2022-02-18

Deep Learning Based Real-Time Facial Mask Detection and Crowd Monitoring

Yang, C-Y

http://hdl.handle.net/10026.1/18620

10.31577/cai_2021_6_1263
Computing and Informatics
Slovak Academy of Sciences

DEEP LEARNING BASED REAL-TIME FACIAL MASK DETECTION AND CROWD MONITORING

Chan-Yun YANG[a], Hooman SAMANI[b], Nana JI[c], Chunxu LI*[d], Ding-Bang CHEN[a], Man QI[e]

[a]*Department of Electrical Engineering, National Taipei University, Taiwan;*
[b]*School of Engineering, Computing and Mathematics, University of Plymouth, UK;*
[c]*Shandong Engineering Research Center for Marine Science, Qingdao, China;*
[d]*Faculty of Science and Engineering, Swansea University, UK;*
[e]*School of Engineering, Technology and Design, Canterbury Christ Church University, UK*
e-mail: `chunxu.li@swansea.ac.uk`

Abstract. During the Covid pandemic, the importance of wearing mask has been noted globally. Additionally, crowded human clusters facilitated the transmission of the virus, which brings up the need for new systems for monitoring such situations. To address such issues, this research proposes an object recognition visual system based on deep learning to monitor the wearing of masks in a certain space and the control of the number of people indoors as an important tool during an epidemic. This research mainly investigates two types of identification. The first is to monitor whether people entering the site wear a mask at the entrance and exit of the field, and the second is to count the number of people entering a specific area. Experimental results show that by utilising the visual sensor, it is possible to detect and identify the people who frequently enter and exit in real-time. An advanced transfer learning approach has been employed to achieve the best discrimination performance. The actual training results prove that the migration learning Mask R-CNN algorithm produced by this method and the original Mask R-CNN algorithm have increased the mAP by 3%, reaching a mAP of 96%. In addition, the accuracy of the random sampling and identification in actual scenes has reached 92.1%. The developed deep learning vision system has an enhanced identification ability for the verification and analysis of actual scenes and has great application potential.

Keywords: Mask R-CNN, object detection, people flow control, deep learning, transfer learning.

1 INTRODUCTION

Coronaviruses are a large group of highly infectious viruses that may cause diseases in animals or humans. In humans, several coronaviruses are known to cause respiratory infections, from the common cold to more serious diseases such as Middle East Respiratory Syndrome (MERS) [1], Severe Acute Respiratory Syndrome (SARS) [2], and COVID-19 [5] which is an infectious disease caused by the coronavirus, and has caused a pandemic. Coronaviruses can be spread primarily through coughing and sneezing, close personal contact such as touching or shaking hands and touching an object or surface with the virus on it, then touching your mouth, nose, or eyes. The COVID-19 outbreak has become a pandemic, affecting all the countries globally.

Deep Learning [16] is an emerging field under machine learning and artificial intelligence. The motivation is to simulate and build the neural network of the human brain for analysis and learning via input perception data into the mechanism of deep neural networks. The identification tasks are classified, grouped, labeled, or positioned by learning and modeling from the data. By an underlying deep non-linear network structure, the learning can achieve an approximation of complex functions, represent a distributed representation of the input data, and get the powerful ability to assess the basic characteristics behind the finite sample set [3]. Deep learning could provide various solutions in computer vision, speech recognition and many other applications [10][26][7]. When facing the challenges of the global pandemic, deep learning could be a good mean to tackle various issues such as medical and social domains. In general, artificial intelligence could be employed to create robotic agents for smart epidemic prevention [17].

This paper aims to provide AI solution for pandemic as follows:

- Employing the two-stage object detection approach to correctly and accurately classify those faces that are wearing a mask.
- Using a universal dataset in order to develop a system which could be used globally even tough a small amount of self-collected data is used for the validation.
- Experimenting and testing the proposed system in actual scenes for performance evaluation.

The main contributions of this paper are as following:

- Using the deep learning based object recognition system to detect whether a mask is worn or not.
- Monitoring and controlling the number of people entering and exiting a specific field to reduce the virus transmission.
- Development of an augmented Mask R-CNN model by integrating transfer learning to improve the performance of the system.

The reminder of the paper is organized as follows: Related works are present in Section 2; the methodologies of model set up, training and optimisation are introduced in Section 3; a series of experiments are presented in Section 4 to validate the proposed method; and a conclusion with future work direction are discussed in the last Section.

2 RELATED WORKS

Rapid progress in Deep Learning (DL) and improvements in its capabilities have improved the performance and cost-effectiveness of several vision based systems. Compared to the traditional computer vision techniques, DL enables computer vision applications to achieve greater accuracy. Since deep neural networks used in DL are trained rather than programmed, applications using this approach often require less expert analysis and fine-tuning and exploiting the tremendous amount of visual data available in the systems. DL also provides superior flexibility because models and frameworks can be re-trained using a custom dataset for any use case, contrary to computer vision algorithms, which tend to be more domain-specific [24].

2.1 Traditional Object Detection

Before deep learning gets widely used to various image precessing applications, traditional image feature extraction methods relied on experts in using experience and design algorithms. In such approaches, after the features are obtained, a linear or non-linear classifier is used for the classification task. Regarding the positioning of the object, it is possible to perform a thorough search of the entire image (sliding window) and find the possible region of the object (region proposal), and solve the problem of multiple predictions. The traditional object recognition is divided into three steps, which are image pre-processing, feature extraction and object positioning.

In the image pre-processing module, due to the environmental influences such as light source interference, camera vibration and excessive image noise, the quality of the obtained image is poor, or there is unnecessary information in the image. For that goal, the common approach is to properly process the image to achieve the effect of removing noise, enhancing the image, RST (Rotation-Scaling-Translation) and invariant adjustment in order to improve the quality of the image and highlight the information needed for the next process.

The feature extraction module is to segment the features that are beneficial to the classification of the objects, and then use these features to describe or represent the shape and color of an object. In the problem of object detection, the method of feature extraction is based on a simple observation. This feature must be able to separate the object from the background and still have the characteristics of describing the invariance of the object after displacement and deformation. In the past, most imaging works used the methods of Histogram of Oriented Gradient (HOG) [6] and Scale-invariant Feature Transform (SIFT) [21] to extract features that can be used for object detection.

Traditional object recognition often relied on the sliding window algorithm. The algorithm uses one or more rectangular windows of different sizes to scan the original image, and extracts the characteristics of each window image according to the above method. A recognition classifier was then used to determine whether the location of the window contains the desired object or not. Usually, the sliding window method causes the object in the test image to have multiple bounding box predictions. For this, non-maximum suppression (NMS) [23] was usually used to select the best bounding box as the final prediction.

2.2 Object Detection Based on Deep Learning

There are currently two deep learning algorithms in the field of object detection: One is one-stage object detection, and the other one is two-stage object detection, as shown in Fig. 1. While the algorithm of the former is faster, the accuracy is relatively lower than for the latter. The difference in the structure of these two can be observed from Fig. 1. After the image input, a backbone network is used for feature extraction. The module of two-stage object detection is employed for separated calculation of object localization and classification, in contrast to that of one-stage object detection which, taking the computational speed into account, combines the object localization and classification to output. Actually, either one-stage or two-stage object detections has their own merits depending on the applications. Considering that the experimental scenes of this paper are more complicated and must calculate the number of people accurately, this research has adopted Mask R-CNN of two-stage object detection for identification accordingly.

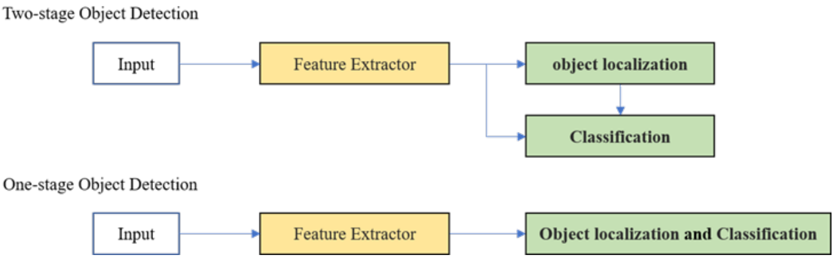

Two-stage Object Detection
Input
Feature Extractor
object localization
Classification
One-stage Object Detection
Input
Feature Extractor
Object localization and Classification

Fig. 1. Comparison of one-stage and two-stage object detection.

2.2.1 Two-Stage Object Detection

For two-stage object detection, various CNN based methods such as R-CNN, Fast R-CNN and Faster R-CNN are common for object detection. Region-Convolution Neural Network(R-CNN) [9], proposed by Girshick et al., is the first algorithmic architecture which was successfully applied the deep learning to CNN as a milestone in the application of CNN for object recognition problems. Beneficial from the CNN's excellent feature extraction and classification performance, and the Region Proposal methodology, R-CNN achieves the object identification in a two-stage manner. Since the Region Proposal plays a key role in the success of the algorithm, the method is named after the initial letter R at the beginning which is referring to Region plus CNN.

Before the popularity of the Fast R-CNN, most of the scholars used SPPnet [12] to find the optimal solution for the problem of repeated convolution in R-CNN. However SPPnet has several limitations. On the other hand Fast R-CNN provides a comprehensive solution. With Fast R-CNN, not only the training steps are reduced, but also there is no need to store additional features. For the network structure, the Fast R-CNN algorithm based on the VGG16 [27] backbone network can be 9 times faster than R-CNN in training speed, and 3 times faster than SPPnet. For testing speed, Fast R-CNN is 213 times faster than R-CNN, and 10 times faster than SPPnet. Fast R-CNN reaches 66% mAP on the VOC2012 training set [8]. In the Fast R-CNN algorithm, the involved CNN is executed once, and then the features extracted by the CNN are used for 2,000 region proposals. Using Region of Interest (RoI) pooling, the extracted regions can correspond to the feature map output, and connect to the fully connected layer for Softmax [13] classification and bounding box regression, as illustrated in Fig. 2.

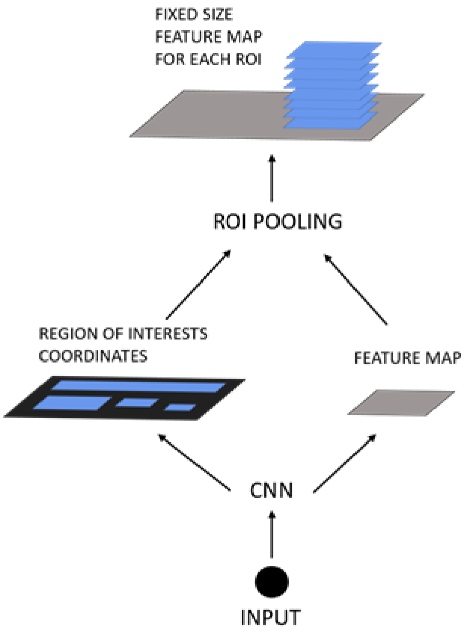


Fig. 2. Fast R-CNN algorithm architecture.

Fast R-CNN integrates and optimizes the steps of R-CNN, which not only greatly improves the detection speed, but also improves the detection accuracy. The improvement includes the convolution of the entire image instead of the convolution of each region proposal. There are three cores of the improved algorithm: the ROI Pooling, the simultaneous network for both classification and regression training, and the multi-task loss. The main disadvantage of Fast R-CNN is that the extraction of region proposals still uses selective search, which is the same as R-CNN. That increases the time for target detection.

2.2.2 Transfer Learning

The main purpose of transfer learning is to assist the training of the target domain model with a large number of training samples of the source domain, that is, to transfer the pre-trained model and parameters that have been trained in the source domain to the new model of the target domain. In the source domain, there are a lot of source data that are not directly related to our task, while in the target domain there are only a small amount of target data that is directly related to the task, but the two kinds of data are related or similar to each other, as shown in the Fig. 3. In the case of mask detection, the large dataset for face is available, however that is really limited in the case of faces wearing a mask. Hence, transfer learning method could be employed for training.

The transfer learning method extracts knowledge from one or more source tasks and applies it to the target task. Pan et al. [25] conducted a survey on transfer learning, in which range, task, and marginal probability were used to provide a framework to help understand transfer learning. The framework is defined as follows: Given the source domain D_s, the corresponding source task T_s, target domain D_T and target task T_T, where D_s is not equal to D_T or T_T (in most cases, it is assumed that the number of target samples is much smaller than the number of source samples). The purpose of transfer learning is to enable us to obtain information from D_s and T_s to learn the target condition probability distribution of D_T.

A small amount of target data can easily cause overfitting of the target model, because when the training set is too small, the model cannot find generalized features. Therefore, in training, the first few layers of the pre-trained model are usually locked to reduce the training parameters and avoid overfitting. Yosinski et al. [29] experimentally quantified the generality versus specificity of neurons in each layer of a deep convolutional neural network. Through experiments, it is known that the effect of fixing the first few layers is significant, because the features captured at the beginning are relatively low-level and have high versatility. It is not required to manually label the data or train the model from scratch for transfer learning, instead the process is just to apply the weight of the trained model to the target field. Through this method, the training time and cost are greatly reduced.

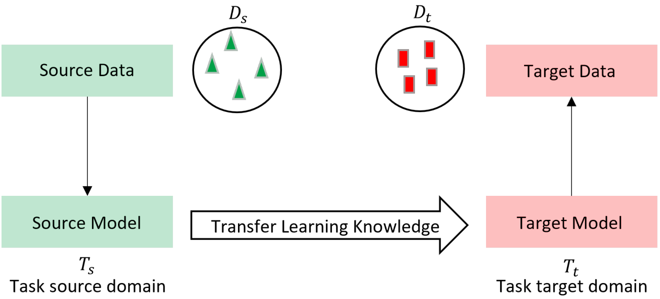


Fig. 3. Transfer Learning concept.

3 METHODOLOGY

3.1 Overall System Design

The aim of our design is to develop a vision system for a robotic system for pandemic situation to count the number of people passing through and detect their facial mask wearing by a Mask R-CNN model. The introduced vision system is sought to help the epidemic control when people enter and crowed in a closed space. The system architecture is shown in Fig. 4.

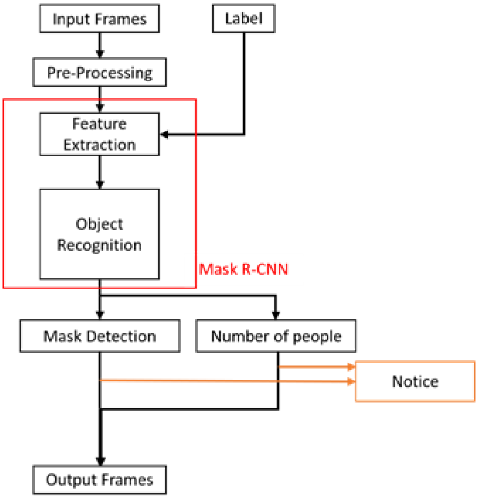


Fig. 4. Visual system architecture diagram.

Wearing a mask is the one of the keys measures to prevent the epidemic such as COVID-19, and the calculation of the number of people can prevent too many people in the closed space, resulting in the lack of circulation and the inability to maintain social distance. Therefore, the function used can prevent the spread of the virus in time. The integration of the above-mentioned multiple function system can send notifications and warnings when someone does not wear a mask or there are too many people in the community. The proposed robot vision system can be employed in the entrances of schools and public spaces or applied to automatic surveillance systems to achieve epidemic prevention and reduce manpower requirements. The overall system diagram is shown in Fig. 5.

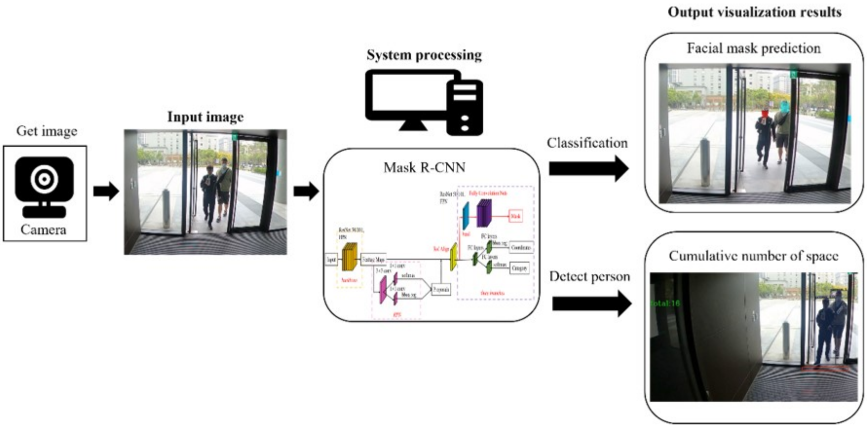


Fig. 5. Overall system diagram.

3.2 Mask R-CNN

Mask R-CNN is a typical two-stage image processing algorithm. The first stage is RPN which will be explained in details later in this paper. The second stage predicts the bounding box and target category in parallel, and then uses RoI to output an additional binary mask. The classification of objects depends on mask prediction. The preprocessed picture feeds into the backbone network as an input to extract the features, so that the corresponding feature map can be obtained, and the features are also input into the RPN to obtain the corresponding anchor frame. Then, using the obtained anchor frame and feature map, using RoIAlign, the feature map corresponding to each anchor frame is unified into a single size to obtain a fixed size feature map. After obtaining the feature map of each anchor box the mask is generated and the classification and bounding box regression are generated through the fully connected layer. This is to apply the bounding box classification and regression at the same time through the concept of Faster R-CNN. This structure is shown in Fig. 6.

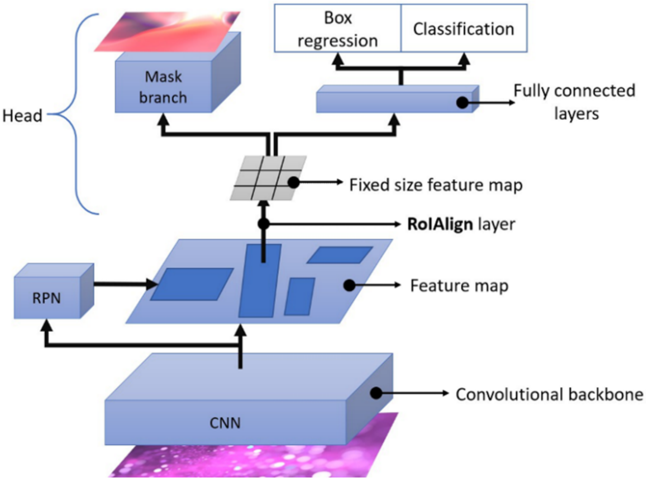


Fig. 6. Mask R-CNN structure diagram [4].

3.2.1 Loss Function

Based on Faster R-CNN, Mask R-CNN is developed by adding a new mask branch. The structural change makes the loss function of Mask R-CNN changed accordingly. Due to the addition of a new mask branch, the loss function of each RoI is expressed as:

$$L = L_{cls} + L_{box} + L_{mask} \tag{1}$$

where L_{cls} denotes the classification loss and L_{box} is the bounding box loss as those defined originally in Faster R-CNN, and L_{mask} denotes the loss function incurred by the addition mask branch. L_{mask}, here, was asserted to be proportional to the output dimension, and was expressed as [11]:

$$L_{mask} = Km^2 \tag{2}$$

where m^2 represents the mask size, and K represents number of channels which are the number of categories to be classified. All the RoIs generate a total of K binary masks. After the predicting mask is obtained, each value of the mask pixel is calculated by a sigmoid function, and it results in one of the outputs of L_{mask} (binary cross entropy loss function). The operation is illustrated in the Fig.7.

The expression of channel is 3 in the figure which means there are 3 categories in total. The RoI on the upper right corresponds to the category of K = 1, that is, the two-class classification that is performed for each pixel on the mask of channel that is 1. The RoI at the bottom left corresponds to the category of K, that is 3. The second classification is performed on each pixel on the mask of channel, that is 3. The classification of each category is through the classification network which tells which category the RoI is corresponding to the segmentation network. The

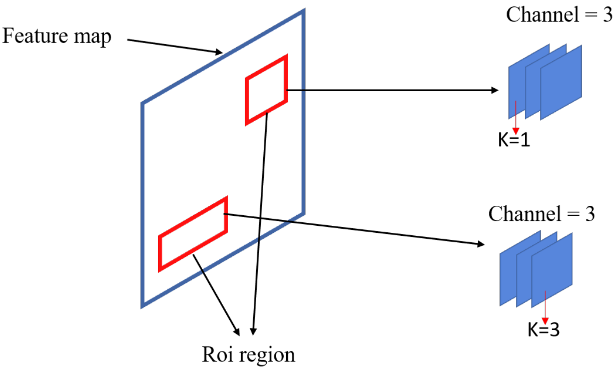


Fig. 7. Output structure diagram of binary cross entropy loss function.

advantage of this is that the defined L_{mask} can generate a mask for each category in the network, which is only used for the calculation loss of that category. So it will not lead to the result of each category for competition. After calculating the loss for the feature map of each fixed category, and then averaging all the pixels, the binary cross-entropy loss function can be obtained.

3.2.2 Region Proposal Network

In Mask R-CNN, the principle is the same as Faster R-CNN. The RPN is first used to find the proposal and then results are further screened through Mask R-CNN. The type and location of each target in the picture can be found in the RPN. However, because the detection effect for smaller objects is poor, further optimization is required through the framework. In the structure of the RPN, the input image is extracted through the backbone network to obtain a feature map. The method to locate the object's location category from the feature map is to apply the concept of anchor. Anchor is generated through any pixel on the feature map by first mapping to a 16 x 16 area on the original image, and then using the center of this area as the transformation center. The next step is to turn it into an area with three aspect ratios, and then expand its area 8, 16, or 32 times. The last pixel corresponds to the 9 different rectangular boxes in the original image. These boxes are called Anchor. The Anchor can be used to generate region proposals. The complete structure is shown in Fig.8.

In the RPN structure diagram shown in Fig.9, it can be seen that Mask R-CNN is a better framework as an improved Faster R-CNN. The RPN is also reserved to generate region proposals. The operation process first performs another convolution on the previously output feature map and then divide it into two branches. The above branch is used to determine whether the anchors belong to the foreground or the background through Softmax. The results of the two classifications are sent to the proposal layer. The other branch is bounding box regression to modify anchors to

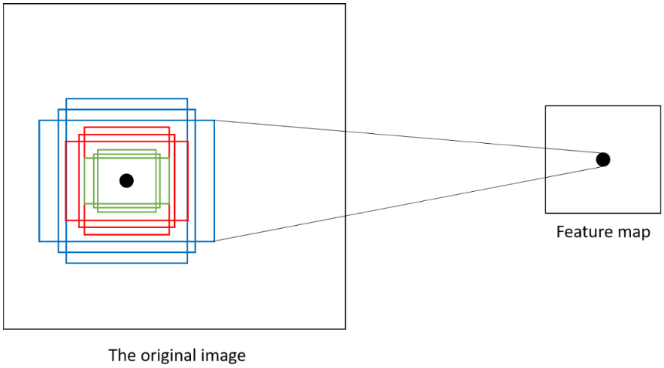


Fig. 8. Anchor generation diagram.

obtain accurate proposals. After the above steps, the final proposal layer adjusts the position of each anchor according to the regression result of bounding box regression, and excludes anchors that exceed the boundary and have a high degree of overlap. Then sorting is performed according to the foreground score, taking out a small number of anchors as proposals, the coordinates of each proposal are the coordinates on the original image, which are finally sent it to the RoIAlign layer to complete the overall process of RPN.

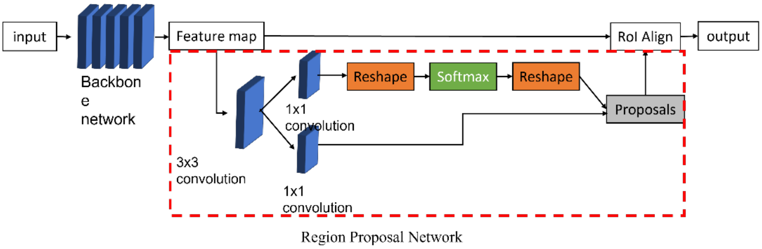


Fig. 9. RPN (Region Proposal Network) structure diagram.

3.2.3 Feature Pyramid Network

FPN (Feature Pyramid Network) [18] is a method that uses models to efficiently extract features in each dimension of an image. In the FPN structure diagram of Fig.10, the part from top to bottom is a process in which the features are gradually reduced from bottom to top to express the features. The lower layer displays the features of the image information of the shallower layers, such as edges. The higher layers display deeper image feature information, such as object outlines or categories. The feature map size of the feature output in the upper layer is relatively small, but it can represent a larger dimensional image information. In the horizontal output part, the top-level features are used for fusion through upsampling and layer features, so that each layer is an independent prediction. This paper uses 1×1 convolution to generate output features, which can reduce the number of feature maps without changing the size of feature maps. After 1x1 convolution and decoder results are added, the relationship between the feature maps output by each stage is doubled, so the size of the feature map obtained by the previous layer upsampling is the same as the size of the current layer, and the corresponding elements can be added directly. Finally, the 3×3 convolution is used as the output of this layer, and the number of channels in the decoder part is 256.

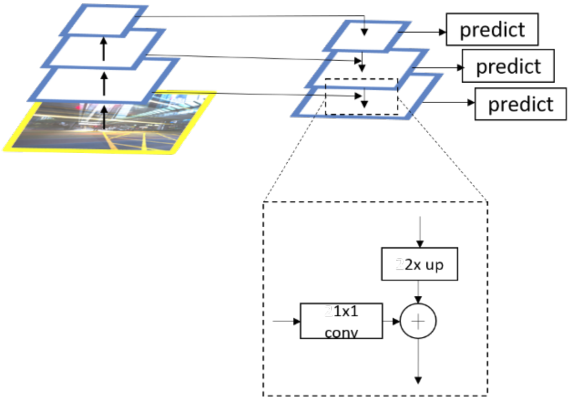


Fig. 10. FPN (Feature Pyramid Networks) structure diagram.

3.2.4 RoI Alignment

RoIAlign is a regional feature aggregation method proposed in Mask R-CNN [11], which solves the problem of regional misalignment by twice discretization steps in the RoI pool. The concept of RoIAlign is to obtain the point values of an image of which the primary values of the points are floating numbers by discretization round-off using a bilinear interpolation. Thereafter, the entire feature aggregation process could be transformed to be operated continuously. The operational process of RoIAlign is shown in Fig.11. The blue dashed boxes in the figure represent the feature maps obtained after convolution. The black solid boxes represent the RoI regions. As shown, the final output size is 2×2. The calculated blue dots are also the ordinary random sampling points in the 2×2 cell, so the number and location of the sampling points will not have a great impact on the performance. There is no overall round-off operation in the process, and also no induced error in the calculation. In other words, the pixels and the pixels in the feature map are completely aligned with no deviation, so the detection accuracy can be improved.

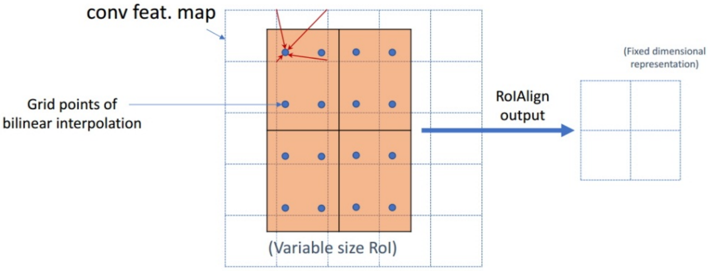

conv feat. map
Grid points of bilinear interpolation
RoIAlign output
(Fixed dimensional representation)
(Variable size RoI)

Fig. 11. RoIAlign operation process.

3.3 Residual Network

The core neural network part is the ResNet neural network [28]. The design concept of the network structure is that as the network deepens, the more features the neural network can calculate, the better results it can achieve. The disadvantages of deeper neural networks require very large training parameters, which leads to a large amount of computing resources. But in fact, as the network deepens, the size of the gradient drops sharply, which will cause the learning rate to be very slow. In rare cases, the gradient will rise sharply, that is, gradient explosion phenomenon. Compared with the shallow network, the accuracy shown on the training set is not improved, but will decrease. The residual network is a kind of network proposed to solve the network deepening gradient disappearance or gradient explosion, and the residual network is easier to optimize, and can increase the accuracy by increasing a considerable depth. The core is to solve the side effects of increasing depth, which can improve network performance by simply increasing network depth. Therefore, this paper uses ResNet101 as the core of neural network.

In the neural network, VGG and ResNet use the same building blocks to build the network [14]. Among them, ResNet proposes two mappings, one is identity mapping, and the other is residual mapping. Identity mapping refers to itself, which is the x in the formula, while residual mapping refers to the difference, which is y-x, so the residual refers to the F(x) part (Fig.12).

In the residual learning unit structure of Fig.12, two weight layers are defined as new nouns called Residual. The Residual operation value can be expressed as:

$$Residual = H(x) - x \tag{3}$$

That is, the difference output of Residual can be obtained through the above formula. Therefore, Residual is also a function of x, so it is also written as F(x). Finally, the expression can be expressed as:

$$F(x) = H(x) - x \tag{4}$$

From the above formula, we can infer that by shifting the term of the expression,

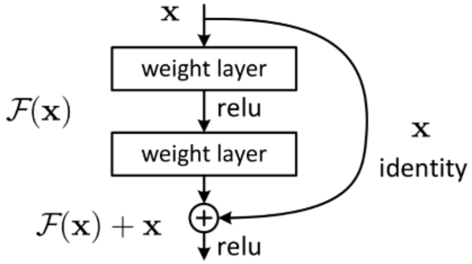

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Fig. 12. ResNet's residual learning unit [28].

the Residual output can be expressed as:

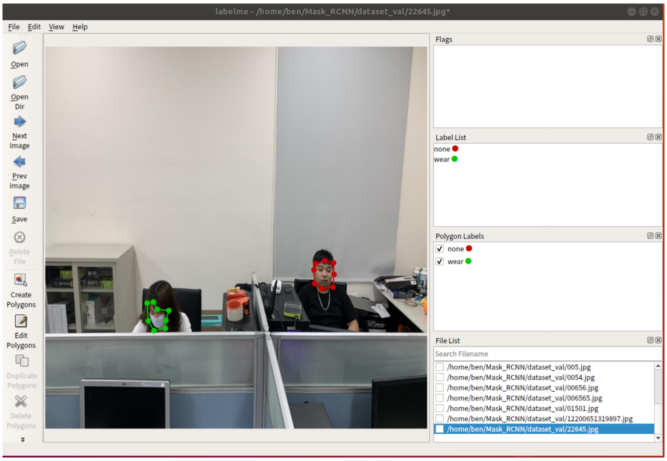
$$H(x) = F(x) + x \tag{5}$$

After the final result is calculated by Residual, the final result is sent to the ReLU layer for calculation.

3.3.1 Experiment Procedures

The experiment process is divided into data collection, labeling, and model training. In order to accurately detect whether a person is wearing a mask, we use the mask dataset on Kaggle, plus about 300 pieces of data collected by ourselves as a training set. In each image, the facial information of each person is manually tagged, and divided into two categories: masked and non-masked. In the manual labeling of data, we use Labelme, an image labeling tool developed in collaboration with the Massachusetts Institute of Technology (MIT) Computer Science and Artificial Intelligence Laboratory (CSAIL). Through the label tool, we can frame the target in the form of a polygon and give it a corresponding label, as shown in Fig.13. The next step is to output the labeled data and use the tags to perform data training.

In the process of training model, parameter setting is an important part of training. This paper uses the marked data in the previous part and uses GPU for training through the CUDA computing platform and TensorFlow-gpu. In terms of parameter setting, the class is set to 3, respectively wearing a mask, not wearing a mask, and background. According to the number of categories, we set each step to 100 and train 200 times, and adjust the batch size to the largest possible value, so we adjust it to 8.n the aspect of image width and height setting. This paper tried two different methods. The first is to directly train the data set after labeling. However, the model of this approach is very inaccurate, and the verified model often has detection errors. So we proceed to use the second method, by first adjusting the size of the images in the dataset, and we changed the size of the images to a uniform size. Next step was to test the recognition results of the two. If the photo is blurry or occluded, the latter has a relatively higher probability of detection error compared with the former. Therefore, we adopt the latter method to change both



Label List:
none ●
wear ●

Polygon Labels:
☑ none ●
☑ wear ●

File List:
□ /home/ben/Mask_RCNN/dataset_val/005.jpg
□ /home/ben/Mask_RCNN/dataset_val/0054.jpg
□ /home/ben/Mask_RCNN/dataset_val/00656.jpg
□ /home/ben/Mask_RCNN/dataset_val/006565.jpg
□ /home/ben/Mask_RCNN/dataset_val/01501.jpg
□ /home/ben/Mask_RCNN/dataset_val/12200651319897.jpg
□ /home/ben/Mask_RCNN/dataset_val/22645.jpg

Fig. 13. Labelme labeling data.

the output image and the input image to the same size. In the part of the learning rate, the learning rate in the Mask R-CNN [11] literature is set to 0.02, but since on the implementation of the platform and the optimization parameters are different, if it is not changed, the gradient will explode. Therefore, after many tests, it is found that the training effect is best if the learning rate is set to 0.001. Finally, we set the verification and training path and set the mAP calculation for training every 5 steps. After the above-mentioned data labeling and training parameter setting process, the neural network could complete the specified number of training iterations to get a trained model.

3.4 Data Augmentation for Training

In deep learning, the performance is usually proportional to the amount of data, so increasing training data is also an easy way to improve the performance of a model. Through the augmentation of training data, the accuracy of detection could be increased, i.e., the generalization ability of the model could be improved. Considering the placement of the visual system, there may be problems with backlighting or insufficient light. Therefore, this paper uses the ImageDataGenerator category in the Keras platform to enhance the brightness of the training data. While increasing the amount of data, it also increases the model's ability to recognize the bright or dark environments. The training data augmentation result is shown in Fig. 14.

With the above-mentioned data augmentation, we expanded the original 1,000 training data to 1,300 by lighting up or darken down the tone of the whole picture, in order to achieve the capability in recognizing the objects even when the exposure is too high or too low. The argumentation dataset owned 1,300 images, and for the sake of training, the dataset was divided into two categories as shown in Table 1.

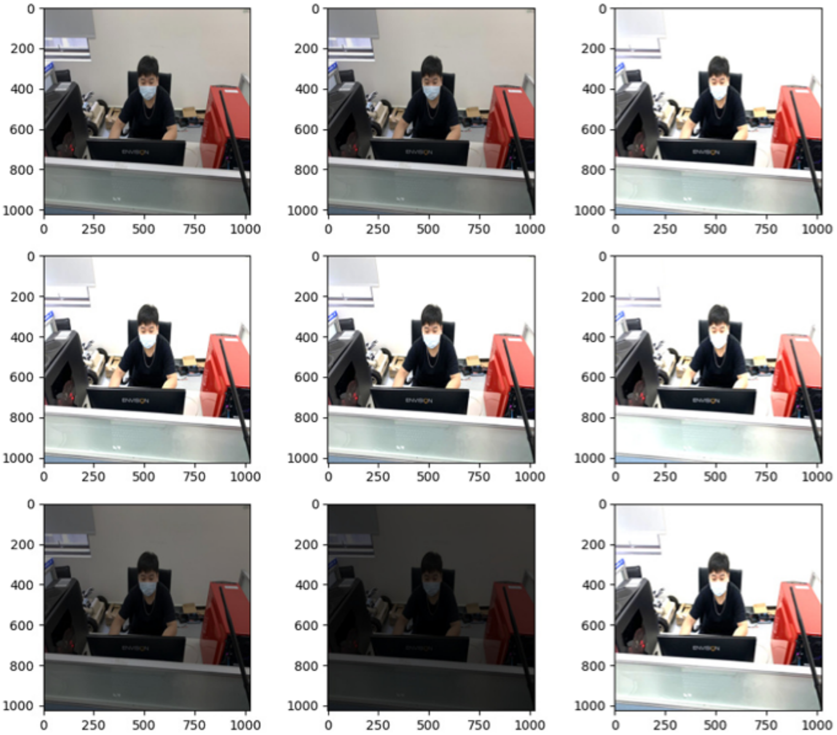


Fig. 14. Generated image after data augmentation.

Table 1. Training dataset category and number.

Class Name	Description	No. of images
With Mask	Faces with masks correctly used	642
Without Mask	Faces with no masks or masks incorrectly used	658

3.5 Enhancing the Model with Transfer Learning

Previously, we have increased the generalization ability of the model by enhancing the data. In this section, we introduce the method of pre-training the model and performing transfer learning. For part of the pre-training model, this research uses COCO weights [19] to train the ResNet-101 model. In the training data part, we use the 1300 sheets in previous section as the verification set and 1040 sheets as the training set and pre-training model for transfer learning. The process is shown in Fig.15. We first input the data enhanced by the training data, and then perform transfer learning through the pre-trained model and data. During transfer learning, the last prediction layer will be replaced with two fully connected layers to output two categories. The architecture of the proposed transfer learning is shown in Fig.16. After continuous evaluation and verification of the model, the new trained weight file is finally generated which can be used for subsequent model input of Mask R-CNN.

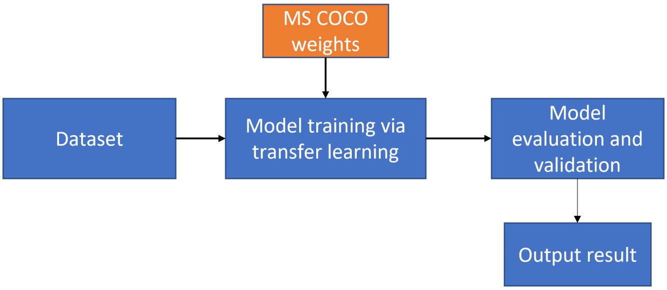

MS COCO weights
Dataset
Model training via transfer learning
Model evaluation and validation
Output result

Fig. 15. Flowchart of Transfer Learning.

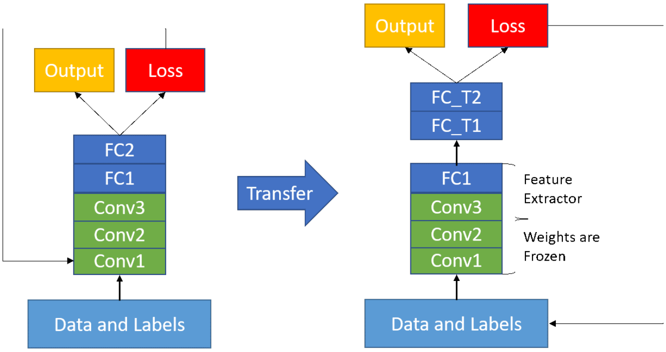


Fig. 16. Transfer Learning model architecture.

4 RESULTS AND DISCUSSIONS

4.1 Training Results on Different Models

Regarding the training part of the mask data, we use the Kaggle dataset of multi-person and environment-intensive photos for data annotation. The data set contains many different scenes. In each photo, the facial information of each person is manually labeled, and divided into two categories: with-mask and without-mask. The training epoch is set to 200, that is, 100 steps are done in each epoch, a total of 20,000 iterations. After the above iterative training process, we calculate the mean average precision (mAP) to observe the evaluation of the training model. The precision refers to the correct number of all predictions, and its concept can be expressed as:

$$precision = \frac{TP}{TP + FP} \tag{6}$$

where TP (True Positive) is a correctly predicted bbox, and FP (False Positive) is a wrong predicted bbox. This paper sets the threshold of IoU to 0.5. The verification data greater than 0.5 is classified as TP, and verification data less than or equal to 0.5 is classified as FP. Recall refers to the percent predicted in all correct categories. The concept can be expressed as:

$$recall = \frac{TP}{TP + FN} \tag{7}$$

where FN (False Negative) is the number of ground truth (GT) not detected. After getting the precision and recall functions, we use recall as the x coordinate and precision as the y coordinate. The PR curve is calculated for each category, and AP can be obtained by the graph area enclosed by the PR curve and the x-axis recall. Take the average of each type of AP to get mAP. This paper uses 800 of the 1,000 pictures as the training set, and the remaining 200 as the verification set. During the training process, repeated verifications are continuously carried out. Verification is performed every 5 epochs, and the mAP calculation of the above process is performed.

Experimental studies with/without transfer learning of the same dataset before and after the data enhancement were conducted. The training data is also 1,300 sheets, which contain random enhancement data with varying brightness and darkness. In the validation set part, this research uses 1,300 image data after enhancement and uses 1,040 random images as the training set, and the remaining 260 images as the validation set. During the training process, repeated verifications are continuously performed, and mAP calculations are performed every 5 epochs. The data comparison is shown in Table 2.

Table 2. Comparison of training total loss and mAP for data augmentation and transfer learning.

	Original model	Data augmentation only	Transfer learning
Training total loss	0.3	0.083	0.05
Training mAP	91%	93%	96%

According to the Table 2 listed the difference before and after the transfer learning, the total loss part of the training is 0.033 less than that before the augmentation, and the part of the mAP is 3% more than that before the transfer learning. In addition, we compare the total loss curves of the three models of original data, data augmentation and transfer learning, as shown in Fig.17.

In Fig.17, the three curves are the total loss line graphs of the original model, the data augmentation model, and the transfer learning model. In the curve in the figure,

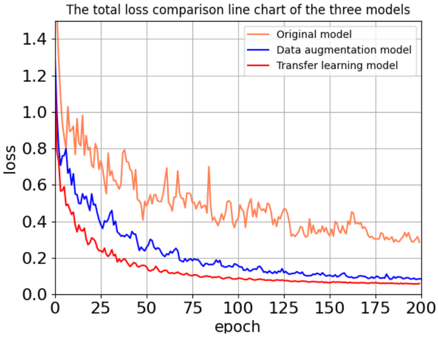


Fig. 17. The total loss comparison line chart of the three models.

it can be observed that the total loss curve after transfer learning has the fastest convergence speed. The total loss has converged to 0.1 at 40 epochs. Compared with the original model and the data augmentation model, the convergence speed is much faster. In the performance of mAP, as shown in Fig18 for the same 40 epochs, the mAP of the original model and the data augmentation model are 75% and 90%, respectively, but the transfer learning model has reached 95%. This data comparison also proves that in the case of a small amount of data, transfer learning not only shortens the model convergence time, but also improves the mAP of the model.

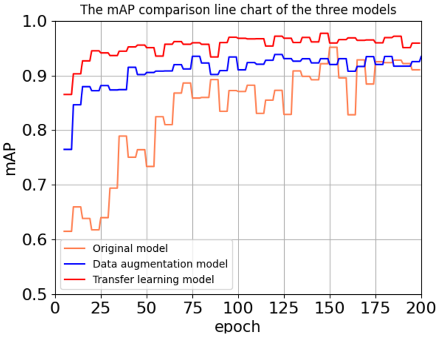


Fig. 18. The mAP comparison line chart of the three models.

4.2 Mask Detection Experiments

After training, the system can clearly distinguish, in simple scenario those, who wear a mask and those who don't, as shown in Fig.19. It can be seen that the red box on the left side of the figure shows that the person is wearing a mask, and the blue box on the right shows that the person is not wearing a mask.

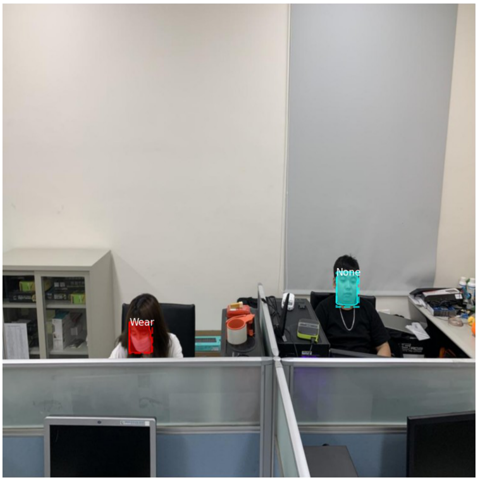


Fig. 19. Figure of identification results in a simple scenario.

As expected primarily, the mask detection aims to be used in surveillance systems or mobile robots. There will be many targets to be detected in one screen for a single-person or multi-person scenes. Usually, the accuracy of the recognition is affected by many factors, such as the occlusion of the detected target, the detection angle, and the size of the detected target in the process of detecting a target. To evaluate the effects from the factors, a primary model verification was designed and experimented. Face directions were arranged in respective angles of 0°(frontally aligned), 45°and 90°(sagittally aligned) for testing (Fig.20). The mask detector was then used for identifying the mask wearing.

The result of experiment could be found in Fig.21. There are no mis-detections occurred in the output regardless of the facing directions even when the face turned to sagittally-aligned 90°. Therefore, as long as the detection angle of the face is not greater than 90°, the developed model is able to clearly detect whether the target is wearing a mask or not.

The face-angle verification was extensively scaled up to test the capability of the model. Through the above implemented environment, sets of different images containing 95 frontally-aligned faces, 88 images 45°faces, and 84 sagittally-aligned faces were used to measure the recognition rate. The results are shown in Table 3.

According to the actual measurement results in Table. 3, the model had the highest recognition rate on the frontally-aligned direction, with a recognition rate of 95.7%. As the angel of the direction increased, the recognition rates of 45°and 90°were reduced to 88.6% and 84.5%, respectively. The experimental results confirmed that the rotation angle of the face has an effect on the recognition rate, but

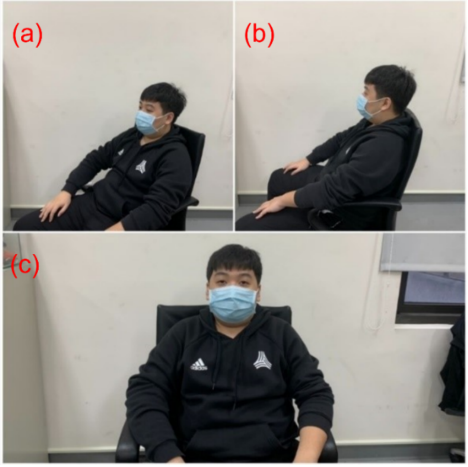


Fig. 20. Experiment with face directions, angle of (a). 45°, (b). 90°(sagittally aligned), (c). angles 0°(frontally aligned).

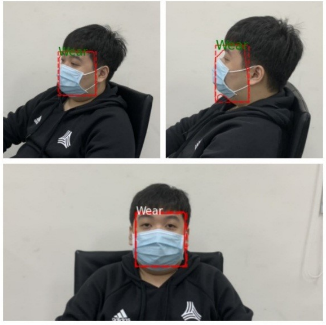


Fig. 21. The detection result of the face direction angles.

the effect was not substantial. In our experiment on the actual scenes, if the rotation angle of the target face is large, the recognition rate is reduced.

Table 3. Recognition rates of different face direction angles.

Angle	Correctly classified/All	Recognition rate
0°	91/95	95.7 %
45°	78/88	88.6 %
90°	71/84	84.5 %

4.3 Real Field Application Verification

For model verification, the model may not be good to be verified through only prepared-ahead pictures with some simple backgrounds. Therefore, cameras were used at a building gates and entrances to capture the actual scenes. in thish experiment when people enter ro exit, a picture was taken like those in Fig.22.

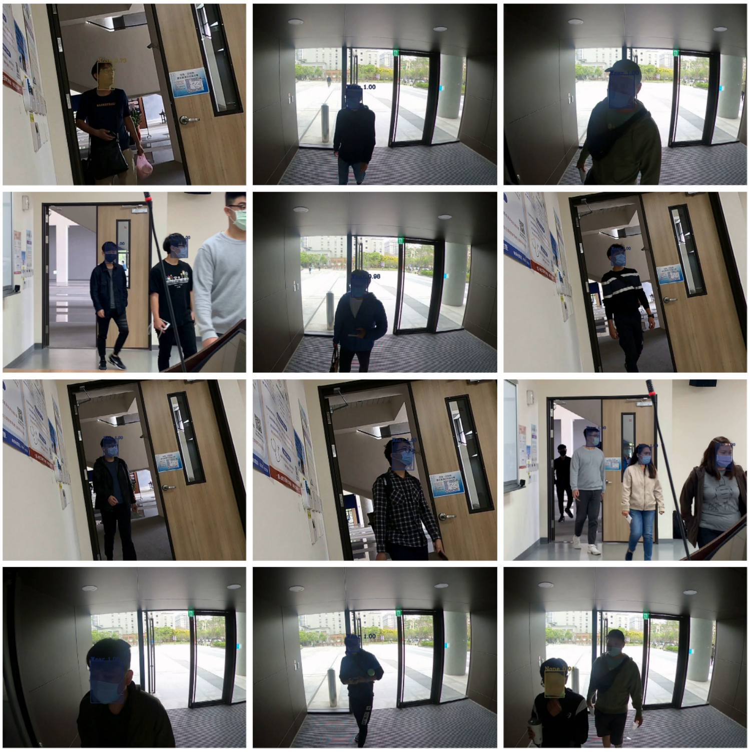


Fig. 22. Recognition results of experiments in various scenarios.

As shown in Fig.22, it can be seen that the people are accurately identified by the model, and the mask and outer frame are displayed in the correct position. In order to achieve more accurate and digital identification results, we set up cameras in many different scenes, such as gates and specific entrances and exits for long-term experiment. After sampling and calculation, we took 107 random passers-bys in different scenarios, and calculated the correct and inaccurate identifications one by one through statistical methods. Table 4 shows the confusion matrix of the classification.

		Actual Condition	
	Total = 107 people	Wearing a mask	Not wearing a mask
Prediction Outcome	Wear a mask	75	10
	Not wearing a mask	8	14

It can be observed that the accuracy of this model can be expressed as:

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

Through the calculation, it can be known that the actual detection precision, which is one of the most common indicators in theory to measure machine learning models, of this model is 88%. As a conservative index, the higher the score, the more accurate the model can predict. In addition, a counterpart Recall variable can be expressed as:

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

In the above formula, TP means that masks are worn in the actual situation, and the predicted output also includes the number of masks worn. FN is actually wearing a mask, but the predicted output is not wearing a mask. Furthermore, we can use Recall and Precision to find the value of F-score which can be expressed as:

$$F - score = \frac{(1 + \beta^2) precision \times recall}{\beta^2 precision + recall} \tag{10}$$

where β is a weighting factor to impact Recall or Precision. Since it is an average problem, there is no need to weight especially Recall or Precision. The same weight brings β=1 to the expression and leads it to a reduction model:

$$F - score = 2\frac{precision \times recall}{precision + recall} \tag{11}$$

It brings the calculation as:

$$2\frac{0.88 \times 0.90}{0.88 + 0.90} = 0.89 \tag{12}$$

From the result of the above formula, we can know that the model calculates F-score of 0.89. F-score is a commonly used index to measure the quality of a model, and it is often used to judge the accuracy of an algorithm. The closer the calculated F-score is to 1, the higher the accuracy of the model. In terms of the discrimination of another models, we use the aforementioned data to generate a confusion matrix, and we use randomly sampled 107 passerby data to make predictions. Based on that, we calculated the recorded 107 pieces of data according to the predicted and actual results of each piece of data, and calculated the true positive rate (TPR) and false positive rate (FPR) in the confusion matrix. Among them, the true positive rate is also called sensitivity, which is a positive sample of functional judgement. The higher the value, the better the performance. The calculation formula can be expressed as:

$$TPR = \frac{TP}{TP + FN} \tag{13}$$

Compared with the true positive rate, the false positive rate is the proportion of the true negative samples in the positive sample category to the total number of all negative samples. The calculation formula can be expressed as:

$$FPR = \frac{FP}{FP + TN} \tag{14}$$

The ROC curve can be obtained by connecting the TPR and FPR of each threshold calculated above, as shown in Fig.23.

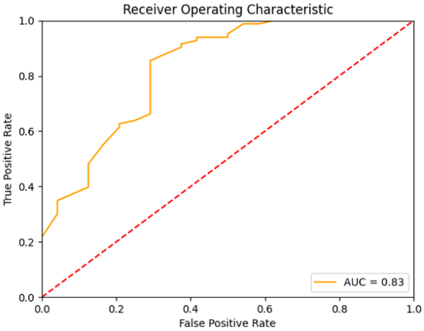


Fig. 23. ROC graph.

The yellow line in the above figure is the TPR (x-axis) and FPR (y-axis) curves of each threshold (0 1), and the orange curve is the separation line of the model prediction. Among them, the larger the value of the area under the ROC curve (AUC), the better the performance, the more accurate the classification. The calculation method of the area under the curve can be expressed as:

$$AUC = \frac{1}{2} \sum_{K=i}^{M} (f(x_{i+1}) + f(x_i)) * (x_{i+1} - x_i) \tag{15}$$

In the above formula, M is the threshold number, The K variable is the number represented by each threshold, $f(x_{(i+1)})$ and $f(x_i)$ are the upper and lower bases, and $(x_{(i+1)} - x_i)$ refers to the high parameter. Therefore, by using these variables, the area under the curve can be calculated from the sum of multiple trapezoids. The AUC value is a value between 0 and 1, and the closer the value is to 1, the better the classification ability of the model. After calculation, a fairly good AUC value 0.83 has been derived, i.e., the quality of the model is of a high level. Finally, based on the actual verification, we separately conducted experiments on different scenarios and Table 4 shows the comparison of the results.

Table 4. Comparison of actual verification results of three models.

	Accuracy	Precision	Recall	F-score	AUC
No data augmentation	83%	0.88	0.9	0.89	0.83
Data augmentation	87%	0.9	0.93	0.91	0.86
Transfer learning	92%	0.93	0.96	0.94	0.90

From the results Table 4, we can know that the performance of the model can be increased through transfer learning. It can be seen that even if the amount of data is very small, it can have a good performance. Under the above-mentioned multiple different evaluation indicators, the performance of the transfer learning model is higher than that of the model without transfer learning.

4.4 Mask Detection Performance Comparison with Related Works

To explore the benefit the proposed model, we compare the performance of our model with three related works: Jiang et al., [15], Nagrath et al., [22] and Loey et al., [20]. The average accuracy and the speed are discussed here for performance comparison. For average accuracy, statistics have been extracted respectively from the original contributions of [15], [22] and [20], and listed in Table 5. Comparing to the others listed in the Table, the AP value, which represents the performance accumulated under the ROC curve, of our proposed model outperforms those models from [22] and [20]. Furthermore, the amount of training samples in [15] is the largest comparing to that of [22] and [20], however the training samples in both [22] and [20] are approximately similar in their sample quantities. The comparison particularly to [15], measures the balance between the amount of 7,735 samples and 98% of AP with that of our proposed model, 1,300 samples and 96% of AP, only 2% accuracy lost from the proposed model. When the amount of data is not huge, the transfer learning model used in this paper also has good results in terms of accuracy. By comparing with the results of other related work, we found that the amount of training data needs to be increased compared with other related work, so that its average accuracy can be improved.

For comparison of detection time to the related works, the statistics were listed in Table 6. It can be found that the method used in proposed model is the slowest one while the method in [15] is the fastest one.

While the proposed architecture in this paper is a two-stage approach, the other stumps in the comparison are one-stage approaches. The long span of the two-stage is the main reason to make the recognition speed slower than those One-Stage

Table 5. Comparison of different related works in terms of average accuracy (AP).

Adopted Methodology	Training Sample	Average Precision (AP)
SE-YOLOv3 with SE-Darknet53	7,385	98%
SSD with MobilenetV2	1,376	92%
YOLOv2 with ResNet-50	1,415	81%
Proposed model with ResNet-101	1,300	96%

Table 6. Performance comparison of different methods in related work.

Reference Model	Adopted Methodology	Detection Time (ms)
Jiang et al., [15]	SE-YOLOv3 with SE-Darknet53	35.5
Nagrath et al., [22]	SSD with MobilenetV2	55.5
Loey et al., [20]	YOLOv2 with ResNet-50	66.6
Proposed model	Mask R-CNN with ResNet-101	71.4

detectors. Therefore, speeding up the recognition without affecting the accuracy of the recognition could be the direction for future improvements.

5 CONCLUSION

This paper proposed a model for crowd controlling with focus on mask detection and people counting for the specific usage of pandemic situations. It was shown and verified that by using deep artificial neural network, the system could perform the assigned task successfully. Even if the detection background is complex, the system has demonstrated an excellent performance on the detection of whether the person is wearing a mask via the model of Mask R-CNN with FPN multi-scale detection. The accuracy of the recognition during training reached 93% mAP, and through transfer learning, the accuracy has been increased to 96%. In the verification part of the actual scene, the identification accuracy reached 92.1%.

In addition to improving the training accuracy and recognition accuracy of the system, improving occlusion and illumination are an important factor which could be considered for the improvement of this system. This research assumes the condition where the person is wearing the mask correctly i.e. the mask is covering mouth and nose. However one of the common issues with mask wearing is that many people are not covering their nose or they wear the mask incompletely. That issue is an important health aspect and it is required to investigate it in the future by analysing the face with and without mask separately in order to analyse the proper mask wearing.

In the future, the proposed system is expected to be employed in various applications. For example this system could be widely used in surveillance systems which is not only limited to the pandemic. As an example wearing a mask could be important due to air pollution. Also robotic applications such as mobile robots could use such system as part of their vision module. It is also expected that through the above improvements, such a system can be applied to outdoor scenes in addition to indoors. Since wearing a mask is an important measure for prevention of spreading the virus, hopefully such system could be beneficial for developing new tools and technologies for the future pandemics.

REFERENCES

[1] S. Al Hajjar, Z. A. Memish, and K. McIntosh, "Middle east respiratory syndrome coronavirus (mers-cov): a perpetual challenge," *Annals of Saudi medicine*, vol. 33, no. 5, pp. 427–436, 2013.

[2] M. Bchetnia, C. Girard, C. Duchaine, and C. Laprise, "The outbreak of the novel severe acute respiratory syndrome coronavirus 2 (sars-cov-2): A review of the current global status," *Journal of infection and public health*, 2020.

[3] J. Chen, M. Glover, C. Li, and C. Yang, "Development of a user experience enhanced teleoperation approach," in *2016 International conference on advanced robotics and mechatronics (ICARM)*. IEEE, 2016, pp. 171–177.

[4] C.-L. Chung, D.-B. Chen, and H. Samani, "Action detection and anomaly analysis visual system using deep learning for robots in pandemic situation," in *2020 International Automatic Control Conference (CACS)*. IEEE, 2020, pp. 1–6.

[5] W. Cullen, G. Gulati, and B. Kelly, "Mental health in the covid-19 pandemic," *QJM: An International Journal of Medicine*, vol. 113, no. 5, pp. 311–312, 2020.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. Ieee, 2005, pp. 886–893.

[7] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams *et al.*, "Recent advances in deep learning for speech research at microsoft," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 8604–8608.

[8] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[10] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[13] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[14] M. Jiang, X. Fan, and H. Yan, "Retinamask: A face mask detector," *arXiv preprint arXiv:2005.03950*, 2020.

[15] X. Jiang, T. Gao, Z. Zhu, and Y. Zhao, "Real-time face mask detection method based on yolov3," *Electronics*, vol. 10, no. 7, p. 837, 2021.

[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[17] C. Li, C. Yang, J. Wan, A. Annamalai, and A. Cangelosi, "Neural learning and kalman filtering enhanced teaching by demonstration for a baxter robot," in *2017 23rd International Conference on Automation and Computing (ICAC)*. IEEE, 2017, pp. 1–6.

[18] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[20] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "Fighting against covid-19: A novel deep learning model based on yolo-v2 with resnet-50 for medical face mask detection," *Sustainable cities and society*, vol. 65, p. 102600, 2021.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[22] P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria, and J. Hemanth, "Ssdmnv2: A real time dnn-based face mask detection system using single shot multibox detector and mobilenetv2," *Sustainable cities and society*, vol. 66, p. 102692, 2021.

[23] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3. IEEE, 2006, pp. 850–855.

[24] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, and J. Walsh, "Deep learning vs. traditional computer vision," in *Science and Information Conference*. Springer, 2019, pp. 128–144.

[25] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[26] F. Pérez-Hernández, S. Tabik, A. Lamas, R. Olmos, H. Fujita, and F. Herrera, "Object detection binary classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance," *Knowledge-Based Systems*, vol. 194, p. 105590, 2020.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[28] S. Wu, S. Zhong, and Y. Liu, "Deep residual learning for image steganalysis," *Multimedia tools and applications*, vol. 77, no. 9, pp. 10 437–10 453, 2018.

[29] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” *arXiv preprint arXiv:1411.1792*, 2014.