feature pyramid, as depicted in Figure 2-7. The autonomous allocation of ground truth boxes in the feature pyramid is achieved by calculating each ground truth box's loss in the FSAF modules of the various layers and allocating it to the feature layer with the smallest loss. The detection is then completed using the anchor-based module. The anchor-free and anchor-based branches are weighted for training during the training phase, and NMS processes both branches' prediction boxes jointly during the inference phase.

GA-RPN [38] also offers to use characteristics to direct the construction of anchor boxes and blends the concepts of anchor-based and anchor-free algorithms. First, two branches are introduced using the feature pyramid as a foundation. One branch is used to forecast the location that could be the anchor frame's center, while the other branch forecasts the dimensions (length and breadth) of the anchor frame before forecasting the convolutional outcome based on those dimensions. Obtain the offset, modify the feature map using deformable convolution, and then finish the classification and regression using the modified feature map.

2.3 Summary of this chapter

The fundamental concept, constituent parts, forward and reverse propagation processes, and theoretical explanation of convolutional neural networks' operational principle are all covered in this chapter. The theory and development of the target detection technique for convolutional neural networks are then thoroughly explained. A wide range of good target identification algorithms from the anchor-based and anchor-free branches are introduced, together with their innovations or fundamental operating principles. The strengthening of the network structure later in this study lays the groundwork.

3 Mask wearing detection method based on YOLOv3

The YOLO series is a typical example of single-stage detectors since it is the most applicable algorithm in the current target detection area. Five different iterations of the YOLOv1 have been proposed since the series' inception. The most recent YOLOv4 and YOLOv5 were both recently suggested, and while their performance is quite near and their network structures are almost identical, they both use some SOTA methods in the area of target detection to integrate and enhance YOLOv3. However, YOLOv3 has broad industry use due to its easy deployment, straightforward structure, high detection accuracy, and speed; hence, this paper choose YOLOv3 as the primary detection network for mask wearing detection.

3.1 The overall network structure of YOLOv3

Darknet-53 is the feature extraction network used by YOLOv3, which fuses feature layers from different scales using a feature pyramid topology. Figure 3-1 depicts the structure of the network as a whole.

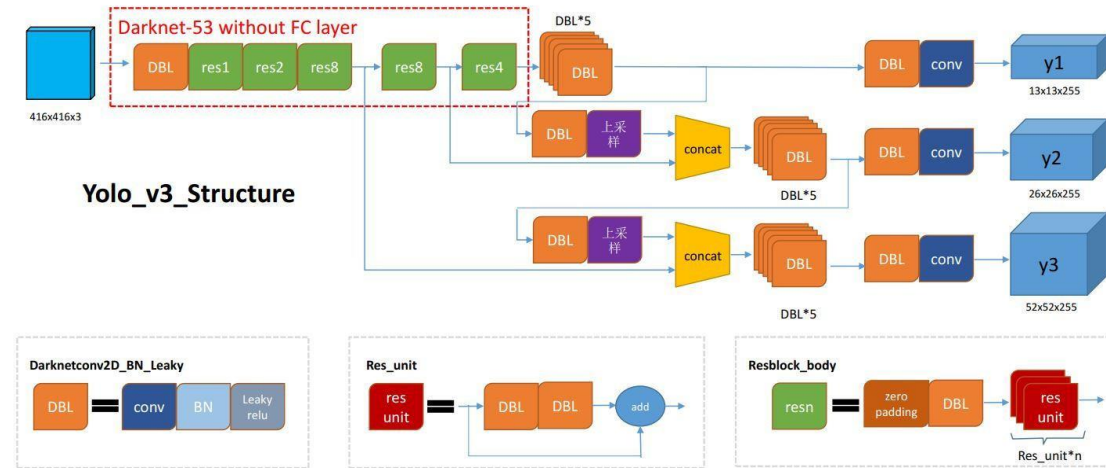


Figure 3-1 YOLOv3 network structure

3.1.1 Feature extraction network Darknet-53

Typically, the necessary feature information must be extracted first, and then these characteristics are employed to perform localisation and categorization of objects from photos or videos. Before deep learning became popular, features like HOG features, SURF features, and SIFT features had to often be extracted manually. Traditional manual techniques have gradually been superseded by neural networks' quick development. It is referred to as a feature extraction network because it applies the classification network's potent feature extraction capabilities to target detection through transfer learning and acts as a feature extractor (backbone).

The feature extraction network used by YOLOv3 is the fully convolutional network Darknet-53, and Figure 3-2 depicts its network architecture. Based on this, YOLOv3 eliminates the classification layer and final global average pooling layer for fine-tuning. In the complete network topology of Darknet-53, the feature map directly modifies the step size of the convolution kernel rather than using the pooling layer to limit the amount of feature information lost due to pooling. A value of 2 corresponds to a feature map with a resolution that is one-fourth of its original size. Convolution, batch normalization, and the LeakyRelu activation function are still used in the structure of the convolution component of YOLOv2. Darknet-53 features a total of 5 convolutional layers with a stride of 2, as seen in Figure 3-2. With an input resolution of 416*416 and an output resolution of 13*13, the feature map is comparable to five downsampling. Additionally, Darknet-53 adapts ResNet's [39] concept to incorporate a significant number of residual connections, which effectively reduces gradient explosion or disappearance during backpropagation while deepening the network.

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

Figure 3-2 Darknet-53 network structure

Target detection tasks in complicated scenarios are well suited for Darknet-53 since it has strong feature extraction capabilities, achieves classification accuracy on the ImageNet dataset comparable to ResNet-152, and is twice as quick as ResNet-152.

3.1.2 Feature pyramid structure

The high-level feature map has a large receptive field, the detailed information of the target is relatively rich, and the semantic information is relatively rich. In the feature extraction process of the convolutional neural network, the low-level feature map has a small receptive field, the detailed information of the target is relatively rich, but the semantic information is less. Because of this, the feature map of the upper layers of the network is typically better at recognizing larger objects than it is for detecting smaller ones, such as P4 in Figure 3-3. For this reason, YOLOv2 is not effective at detecting small targets. Some small-sized items are likely to vanish from the feature map and become undetectable when the convolutional neural network downsamples a specific number of times. The emergence of the feature pyramid significantly reduces this issue. The network's detection performance is further enhanced by adding top-down pathways and lateral connections to combine rich high-level semantic information with rich low-level detailed information. YOLOv3 makes use of this framework. The rich semantic information of the high-level (P4) is fused into the low-level (P3, P2) through the pyramid structure by combining the features of the three scales, which increases the detection precision on small targets.

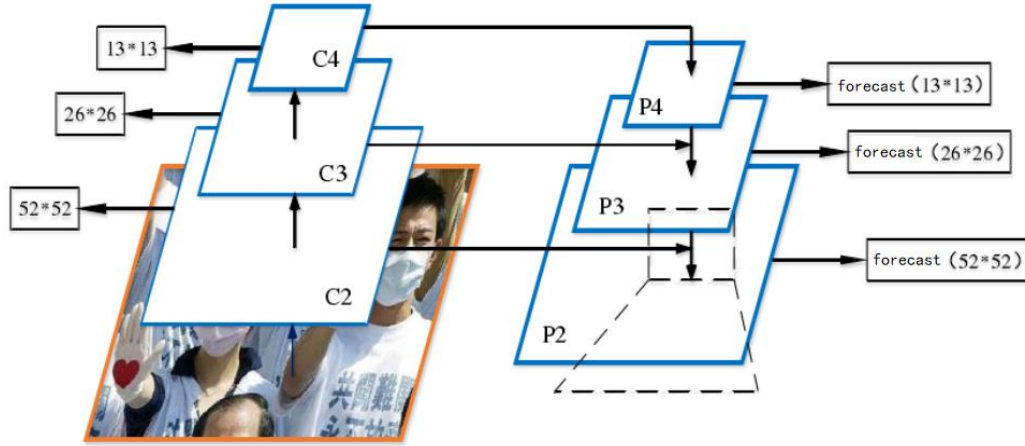


Figure 3-3 Feature pyramid structure for multi-scale prediction

As shown in Figure 3-3, this study predicts using the feature maps of three scales: 13×13 , 26×26 , and 52×52 . When performing horizontal image fusion, such as the fusion of P4 and C3, P4 is double-upsampled before being combined with C3 for channel merging. This replaces the element-by-element addition process used in the original feature pyramid, and the upsampling method makes use of nearest neighbor interpolation.

3.1.3 Multi-scale detection layer

The principle behind YOLOv3 for mask wearing detection is to first generate a feature map of a specific scale, like 13×13 , through the feature extraction network. Next, divide the image into 13×13 grids, and then determine which grid the target's center coordinates fall on. This target's prediction is handled by the grid. Each grid has three anchor boxes of varying sizes, and each anchor box predicts five values (the center point, width and height of the edge box, and confidence), as well as two category values (with masks and without masks), so each grid requires 21 convolution kernels. The dimensions of the three detection layers' outputs are $13 \times 13 \times 21$, $26 \times 26 \times 21$, and $52 \times 52 \times 21$, respectively.

The logistic regression layer decodes the category's prediction technique, and in YOLOv2, the Sigmoid activation function takes the role of the Softmax. The edge box's coordinate prediction method, which adopts the YOLOv2 methodology, accurately predicts the offset with respect to the grid's upper left corner.

3.2 Anchor frame mechanism

3.2.1 Position prediction based on anchor box

The prediction method of the edge box is the method of predicting the coordinate position relative to the grid. As shown in Figure 3-4, for each edge box, the model predicts 5 values, namely t_x , t_y , t_w , t_h and t_o , which are t_o related to the predicted value of the confidence, t_w and t_h is the predicted width and height offset, t_x and t_y is the predicted offset relative to the coordinate of the upper left corner of the grid. The sigmoid activation function limits it to the current grid by normalizing it to the output interval of $[0, 1]$, as else the center of the edge box would be seen in the image. Any location of, is not favorable for the model's convergence.

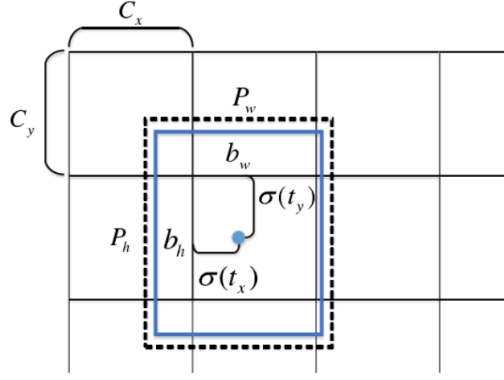


Figure 3-4 Position prediction of edge boxes

The target center is positioned in the upper left corner of the image, and in Figure 3-4, C_x and C_y show the ordinate and abscissa distances from the grid where it is located. The anchor box, P_w and P_h is the set anchor box size that is mapped to the width and height on the feature map. The blue solid line box is the edge box projected by the model, and its predicted value is calculated as follows:

$$b_x = \sigma(t_x) + C_x$$

$$b_y = \sigma(t_y) + C_y$$

$$b_w = P_w e^{t_w}$$

$$b_h = P_h e^{t_h}$$

$$\sigma(t_0) = \Pr(object) \times IOU(b, object)$$

where σ is the sigmoid activation function, b_x and b_y is the center coordinate of the predicted edge box, b_w and b_h is the width and height of the border box, $\sigma(t_0)$ is the predicted value of confidence, $\Pr(object)$ is the probability that the prediction box has a target, $IOU(b, object)$ is the intersection ratio of the predicted box and the ground-truth box. The loss of the prediction frame and the real label frame is determined based on the obtained parameters, and the loss is then minimized using the gradient descent method, causing the prediction frame to gradually approach the real frame.

3.2.2 Size priors for anchor boxes

Three feature map scales are utilized by YOLOv3 for detection. There are a total of nine different anchor box sizes that must be configured, three of which must be set for each feature map. When the network is launched, if it can choose an anchor box that is the right size, it will undoubtedly be able to anticipate outcomes more accurately and quickly. The nine types of annotation boxes with the highest frequency are utilized as the anchor box size before to speed the convergence of the network in YOLOv3 as they are clustered using the K-means technique. Since the error will rise with the size of the rectangular box, the traditional K-means algorithm's use of Euclidean distance to compare data objects is evidently inappropriate for determining how different two rectangular boxes are from one another. In order to redefine the distance function, IoU is used:

$$d(box, centroid) = 1 - IOU(box, centroid)$$

In the equation, centroid is the location of the cluster, box denotes each actual labeled box in the training set, and IoU denotes the ratio of intersection between the two rectangular boxes. The K-means algorithm's primary steps are:

(1) Initialize the number of categories and cluster centers, where the category is 9 categories, and the cluster center is the width and height of the real label box (W_i, H_i);

(2) Determine the separation between all cluster centers and each actual label box, then assign each actual label box to the nearest cluster center;

(3) After assigning all the annotation boxes , recalculate the cluster center (W_i^* , H_i^*) of each category cluster;

$$W_i^* = \frac{1}{N_i} \sum w_i$$

$$H_i^* = \frac{1}{N_i} \sum h_i$$

(4) Repeat steps (2) and (3) until the center position of the category cluster no longer changes.

Three feature layers based on scale are used to separate the nine distinct types of anchor boxes that were obtained by clustering. Because the low-resolution feature map's associated receptive field is larger than the high-resolution feature map's, the anchor box size mapped to the original image is larger for the low-resolution feature map. The figure's anchor box is a modest rectangle in size. As a result, the three 52*52, 26*26, and 13*13 feature maps are given, in increasing order, nine different types of anchor boxes. These nine distinct types of anchor boxes in YOLOv3 have default sizes of 10*13, 16*30, 32*23, 30*61, 62*45, 59*119, 116*90, 156*198, and 373*326. This is in accordance with COCO. The size of the anchor box used in this paper has to be re-clustered on the self-created data set.

3.3 Design of Loss Function

The loss function is a common tool in machine learning and control theory. Its purpose is to assess the effectiveness of the model and, by minimizing its value, to improve the model's parameters. The loss function, a crucial component of deep learning, directs the network parameters to iterate in the direction of optimization during the network training process, enabling the model to produce the best predictions possible.

3.4 Dataset Construction and Labeling

Three publicly available standard datasets, including 5445 photos from the WIDERFace [40] dataset, 4055 images from the MAFA [41] dataset, and 1829 images from the RMFD [42] dataset, are included in the 11329 image files of the collection, which are in JPG format. The photos' sizes vary, as seen in Figure 3-5.



(a)



(b)

Figure 3-5 Dataset image

In order to increase the robustness of the model, the data set in this paper also includes some images of the mouth and face covered by hands or other objects. This prevents the model from incorrectly identifying the mask as being worn when the face is covered.

3.4.1 Annotation files

The dataset's annotation, which includes information about the name, width, and height of the image, is in the VOC format. The location of the object's edge frame and the object's category are the two primary components of the annotation information. Faces with masks and faces without masks are the two sorts of object categories in this study, and they are denoted by the symbols "face" and "face mask," respectively. The coordinates of the upper left and bottom right points show where the edge box is located.

MAFA and WIDERFace. In this study, the annotation file format is changed to VOC format via script, and the annotation files of various photos are modified to achieve the most accurate and consistent edge box annotation. Since the RMFD image does not include a corresponding label file, this article labels the image using the expert labeling program LabelImg. In Figure 3-6, the labeling procedure is displayed. You must first import the image into the program. After that, you must add category information to the rectangular box you just created, save the image, and create an XML file.

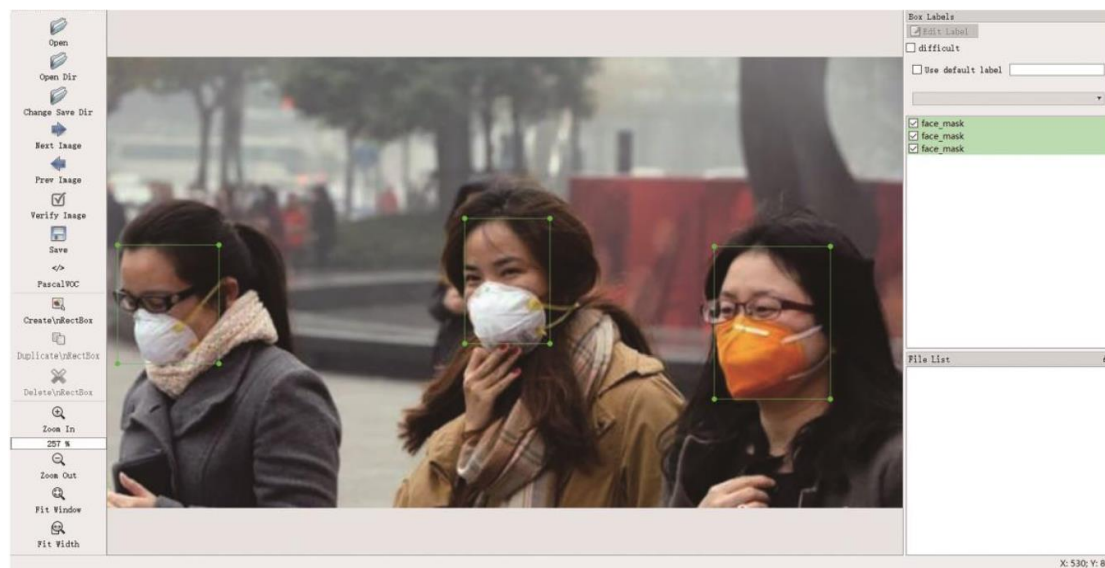


Figure 3-6 Labeling images with LabelImg software

3.5 Experimental results

3.5.1 Experimental platform and parameter settings

The following describes the hardware platform setup used in this experiment: GPU acceleration library is CUDA10. 2 and the matching version of CUDNN. CPU is Intel Core i7-9750H processor, 2.60GHz main frequency, 16GB memory; graphics card is NVIDIA GeForce GTX1660TiwithMax-Q graphics card, 6G video memory; operating system is Windows 1064-bit system. The experiment's software platform is the C++-compiled Darknet deep learning framework.

The data set is partitioned into five mutually exclusive subsets, each of which is used as the test set. The remaining four subsets are utilized as the training set throughout the procedure, which is repeated five times. This approach is known as five-fold cross-validation. The average of the five training results is ultimately employed as the model's final indicator for training with various data distributions. The training optimizer uses the Nesterov Accelerated Gradient (NAG) optimizer, the momentum is set to 0.9, the weight decay coefficient is set to 0.0005, and each batch of input images (BatchSize) is set to be 4. The initial learning rate is set to 0.001 and the learning rate adjustment strategy adopts a fixed-step decay. The model is trained using migration learning of the pre-trained model, with 8000 iterations.

3.5.2 Evaluation Index of Detection Algorithm

Target detection algorithms frequently gauge the detection accuracy using mean average precision, or mAP. AP (AveragePrecision) is the region contained by the curve and the coordinate axis drawn by the precision and recall of each category in the prediction result. mAP is the average value of AP (AveragePrecision) of all categories in the test set. recall, accuracy, and AP.

Representation of common indicators		predicted state	
		positive (the prediction result is true)	negative (the prediction result is false)
real	correct (positive example)	TP	FN
	error (counter example)	FP	TN

TP stands for TruePositives, which guarantees that positive samples are recognized as such; FalsePositives, or FP, refers to instances in which negative samples are mistakenly thought to be positive ones; FalseNegatives stands for positive samples that are mistakenly labeled as negative ones; P stands for precision, and R for recall. This work also introduces the F1 metric, the harmonic mean of precision and recall, to more thoroughly assess the performance of various models.

FPS, or the number of frames of images that can be processed per second, is another crucial indicator of the target detection method in addition to the detection accuracy. For the sake of fairness, the FPS of several models must run on the same hardware because the processing speeds of various pieces of hardware vary.

3.5.3 K-means clustering results

YOLOv3's default anchor box size is modified based on the produced dataset, and the dataset is clustered using the K-means technique to determine the size priors of nine different types of anchor boxes. Each image in the collection varies in size, so it makes sense that some annotation boxes have fairly varying sizes. Therefore, this work normalizes the label box's height and breadth before preparing the input data for clustering. Figure 3-7 displays the normalized visualization findings, and Figure 3-8 displays the final clustering visualization results. Show.

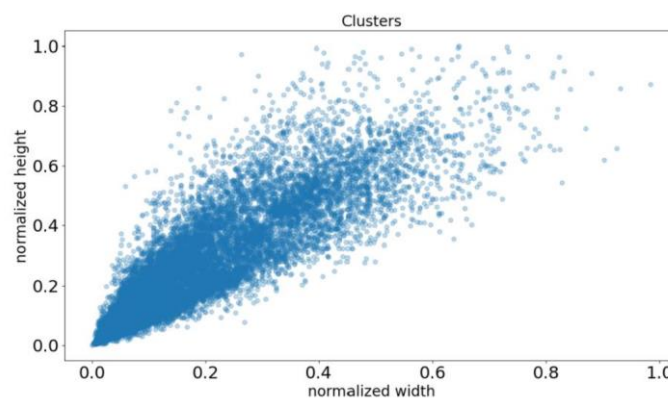


Figure 3-7 The normalized result of the width and height of the callout box

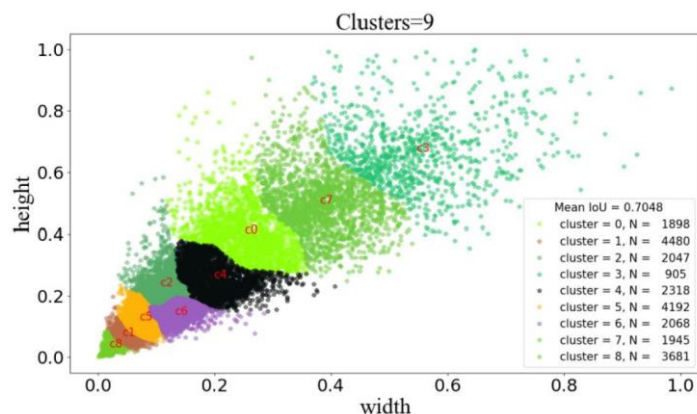


Figure 3-8 Clustering results of callout boxes

The nine cluster centers discovered through the clustering experiment are arranged from small to large as (0.019, 0.033), (0.042, 0.070), (0.071, 0.120), (0.132, 0.139), (0.107, 0.231), (0.199, 0.258), (0.252, 0.403), (0.381, 0.498), and (0.546, 0.667), and are then mapped to the resolution of 416 and rounded to obtain the size of the anchor box as (8 The first three are given 52*52 feature layers, the middle three are given 26*26 feature layers, and the last three are given 13*13 feature layers. The largest order is given to three feature layers.

3.5.4 YOLOv3 Model Evaluation Results

The nine cluster centers discovered through the clustering experiment are arranged from small to large as (0.019, 0.033), (0.042, 0.070), (0.071, 0.120), (0.132, 0.139), (0.107, 0.231), (0.199, 0.258), (0.252, 0.403), (0.381, 0.498), and (0.546, 0.667), and are then mapped to the resolution of 416 and rounded to obtain the size of the anchor box as (8 The first three are given 52*52 feature layers, the middle three are given 26*26 feature layers, and the last three are given 13*13 feature layers. The largest order is given to three feature layers.

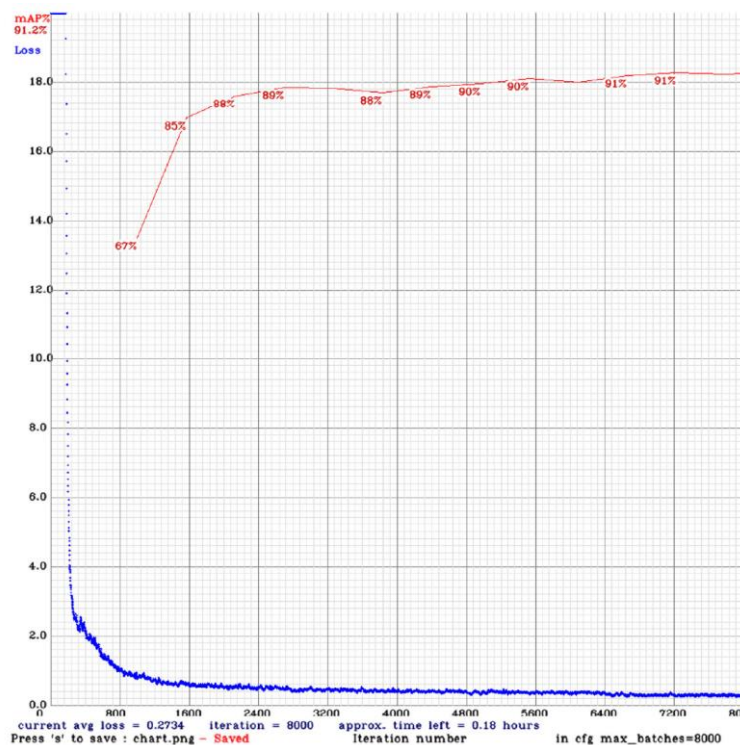


Figure 3-9 YOLOv3 loss convergence

This paper also tests on mAP@0.75 in order to more accurately gauge the model's detection accuracy. The performance of the YOLOv3 model on mAP@0.5 and mAP@0.75 is displayed in Table 3-1. The terms "mAP@0.5" and "mAP@0.75" above refer to the mAP indicators when the IoU threshold is 0.5" and "0.75", respectively, meaning that the prediction is deemed accurate when the intersection ratio of the predicted frame and the real frame hits 0.5" or 0.75".

Table 3-1 The detection results of YOLOv3 on mAP@0.5 and mAP@0.75

Index	Test 1	Test 2	Test 3	Test 4	Test5	Average
mAP@0.5 (%)	90.93	90.34	90.22	91.38	91.79	90.93
mAP@0.75 (%)	67.02	66.79	67.16	67.73	66.47	67.03

The dataset was divided into five different groups for the five-fold cross-validation (Table 3-1, Tests 1 through 5). On mAP@0.5 and mAP@0.75 respectively, it can be noted that YOLOv3 has detection accuracy of roughly 91 percent and 67 percent.

3.6 Chapter Summary

This chapter focuses mostly on the mask wear detection capability of the YOLOv3 algorithm. The foundation of the YOLOv3 algorithm, the network structure's design and principle, the anchor frame mechanism, and the loss function are first discussed. The approach used to label the dataset for this study, as well as how it was created, are then detailed. Finally, a description of the experimental setup for this investigation is provided. Through the configuration and evaluation index of the target detection method and the five-fold cross-validation model training approach, the employment of the YOLOv3 algorithm in the identification task for mask wearers has been preliminary realized. The testing results show that the initial YOLOv3 detection speed is only 10.62 frames per second, which is not enough to meet the requirements of real-time detection on the low-performance PC side. This makes way for the description of the YOLOv3 network structure in the following chapter.

4 Optimization of Network Training Process

The network's parameter optimization depends heavily on the loss function. In the first YOLOv3 training procedure, the loss of the edge box regression is calculated along with the loss of the edge box's width and height. It also includes the loss of the edge box's upper left corner. the first It uses binary cross-entropy, whereas the latter uses mean square error. The IoU indication is employed in the evaluation process to determine whether the target has been discovered, despite the fact that the two are obviously not equal. Determining this portion of the loss based on IoU during the training process is what this research proposes to do to optimize the loss of edge box regression.

The easiest and most straightforward way to enhance deep learning model performance is through data supplementation. Its main goal is to increase data by introducing prior knowledge, enhancing the model's capacity for generalization. The performance of the mask wearing detection model is further enhanced by this paper by introducing and contrasting the two data augmentation techniques Mixup [43] and Mosaic [44].

4.1 Optimization method for loss of edge box regression

In object detection algorithms, L norm (L1 and L2) is frequently used to determine the loss of edge box regression. The L1 norm loss function is called the absolute value loss function (LAE), which is y_i the sum of the absolute difference between the true value and the predicted value $f(x_i)$, and its

expression is as follows:

$$L = \sum_{i=1}^n |y_i - f(x_i)|$$

The difference between the true value and the predicted value, expressed as the sum of the squares of the L2 norm loss function, is known as the least squared error (LSE). It is $f(x_i)$ generally used for regression tasks and is expressed as follows:

$$L = \sum_{i=1}^n (y_i - f(x_i))^2$$

In contrast to the L2 norm loss function, which is more sensitive to data with outliers, the L1 norm loss function may obviously successfully disregard outliers in the data. The square will exacerbate the inaccuracy if it is greater than 1. The L1 norm loss function is therefore comparatively more robust. They both have issues when L1 norm and L2 norm are used to calculate the loss for edge box regression. The bounding box's upper left corner's coordinates, together with its four length and width parameters, are directly regressed in the L-norm prediction. Since the L-norm is insensitive to object scale, it holds true even if the scales of the edge box and the target box differ by an equal amount. The number's value can differ significantly. It makes more sense to utilize IoU as the regression loss benchmark in this situation. The intersection ratio between the actual label frame and the prediction frame is known as IoU. As seen in Figure 4-1, it is a scale-invariant indicator that reflects the rate at which the two rectangular frames overlap.

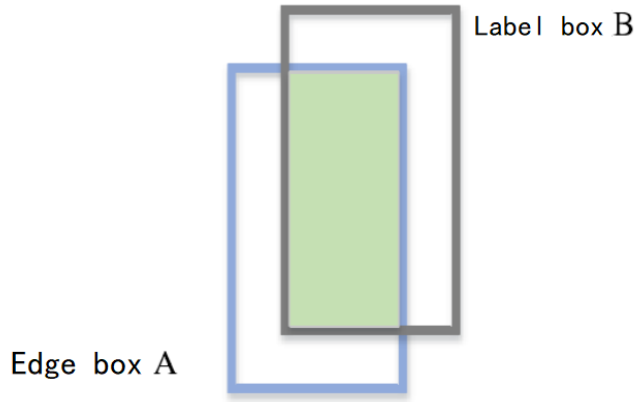


Figure 4-1 Cross-Union Ratio

Figure 4-1 shows a significant overlap between the real label frame and the predicted edge frame. IoU is calculated as the intersection of the union of two rectangular frames divided by their intersection area.

4.1.1 Loss function GIoU

When utilizing IoU as the edge box regression, there are two issues: (2) IoU cannot distinguish between different methods, that is, when the IoU value is the same, the overlapping method of the prediction frame and the real annotation frame is different. (1) If the predicted box and the real annotation box have no overlapping area, the IoU value equal to 0 will cause the loss function to be unsteerable and cannot be optimized. The alignment method is obviously different, as shown in Figure 4-2, but the IoU value of the three overlapping cases is 0.33. The leftmost prediction box has the best regression effect, and the rightmost prediction box has the worst regression effect.

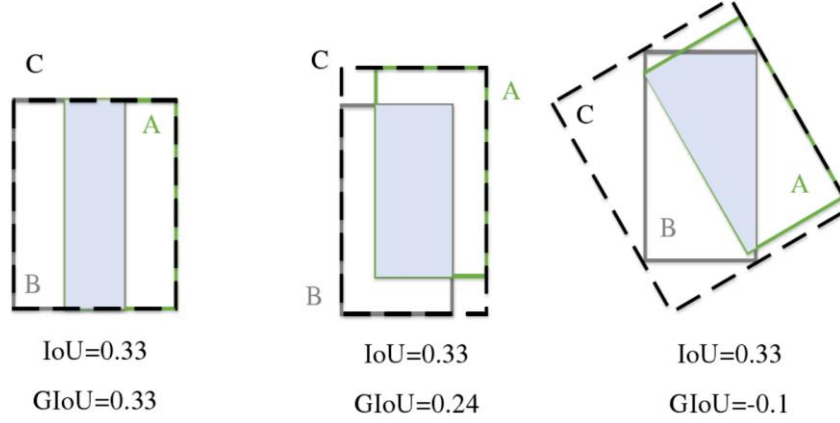


Figure 4-2 The same Io U value with different alignment

In order to accomplish edge box regression, HamidRezatofighi et al. presented a new loss function called GIoU [45] in answer to this issue. Figure 4-2 illustrates how the GIoU value differs when the IoU value is the same. The greater the GIoU score, the better GIoU is at differentiating alignments and the more accurate the alignment direction. In Figure 4-2, the leftmost edge box has the best regression effect and the highest GIoU value, whereas the rightmost edge box has the worst regression effect and the lowest GIoU value. Equation depicts the GIoU calculating process:

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$

In the formula, A and B stand in for the actual annotation and prediction frames, respectively, while C represents the minimal closed shape enclosing A and B. According to conventional wisdom, this means locating the smallest circumscribed rectangular frame around A and B. Like IoU, GIoU is not scale-sensitive. In other words, the value range of GIoU is $(-1, 1]$, which is highly suitable as a measure of distance. When A and B coincide, its value is 1, and when A and B have no intersection and are indefinitely far away, its value is infinitely close to -1. In contrast to IoU, GIoU considers both the overlapping and non-overlapping portions of the two rectangular boxes and provides good feedback on both. The loss function can be determined as follows:

$$L_{GIoU} = 1 - GIoU$$

4.1.2 Loss functions DIoU and CIoU

The goal of GIoU is to increase the IoU loss with a penalty term so that backpropagation can continue to optimize the parameters even when the predicted frame and the real frame do not overlap. In Equation, this penalty term is represented by the rightmost ratio:

$$L_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{GT}|}{|C|}$$

In the equation, B stands for the edge box, which, according to B-GT, is the actual annotation box, and C for B, the smallest rectangular box enclosing the total. The predicted frame will continue to approach the real frame even if the predicted edge frame does not overlap with it, although Zheng [46] and others think GIoU still has limitations. When the value of the penalty term is 0, the penalty term is no longer meaningful, and the entire loss function degenerates to the IoU loss. This occurs when both the edge frame and the real frame are covered. They therefore suggest the DIoU loss function, which is derived as follows and accounts for the overlap area and the center point distance between the forecast box and the ground-truth box:

$$L_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{GT})}{c^2}$$

In the formula, ρ^2 is the Euclidean distance between the center points of the two boxes, b and b^{GT} is the center point of the prediction box and the real box, respectively, c is the diagonal distance of the smallest rectangular box that surrounds the prediction box and the real box at the same time, as shown in Figure 5. As shown in -3, d in the figure is $\rho^2(b, b^{GT})$. It can be seen that the solution of DIoU is similar to that of GIoU, but the definition of the penalty term is different.

The Euclidean distance between the center point of the predicted box and the actual ground truth as well as the diagonal distance of the bounding box are introduced by the DIoU penalty term. This method can converge quicker than GIoU when the projected frame does not overlap the actual frame since it directly minimizes the distance between the two frames as well as helps the predicted frame approach the ideal direction.

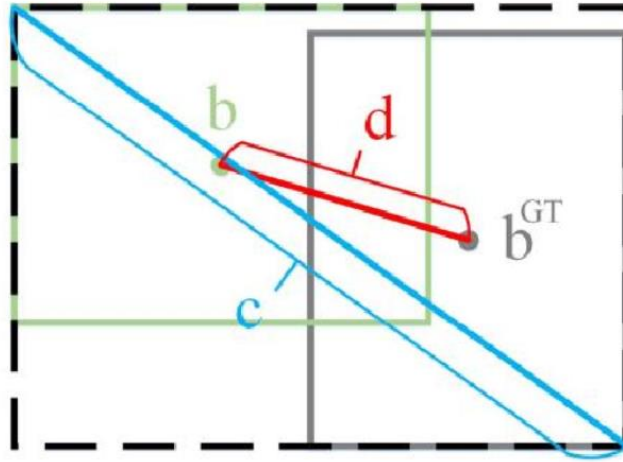


Figure 4-3 Schematic diagram of DIoU loss calculation

In order to improve the regression effect of the edge box, DIoU inserts the distance between the center points of the two boxes on the basis of IoU and modifies the penalty term. However, the aspect ratio difference also has a significant bearing on the regression, so Zheng is in DIoU's. The following CIoU is proposed on the basis of include the aspect ratio element in the punishment term:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{GT})}{c^2} + \alpha v$$

In the formula, v used to measure the consistency of the aspect ratio, it α is a weight coefficient, and the calculation method is as follows:

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w^{GT}}{h^{GT}} - \arctan \frac{w}{h} \right)^2$$

$$\alpha = \frac{v}{(1 - IoU) + v}$$

In the formula, w^{GT} and h^{GT} are the width and height of the real box, respectively, and w and h are the width and height of the prediction box. It can be seen that the gradient of CIoU loss is very close to DIoU loss, and only has more gradients than v DIoU.

When the center points of the two boxes coincide, the Euclidean distance between the two center points is 0, and the penalty term is removed, even if the DIoU and CIoU loss functions still exhibit strong adaptability when the edge box and the actual box overlap. In other words, the loss function will once more degenerate into the IoU loss. The optimization capabilities of DIoU and CIoU are therefore worse

to GIoU when the two frames overlap, but superior to GIoU when they do not. Each has benefits and drawbacks of its own. As a result, tests comparing these three loss functions are conducted in this study. The loss function for network training is chosen based on the performance.

4.2 Data Augmentation

The quantity of the data collection has always been better, but the expenditures associated with data acquisition and labeling are high. The goal of data augmentation is to increase the quantity of data available for network training without the need for new data collecting. Data augmentation can be separated into single-sample data augmentation and multi-sample data augmentation for the supervised learning in this study. Multi-sample data enhancement, which includes SMOTE[47], SamplePairing [48], CutMix, Mixup and Mosaic et al., refers to creating new samples around numerous samples. Single-sample data enhancement entails performing some geometric and color alterations around a sample itself. In order to further enhance the model's generalizability, comparison tests using the two approaches Mixup and Mosaic are conducted in this study.

4.2.1 Mixup data enhancement

A technique for reducing neighborhood risk is mixup data augmentation. New virtual samples are created for training by executing a straightforward weighted summing of two random samples and matching labels in the training set, after which the prediction results and weighted results are computed. Additionally, back-propagation is performed to update the parameters following the loss between the labels. This method of linear interpolation reduces computing cost while improving the model's ability to adapt to unknown samples. It is analogous to widening the training distribution. Mixup creating a virtual sample is described as follows:

$$\begin{aligned}\lambda &= \text{Beta}(\alpha, \beta) \\ \tilde{x} &= \lambda x_i + (1 - \lambda)x_j \\ \tilde{y} &= \lambda y_i + (1 - \lambda)y_j\end{aligned}$$

In the formula, λ is the parameter obeying the beta distribution, that is, the fusion ratio, which takes the value $[0, 1]$; x_i and x_j is the original input vector, y_i and y_j are the labels of the one-hot encoding format, and $\tilde{x}$ and $\tilde{y}$ are the new samples. Input vector and label.

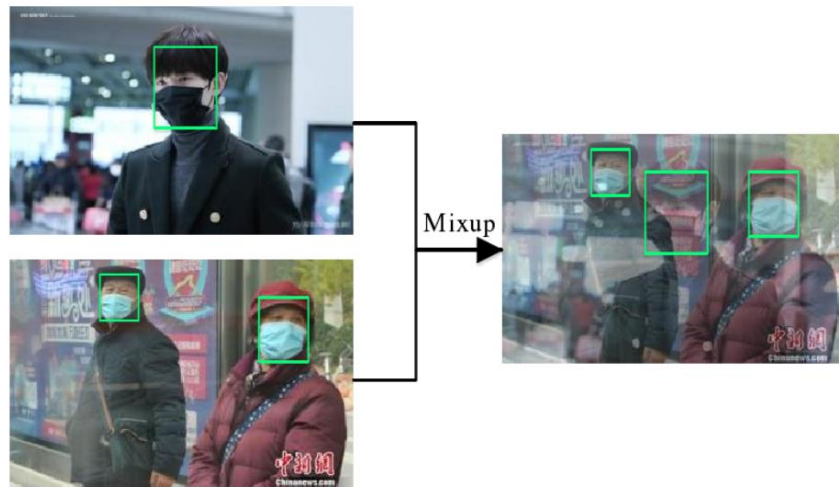


Figure 4-4 Mixup data enhancement

To put it simply, it entails fusing two images with labels in a specific ratio, as seen in Figure 4-4. This approach is analogous to employing discrete sample point continuity to fit the real sample

distribution, enhance the neighborhood's smoothness, and contribute to the model's resilience and stability.

4.2.2 Mosaic Data Augmentation

In YOLOv4, mosaic data improvement is suggested. It randomly scales and crops four photos before stitching them together to create a new image as training data, as illustrated in Figure 4-5. This approach differs from the general data enhancement method in that it does not conduct simple geometric transformation or color transformation on an image.

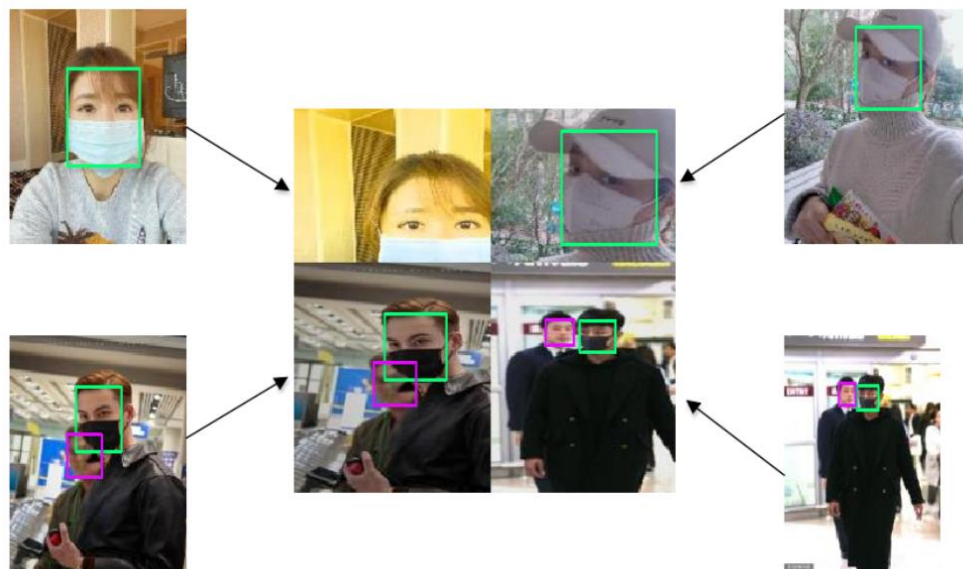


Figure 4-5 Mosaic data enhancement

The main steps in the Mosaic data enhancement process are as follows: (1) randomly read the four images in each batch; (2) flip, zoom, and other operations on the four images, respectively, and follow the placement of upper left, upper right, lower left, and lower right; (3) crop and stitch to create a new image for training, and only keep the entire annotation frame after cropping.

This data improvement technique has two benefits: (1) It significantly improves the background of the detected object, and in addition, the method of cropping and scaling reduces the scale of the object in disguise, balancing the distribution of objects of different scales in the training process, which is helpful to improve the detection performance of small targets; (2) The method of secretly splicing four pictures increases the number of input images per batch in the training process, so that the training hyperpaque is balanced.

4.3 Experimental results

4.3.1 Loss function comparison experimental results

For comparison tests, the loss function is optimized in this paper to GIoU, DIoU, and CIoU using MobileNetV2-SPP-YOLOv3-A as a foundation. Tables 4-1 and 4-2 display the five-fold cross-findings. validation's .

Table 4-1 Comparison test of GIoU , DIoU and CIoU (mAP @0.5)

Loss function	Test1 (%)	Test2 (%)	Test3 (%)	Test4 (%)	Test5 (%)	Average(%)
Raw _	8 9.55	8 8.75	8 8.34	8 9.17	8 9.47	8 9.06
G IoU	9 1.34	9 0.98	9 0.51	9 1.37	9 1.32	9 1.10

D ioU	9 1.56	9 0.93	9 0.39	9 1.30	9 1.41	9 1.12
C ioU	9 1.78	9 0.89	9 0.39	9 1.64	9 1.33	9 1.21

Table 4-2 Comparison test of GIoU , DIoU and CIoU (mAP @0.75)

Loss function	Test1 (%)	Test2 (%)	Test3 (%)	Test4(%)	Test5(%)	Average(%)
Raw _	6 1.95	6 0.10	5 9.91	5 7.94	6 0.33	6 0.05
G IoU	6 8.80	6 8.40	6 8.48	7 0.28	6 9.88	6 9.19
D ioU	6 8.91	6 8.54	6 8.88	6 9.32	6 8.98	6 8.90
C ioU	6 8.14	6 9.05	6 7.72	6 8.61	6 8.73	6 8.46

From Tables 4-1 and 4-2 that after the optimization of the loss function, the model has been significantly improved in mAP @ 0.5 and mAP @ 0.75, and GIoU , DIoU and CIoU on mAP @ 0.5 . The effect is very close, CIoU has the best effect, with an increase of 2.15%; mAP @ 0.75, the three improved effects are very significant, and GIoU has the best effect, with an increase of 9.14%. Comprehensive consideration, this paper selects GIoU as the final loss function, and names the model MobileNetV2-SPP-YOLOv3-A-GIoU.

4.3.2 Data enhancement comparison test results

In this paper , data enhancement is carried out on the basis of MobileNetV2-SPP-YOLOv3-A-GIoU, and comparison experiments are carried out in two ways: Mixup and Mosaic. The specific experimental results are shown in Tables 4-3 and 4-4.

Table 4-3 Comparison test of Mixup and Mosaic data enhancement (mAP@0.5)

Data augmentation	Test1 (%)	Test2 (%)	Test3 (%)	Test4 (%)	Test5 (%)	Average(%)
Not enhanced	9 1.34	90.87 _	9 0.51	9 1.37	9 1.32	9 1.10
Mixup _	9 1.40	9 1.14	9 0.73	9 1.56	9 1.76	9 1.32
Mosaic	9 1.89	9 1.56	9 0.82	9 1.68	9 1.72	9 1.54

Table 4-4 Comparison test of Mixup and Mosaic data enhancement (mAP@0.75)

Data augmentation	Test1 (%)	Test2 (%)	Test3 (%)	Test4 (%)	Test5 (%)	Average(%)
Not enhanced	6 8.80	6 8.50	6 8.48	7 0.28	6 9.87	6 9.19
Mixup _	6 8.66	68.67	69.28	69.01	69.21	68.97
Mosaic	69.70	70.25	69.19	70.45	70.74	70.07

Tables 4-3 and 4-4 show that on mAP@0.5, the detection performance was improved by both the Mixup and Mosaic methods, with the Mixup method improving by 0.22 percent and the Mosaic method improving by 0.44 percent. On mAP@0.75, the Mosaic method has significantly improved, by 0.88 percent, but the Mixup method will decrease the detection accuracy. This study selects the Mosaic approach as the data enhancement strategy for the final model after carefully considering it.

4.4 Chapter Summary

This chapter's major topic is network training process optimization, starting with loss function and data enhancement optimization. In the first place, examine the issues with the edge box regression loss in the original YOLOv3 and suggest optimizing the loss as a loss function connected to the intersection and union ratio. Explained and compared amongst studies, the convergence impact of the GIoU model

is the best, according to the experimental findings. Finally, this paper adopts a data augmentation strategy and conducts comparative tests using two great data augmentation methods in order to further improve the generalization ability of the model. According to the experimental findings, Mosaic data augmentation performs better on this dataset than Mixup. On $mAP@0.5$ and $mAP@0.75$, the generalization ability shows a very clear improvement effect.

5 Test Results and Evaluation of Mask Wearing Test

5.1 Performance evaluation of the model

This study optimizes the training procedure while reducing the weight of the network structure based on YOLOv3. The main changes to the network structure include the replacement of the feature extraction network with a condensed version of MobileNetV2, the addition of spatial pyramid pooling for feature fusion after the feature extraction network, and the replacement of the detection head's standard convolutional layer with a depthwise separable convolution. Training is also improved. The approach is primarily optimized by using the Mosaic method for data augmentation and optimizing the loss of edge box regression as GIoU. Figure 5-1 depicts the method's loss convergence impact.

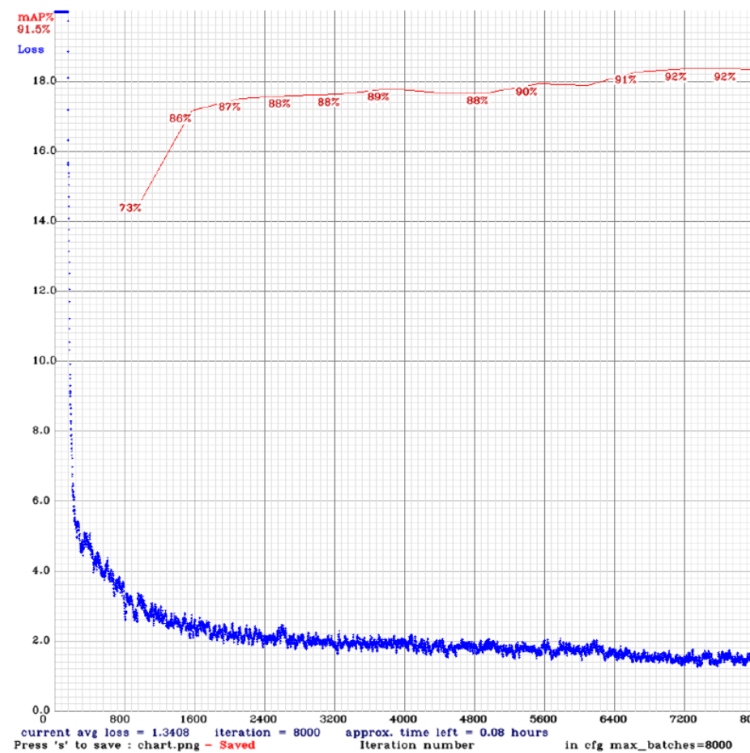
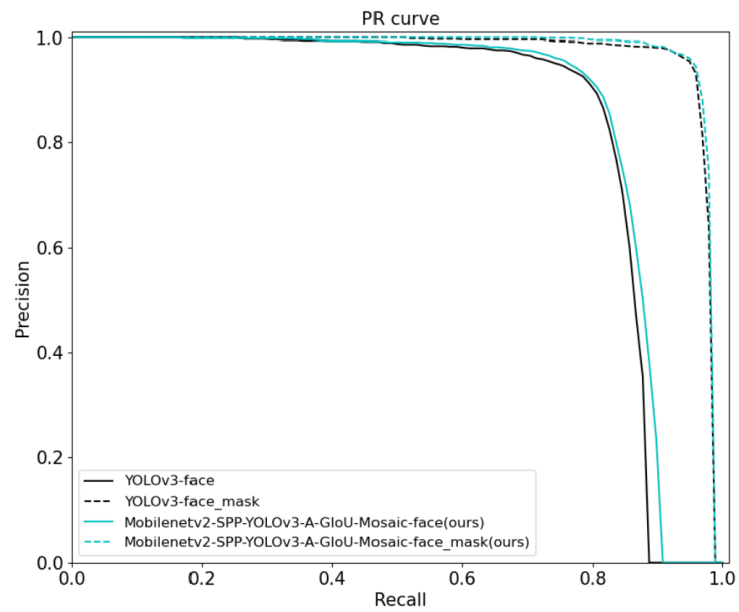


Figure 5-1 Loss convergence of our method

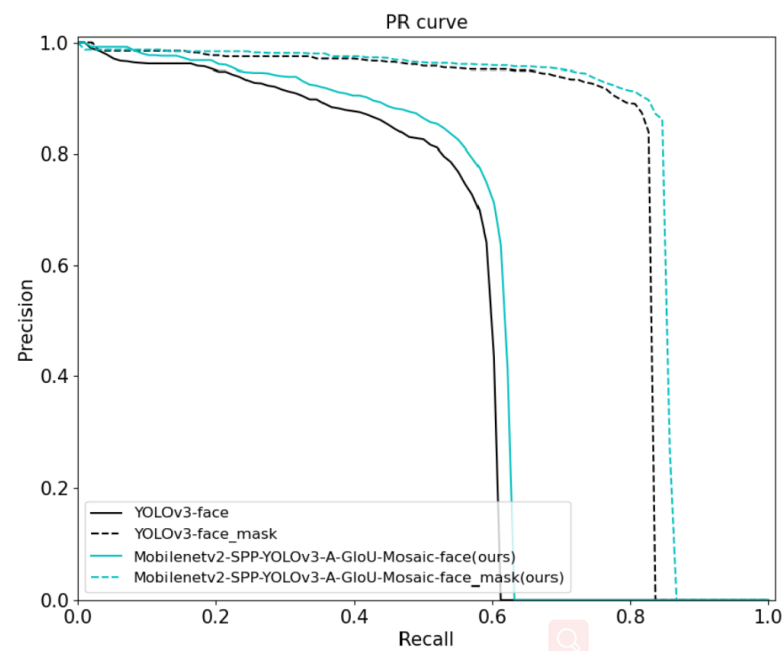
Figure 5-1 demonstrates how the method described in this research can cause the network to converge fast and steadily. The model's training time is cut in half when compared to the original YOLOv3, while the model's detection accuracy remains same.

The most logical representation of mAP, the PR curve can clearly demonstrate the accuracy and recall rate of the detector on the test samples. It is frequently used to assess the effectiveness of target detection algorithms. When contrasting the performances of two detection models, it may be said that the latter's performance is superior to the former if the PR curve of the former is entirely encircled by the PR curve of the later. The PR curve from this research is compared to that of YOLOv3 in order to further

demonstrate the superiority of the approach, as seen in Figure 5-2.



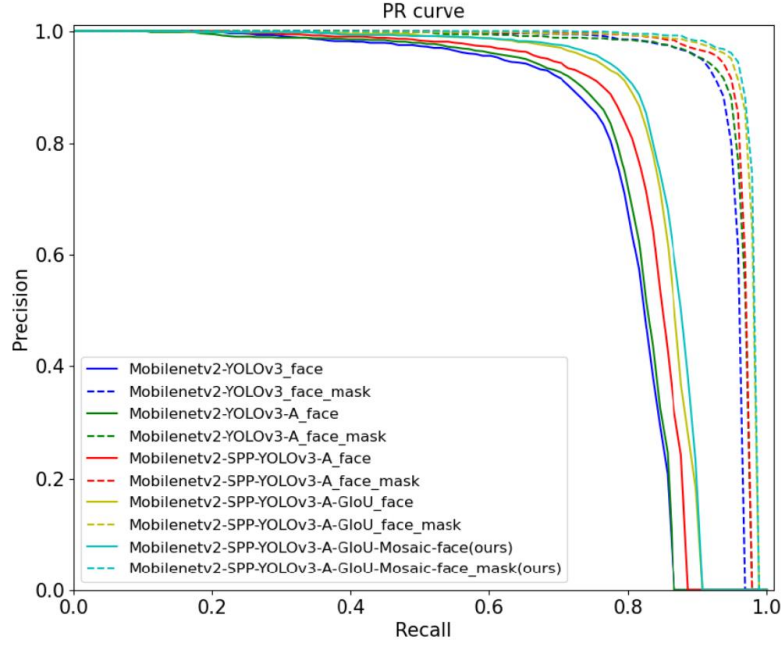
(a) $\text{IoU} = 0.5$



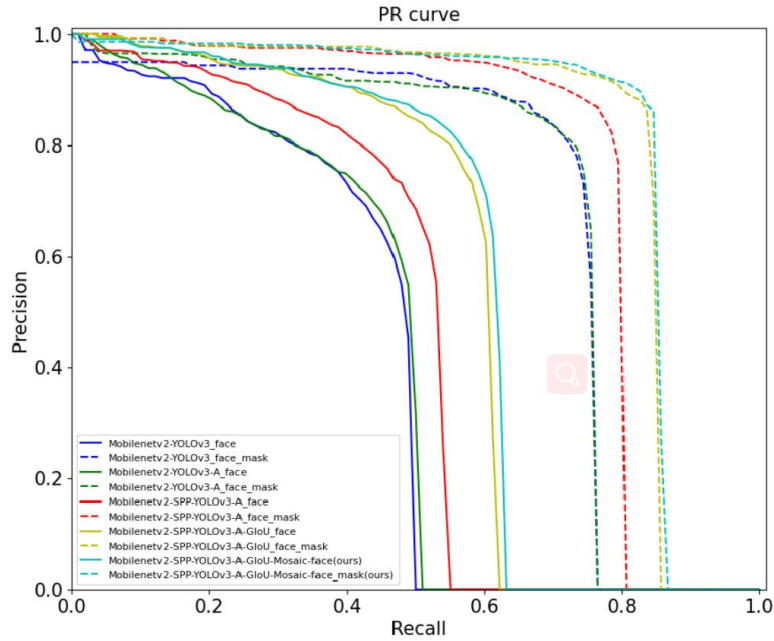
(b) $\text{IoU} = 0.75$

Figure 5-2 The PR curve comparison between the method in this paper and YOLOv3

Figure 5-2 illustrates how, on $\text{mAP}@0.5$, the PR curve of the method in this paper almost matches that of YOLOv3, particularly in the face detection (face mask) wearing a mask, but overall, the method in this paper has a slight advantage. On $\text{mAP}@0.75$, the method in this paper achieves the encirclement of YOLOv3, indicating that it is more accurate than the original YOLOv3 in terms of detection.



(a) IoU=0.5



(b) IoU = 0.75

Figure 5-3 Change of PR curve during model improvement

At the same time, the PR curve is used to convert the MobileNetV2-YOLOv3, MobileNetV2-YOLOv3-A, MobileNetV2-SPP-YOLOv3-A, and MobileNetV2-SPP-YOLOv3-A in the process of model improvement in order to quantify the influence of each change made in this study on the model detection accuracy. Figure 5-3 compares the final approach used in this paper, MobileNetV2-SPP-YOLOv3-A-GIoU-Mosaic, to -GIoU. The figure shows that almost all of the in this paper's improvements achieve envelopment of the prior detector, particularly on mAP @ 0.75. This work uses OpenCV to illustrate the detection results in order to more intuitively portray the detection effect of the model. As shown in Figure 5-4, the left column represents the detection effect of YOLOv3, and the right column represents the detection effect of this method. As can be shown, the method described in this

paper has a greater real detection effect when dealing with small objects or side faces than YOLOv3.



(a)

(b)



(c)

(d)



(e)

(f)



(g)

(h)

Figure 5-4 Comparison of the detection effect of this method and YOLOv3

In summary, it can be said that the method described in this research has a definite speed advantage over YOLOv3. The method used in this paper is quite similar to YOLOv3 and somewhat more accurate in terms of detection. In the case of side faces, the detecting effect is better.

5.2 Chapter Summary

This chapter primarily carries out a thorough analysis of the model's performance and experimental findings. First, this study compares the approach in many ways to YOLOv3. The comparison results demonstrate that the method in this paper not only significantly reduces the model's complexity, but also significantly increases the model's detection speed and also achieves a slight improvement in detection accuracy; secondly, this paper also addresses the issue of model complexity. The resilience of the model is tested through a series of experiments. The experimental findings demonstrate that the model presented in this study does not have evident overfitting and nevertheless performs well for brightness, scale change, and occlusion detection.

6 Summary and Outlook

6.1 Summary

The detection of mask wear is investigated in this work. An accurate, quick, and light-weight intelligent detection algorithm is proposed, which can quickly and effectively complete the detection of mask wearing in various scenarios, in light of the slow detection speed of common detection methods and the inability to realize real-time detection on low-performance PC terminals. The following features are present in the paper's primary work:

(1) This paper chooses YOLOv3, which has excellent accuracy and quick detection speed, as the main detection network in light of the enormous flow of persons wearing masks and the need for quick detection. This research chooses the lightweight network MobileNetV2 as the feature extraction network and streamlines the network structure to further minimize the network parameters in order to increase the detection speed of the network on the low-performance PC side. In addition, to further increase the network's detection speed, depthwise separable convolution is employed to solve the issue of the detection head part's massive conventional convolution calculation.

(2) This paper introduces a spatial pyramid pooling structure after the feature extraction network to address the issue of lower detection accuracy once the network is lightweight. The fusing of various receptive fields and the fusion of local features and global features are both accomplished by altering the size of the pooling kernel, which enhances the detection performance of the network without raising its computational cost.

(3) This research optimizes the training procedure to increase the detection accuracy of the network without raising the computational complexity of the network. First, the issue of employing the L norm for the loss of edge box regression is examined, and then a theoretical analysis of the loss function of the current edge box regression is done. The studies were compared, and it was discovered that GIoU can reflect the edge box better than other loss functions. Scale invariance exists in the relative position relationship with the ground-truth box. Finally, Mosaic is chosen for data improvement through comparative experiments in order to further boost the network's capacity for generalization. The experimental results demonstrate that, without increasing computational complexity, the optimization strategy described in this study considerably enhances the model's detection accuracy.

6.2 Outlook

Due to the short research period and the quick advancement of deep learning-related technologies, this issue has achieved some results in the research on the method of mask wearing detection, but there are still certain weaknesses and areas where the research process can be strengthened.

(1) The dataset's size and quality will greatly enhance the model's performance. The data set for this subject is tiny in volume and has some overfitting, and there is currently no public standard data set for mask wearing detection. Later on, the data set needs to be enlarged even more.

(2) There is still a considerable deal of opportunity for improvement in detection, particularly the accuracy of faces without masks. The strategy described in this work has significantly increased the model's detection speed. Face detectors do not match the detection accuracy of 86 percent. Imagine there is a specific gap. Later, we'll research the face-related literature, take note of techniques that can increase face detection accuracy, and further refine the algorithm presented in this paper.

References

- [1] Chavez S, Long B, Koyfman A, et al. Coronavirus Disease (COVID-19): A primer for emergency physicians[J]. The American Journal of Emergency Medicine, 2020.
- [2] Liu X, Zhang S. COVID - 19: Face masks and human - to - human transmission[J]. Influenza and Other Respiratory Viruses, 2020, 14(4): 472-473.
- [3] Jiao L, Zhang F, Liu F, et al. A Survey of Deep Learning-Based Object Detection[J]. IEEE Access, 2019, 7: 128837-128868.
- [4] Agarwal S, Terrail J O D, Jurie F. Recent Advances in Object Detection in the Age of Deep Convolutional Neural Networks[J]. arXiv preprint arXiv:1809.03193, 2018.
- [5] Liu L, Ouyang W, Wang X, et al. Deep Learning for Generic Object Detection: A Survey[J]. International Journal of Computer Vision, 2020, 128(2): 261-318.
- [6] Redmon J, Farhadi A. Yolov3: An Incremental Improvement[J]. arXiv preprint arXiv:1804.02767, 2018.
- [7] Loey M, Manogaran G, Taha M H N, et al. A Hybrid Deep Transfer Learning Model with Machine Learning Methods for Face Mask Detection in the Era of the COVID-19 Pandemic[J]. Measurement, 2020, 167:108288.
- [8] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]// IEEE Conference on Computer Vision and Pattern Recognition, 2016: 779-788.
- [9] Ejaz M S, Islam M R, Sifatullah M, et al. Implementation of Principal Component Analysis on Masked and Non-masked Face Recognition[C]// 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT). IEEE, 2019: 1-5.
- [10] Nieto-Rodríguez A, Mucientes M, Brea V M. System for Medical Mask Detection in the Operating Room Through Facial Attributes[C]// Iberian Conference on Pattern Recognition and Image Analysis. Springer, Cham, 2015: 138-145 .
- [11] Niu Zuodong, Qin Tao, Li Handong , et al. Improving RetinaFace 's Mask Wearing Detection Algorithm in Natural Scenes [J]. Computer Engineering and Applications, 2020, 56(12): 1-7.
- [12] Zhang Xiubao, Lin Ziyuan, Tian Wanxin, et al. Face mask recognition technology in all-weather natural scenes [J]. Science in China: Information Science, 2020, 50(07): 1110-1120.
- [13] Xu Y, Yan W, Sun H, et al. CenterFace : Joint Face Detection and Alignment Using Face as Point[J]. arXiv preprint arXiv:1911.03599, 2019.
- [14] Tang X, Du DK, He Z, et al. Pyramidbox : A Context-assisted Single Shot Face Detector[C]// European Conference on Computer Vision (ECCV), 2018: 797-813.
- [15] Qin B, Li D. Identifying Facemask-Wearing Condition Using Image Super-Resolution with Classification Network to Prevent COVID-19[J]. Sensors, 2020, 20(18).
- [16] Bochkovskiy A, Wang CY, Liao HY M. YOLOv4: Optimal Speed and Accuracy of Object Detection[J]. arXiv preprint arXiv:2004.10934, 2020.
- algorithm based on improved YOLOv3 in complex scenes [J]. Computer Engineering, 2020, 46(11): 12-22.
- [18] Xiao Junjie. Face mask detection and standardized wearing recognition based on YOLOv3 and YCrCb [J]. Software, 2020, 41(07): 164-169.
- [19] Cao Chengshuo, Yuan Jie. Mask wearing detection method based on YOLO-Mask algorithm [J/OL]. Progress in Laser and Optoelectronics : 1-13, 2020-10-10 [2020-11-20]. <http://kns.cnki.net/kcms/detail/31.1690.TN.20201009.1330.006.html>.
- [20] Guan Junlin, Zhi Xin. Mask wearing detection method based on YOLOv4 convolutional neural network [J]. Modern Information Technology , 2020, 4(11): 9-12.
- [21] Deng Huangxiao. Mask wearing detection method based on transfer learning and RetinaNet [J]. Electronic Technology and Software Engineering , 2020(05): 209-211.
- [22] Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot Multibox Detector[C]// European Conference

-
- on Computer Vision. Springer, Cham, 2016: 21-37.
- [23] Lin M, Chen Q, Yan S. Network In Network[J]. arXiv preprint arXiv:1312.4400, 2013.
- [24] Zhang C L, Luo J H, Wei X S, et al. In Defense of Fully Connected Layers in Visual Representation Transfer[C]// Pacific Rim Conference on Multimedia. Springer, Cham, 2017: 807-817.
- [25] Hecht-Nielsen R. Theory of the Backpropagation Neural Network[M]// Neural Networks for Perception. Academic Press, 1992: 65-93.
- [26] Ren S, He K, Girshic R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6):1137-1149.
- [27] Girshick R, Donahue J, Darrell T, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation[C]// IEEE Conference on Computer Vision and Pattern Recognition, 2014: 580-587.
- [28] Van de Sande K E A, Uijlings J R R, Gevers T, et al. Segmentation as selective search for object recognition[C]// 2011 International Conference on Computer Vision. IEEE, 2011: 1879-1886.
- [29] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features[C]// 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. IEEE, 2001, 1: I-I.
- [30] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]// 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). IEEE, 2005, 1: 886-893.
- [31] Felzenszwalb P, McAllester D, Ramanan D. A discriminatively trained, multiscale, deformable part model[C]// 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2008: 1-8.
- [32] He K, Zhang X, Ren S, et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1904-1916.
- [33] Girshick R. Fast R-CNN[C]// IEEE International Conference on Computer Vision, 2015: 1440-1448.
- [34] Dai J, Li Y, H K, et al. R-FCN: Object Detection via Region-based Fully Convolutional Networks[C]// Advances in Neural Information Processing Systems, 2016: 379-387.
- [35] Li Z, Peng C, Yu G, et al. Light-Head R-CNN: In Defense of Two-Stage Object Detector[J]. arXiv preprint arXiv:1711.07264, 2017.
- [36] Lin T Y, Dollár P, Girshick R, et al. Feature Pyramid Networks for Object Detection[C]// IEEE Conference on Computer Vision and Pattern Recognition, 2017: 2117-2125.
- [37] Cai Z, Vasconcelos N. Cascade R-CNN: Delving Into High Quality Object Detection[C]// IEEE Conference on Computer Vision and Pattern Recognition, 2018: 6154-6162.
- [38] Qin Z, Li Z, Zhang Z, et al. ThunderNet: Towards Real-Time Generic Object Detection on Mobile Devices[C]// IEEE International Conference on Computer Vision, 2019: 6718-6727.
- [39] Ma N, Zhang X, Zheng H T, et al. Shufflenet v2: Practical Guidelines for Efficient CNN Architecture Design[C]// European Conference on Computer Vision (ECCV), 2018: 116-131.
- [40] Szegedy C, Liu W, Jia Y, et al. Going Deeper With Convolutions[C]// IEEE Conference on Computer Vision and Pattern Recognition, 2015: 1-9.
- [41] Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger[C]// IEEE Conference on Computer Vision and Pattern Recognition, 2017: 7263-7271.
- [42] Hartigan J A, Wong M A. Algorithm AS 136: A K-Means Clustering Algorithm[J]. Journal of the Royal Statistical Society. Series C (Applied Statistics), 1979, 28(1): 100-108.
- [43] Fu C Y, Liu W, Ranga A, et al. DSSD: Deconvolutional Single Shot Detector[J]. arXiv preprint arXiv:1701.06659, 2017.
- [44] Li Z, Zhou F. FSSD: Feature Fusion Single Shot Multibox Detector[J]. arXiv preprint arXiv:1712.00960, 2017.
- [45] Zheng L, Fu C, Zhao Y. Extend the shallow part of single shot multibox detector via convolutional neural network[C]// Tenth International Conference on Digital Image Processing (ICDIP 2018).

International Society for Optics and Photonics, 2018, 10806: 1080613.

[46] Jeong J, Park H, Kwak N. Enhancement of SSD by concatenating feature maps for object detection[J]. arXiv preprint arXiv:1705.09587, 2017.

[47] Cui L, Ma R, Lv P, et al. MDSSD: Multi-scale Deconvolutional Single Shot Detector for Small Objects[J]. arXiv preprint arXiv:1805.07009, 2018.

[48] Zhao Q, Sheng T, Wang Y, et al. M2Det: A Single-Shot Object Detector Based on Multi-Level Feature Pyramid Network[C]// AAAI Conference on Artificial Intelligence, 2019, 33(01): 9259-9266.