Proceedings of the 2023 International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/8/20230295

# Using convolutional neural network for detection of face mask

**Jiajin Yang**

Department of Information and Computing Science, Xi'an Jiaotong-liverpool University, Suzhou, 215126, China

Jiajin.Yang20@student.xjtlu.edu.cn

**Abstract.** The COVID-19 pandemic has had a sweeping impact across the globe, resulting in enormous economic losses and significant changes to people's way of life. Despite the World Health Organization's (WHO) assertion that the COVID-19 pandemic will conclude by 2023 and people's lives will begin to settle down, the possibility of a resurgence of the virus cannot be overlooked. In crowded public places, it is essential to have a system that can rapidly and accurately detect whether people are wearing masks and adhering to proper usage protocols. Therefore, it is crucial to make necessary preparations for such a system. Fortunately, there have been notable advancements in facial recognition technology, which can aid in this endeavor. This paper aims to build a model for face-mask-detection with convolutional neural network to help perform a rapid mask-wearing check and carry out the system model training with a final accuracy reaching 0.95. In the end, high accuracy is observed in the classification of correctly wearing masks and incorrectly wearing masks, demonstrating that the model is capable to identify whether people wear masks and wear correctly with the final accuracy reaching 0.97.

**Keywords:** facial recognition, face mask detection, convolutional neural network.

## 1. Introduction

COVID-19's sudden invasion has had a huge impact on the global economy. For instance, in China, while all levels of government spent a combined 80.55 billion CNY, the central government was responsible for 17.29 billion CNY of that total. More than 300 billion CNY has been invested in assisting many places, purchasing medical and medical equipment needed for the prevention and control of the novel coronavirus pneumonia, building and requisitioning quarantine hospitals, and deploying human, material and financial resources for various screening and quarantine measures were taken across the country to stop the spread of the novel coronavirus pneumonia [1]. People's lives have completely transformed in a social sense. Humans are unable to congregate, and in congested areas, they are required to wear masks. Therefore, it is of great significance to complete facilities to prevent the return of the virus.

Machine learning has made gains in the face recognition field, but as the technology has advanced, they have been gradually replaced by deep learning based on convolutional neural networks [2]. The advantage of the neural network is that instead of designing specific features that are robust for different types of in-class differences (such as light exposure, posture, facial expression, age, etc.), this approach allows them to be learned from training data [3]. Nevertheless, there still exist challenges in

DOI: 10.54254/2755-2721/8/20230295

practical operation. Firstly, the data set is not large enough. Secondly, the pictures are taken from variable angles and multiple faces may appear in one image. Last but not least, the extent to which masks are worn incorrectly is difficult to establish. This study attempts to overcome the listed problems and succeed in building a model for face mask-wearing detection. Hopefully, the final accuracy can reach 0.95. To acquire a valid experimental outcome, there are a total of eight steps. Five of them are covered in the method part and the result part contains the rest of the steps.

## 2. Method

This section includes some basic information about the experimental process. Figure 1 shows the flow of the procedures.

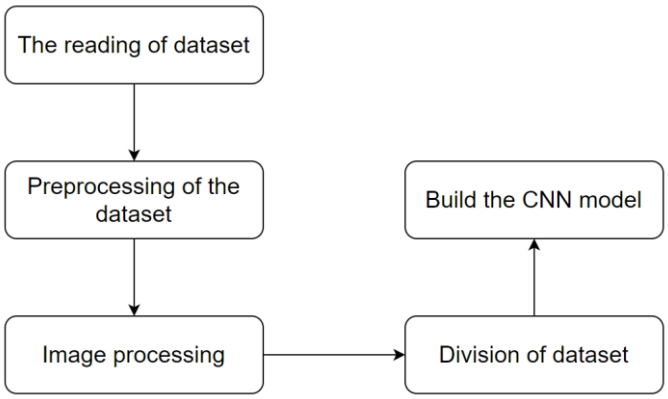



**Figure 1.** Experimental process.

### *2.1. Reading of the data set*

The first step is to do the data reading. The chosen data set is about face-mask-detection from Kaggle [4]. There are totally 851 colorful images and 4072 annotations corresponding to the images. The images are divided into three categories, with mask, without mask, and wearing mask incorrectly.

### *2.2. Preprocessing of the data set*

Since the annotations are in form of xml, they can not be read by persons. Translating annotations (XML) to a readable format such as CSV is helpful. The table has 8 columns,
representing (xmin), (ymin), (xmax), (ymax), name, file width and height from left to right, respectively. The first four columns identify the location and size of the box, which locks the position of the face. The last two columns stipulate the size of the picture. It is important to note that more than one face may appear in a picture, so a picture may have many lines of labels (one face for one label). That is why the number of annotations is larger than that of images. Thus images are necessary to be cropped to make sure that one image only has one face.

By cropping the images, the annotations can match the images one by one. Change three labels, the final labels are class 0 (incorrectly wearing mask), class1(correctly wearing mask), and class2(not wearing mask). The final data set fed into the model after processing has three columns. Annotation_file shows the label of the image, image_file shows which images the cropped image comes from and cropped_image_file represents a newly created file containing all the cropped images. There are totally 4072 images with a single face and 4072 corresponding annotations.

### *2.3. Image processing*

Since the cropping process will lead to pictures of different sizes, the picture specification is standardized. Images are uniformly set to 60×60 pixels. Then grayscale processing is carried out. The purpose of image graying is to simplify the matrix and improve the operation speed [5]. Normalized

DOI: 10.54254/2755-2721/8/20230295

processing is performed in the next step. As images are normalized, the pixel of the images falls in [0,1] [6]. At the same time, removing unnecessary parts helps to highlight individual differences. Finally, the image enhancement is employed.

Image_generator is used to reduce the effect of image flipping, rotation, and movement. To provide greater clarity, the rescale function standardizes the images by adjusting their scales. Horizontal and vertical flip functions enable the flipping of images along the horizontal and vertical axes, respectively. The zoom range function controls image scaling, while the shear range function controls image shearing. Width and height shift functions control horizontal and vertical offset, respectively, and the rotation range function controls image rotation.

### *2.4. Data division*

The last step before building the model is splitting the data. By using train_test_split, train set and test set can be split. The test set size is set to 0.25. Then 2424 images are gotten with the label with_mask ,538 images with the label without_mask, and 92 images with the label mask_worn_incorrect.

Iterating over the exercise image data set, as the picture shows, it is obvious that the distribution of quantities is uneven. So the identification of class0 (incorrectly wearing mask) in the model may be difficult. To solve this problem, Stratified_K_Fold is chosen to further split the training set and the validation set [7]. The idea of stratified random sampling makes the experimental data more reliable.

### *2.5. Build CNN model*

The input images are 60×60 pixels and the number of channels is three, so the input layer is firstly set as (60,60,3), which means the pictures have a width of 60, a height of 60, and RGB three-color channel. Then comes the first Conv2D layer with the 32 kernels whose size are 3. Conv2D is used to extract information from the input picture, which is called image features. The next layer is the Batch Normalization layer. In order to increase the gradient, speed up learning convergence, and prevent disappearing gradient issues, the increasingly biased distribution is pulled back to the standardized distribution using batch normalization. This causes the input value of the activation function to fall in the region where the activation function is sensitive to the input. Then follows the activation function. The activation function chosen is relu. Relu can make irrelevant features become not important in this model, preventing the probability of over-fitting. Max-pooling retain the main features while the parameters and calculation amount are reduced to prevent over-fitting and improve the model generalization ability.

The above four layers combine as a plate and there are four plates like this (the Input layer is counted as layer 0). The creation of the model repeats the above steps four times and set the size of convolutional kernels as 32,64,128,256. To fully connect with all neurons, flattening the 2-dimensional matrix to 1and dropping out 71% neurons on each plate with linear rectification function as activation function to prevent over-fitting is useful. Finally, the result is output by using softmax, which is a generalization of sigmoid [8].

DOI: 10.54254/2755-2721/8/20230295

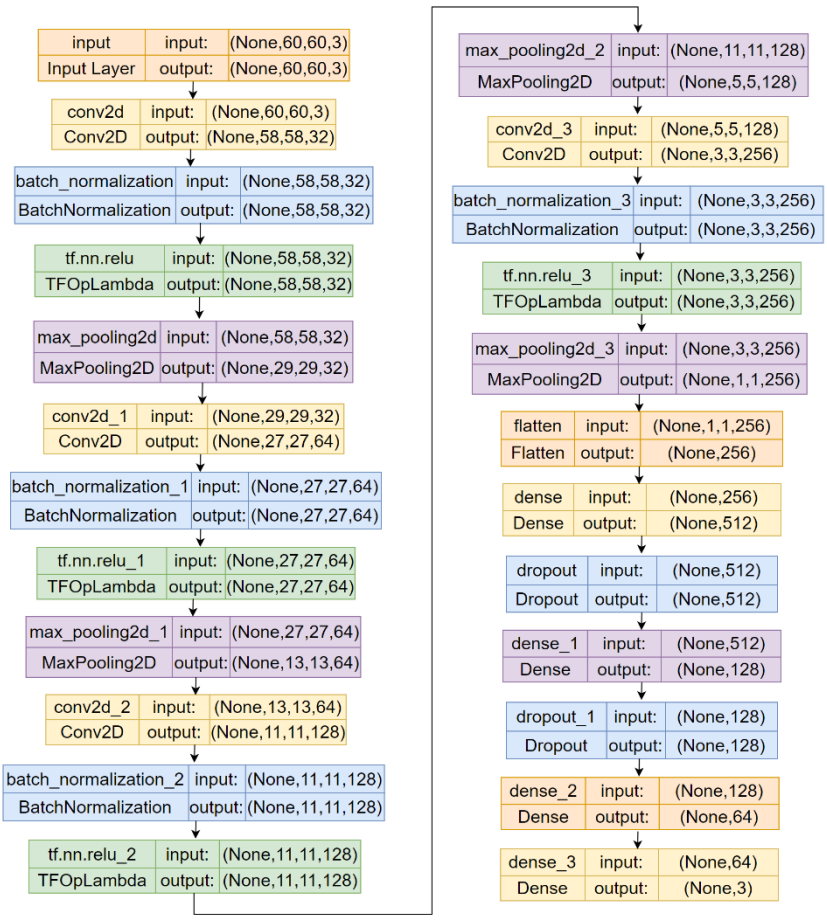



**Figure 2.** Convolutional neural network model diagram.

### *2.6. Evaluation metrics*

The criterion to judge the model is accuracy and loss functions. Metrics as the evaluation criteria generate the accuracy while loss functions output the loss rate. Accuracy and categorical accuracy are optional metrics while MSE and categorical crossentropy are optional loss functions. Accuracy outputs integer values and categorical accuracy outputs one-hot vectors.

*2.6.1. MSE.* The full name of MSE is mean square error loss function (formula 1) [9]. The purpose of MSE is to measure the distance between the sample point and the regression curve. By minimizing the square loss, the sample point fits the regression curve better. When the mean square error loss function (MSE) is smaller, the accuracy of prediction model is higher.

$$\mathrm{L}\left(Y \mid f(x)\right) = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - f(x_i)\right)^2 \tag{1}$$

*2.6.2. Categorical* crossentropy. The closeness between the actual output probability and the predicted output probability (formula 2) is described by categorical cross entropy, which states that the closer the two probability distributions are, the more efficiently gradient dissipation may be avoided.

$$\mathrm{L}\left(Y \mid f(x)\right) = -\sum_{i=1}^{n} Y_i \log f(x_i) \tag{2}$$

*2.6.3. Accuracy.* F1-Score is used to evaluate the model's performance in the test set [10]. A total of 7

DOI: 10.54254/2755-2721/8/20230295

parameters will be introduced. Accuracy (formula 3) is the ratio of the correct prediction sample size to the total sample size.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

*2.6.4. Precision.* In terms of prediction outcomes, precision(formula 4) indicates the proportion of genuine positive instances—that is, the proportion of positive cases that the binary classifier correctly predicted—among the positive examples it predicted.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

*2.6.5. Recall.* Recall(formula 5) specifies how many genuine positive examples in the test set are selected by the binary classifier, or more specifically, how many real positive cases are recalled by the binary classifier, from the standpoint of actual outcomes.

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

*2.6.6. F1-Score.* Since recall and accuracy cannot be used individually to assess a model, F1-Score is utilized to balance the two indications and make them compatible. F1-Score (formula 6) is the harmonic average of precision and recall.

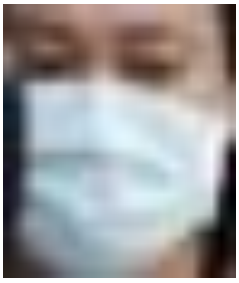
$$F_1 = \frac{2TP}{2TP + FP + FN} \tag{6}$$

## 3. Results

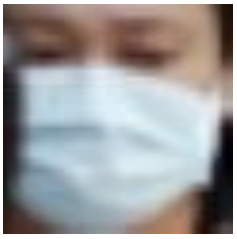
### *3.1. Processed images*

This section is going to show the comparison between the original image and the processed image. Figure 3 is from original images. It can be found that multiple faces appear in one image, and this picture cannot be put into the model as data obviously. Therefore, the images need to be processed.

**Figure 3.** Original images.



**(a).** Cropped Images



**(b).** Input images

**Figure 4**. These two images show the picture processing procedure. (a) is the cropped image and (b) is the final image after process.

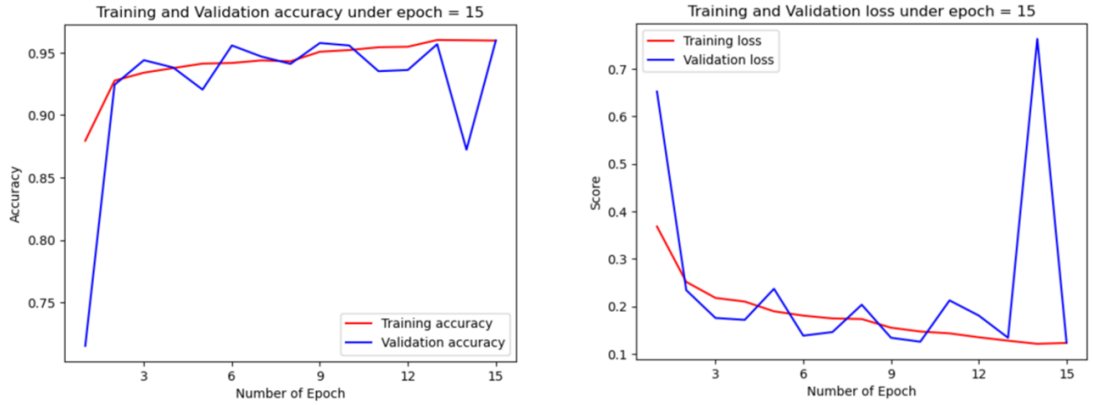
DOI: 10.54254/2755-2721/8/20230295

By cropping the ordinary image, figure 3 can be split into several small images like figure 4(a), which is a single face. Then by applying image enhancement and regularizing the size as 60 × 60, the final figure 4(b) comes out. There are in total of 4072 images in the form of figure 4(b) and they will be put into the train set.

### *3.2. Evaluation of the model*

Changing the type of optimizer, type of loss and the type of metrics to evaluate the model makes results more comprehensive. A validation set will be used to determine the selection of parameters. The optimizers are Adam, SGD, Adagrad and Adadelta. Loss functions are MSE and categorical crossentropy. Metrics are accuracy and categorical accuracy [11].

By using the controlling variables method, the epoch is firstly set 15. Figure 5(a) and figure 5(b) are one of the results when epoch = 15. The former is training and validation accuracy and the latter is training and validation loss. The chosen parameters are Adam, categorical crossentropy, and accuracy.




**(a)** **(b)**

**Figure 5.** These two pictures show the validation result when epoch is set 15. (a) represents training and validation accuracy under epoch = 15 and (b) represents training and validation loss under epoch = 15.

According to Figure 5(a) and figure 5(b), it is found that the highest accuracy occurs when the epoch equals 15. Repeat this step and the highest values that appear in each experiment will be filled in the bar chart. One optimizer will have four results since there are two metrics and two loss functions. As introduced above, there are total 4 optimizers. The level of accuracy will be shown in a bar chart. Label 1 is MSE and accuracy, label 2 is categorical crossentropy and accuracy, label 3 is MSE and categorical accuracy and label 4 is categorical crossentropy and categorical accuracy. The former is loss function and the latter is evaluation metric.

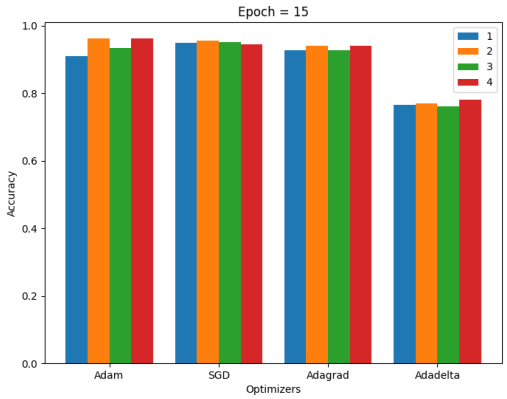
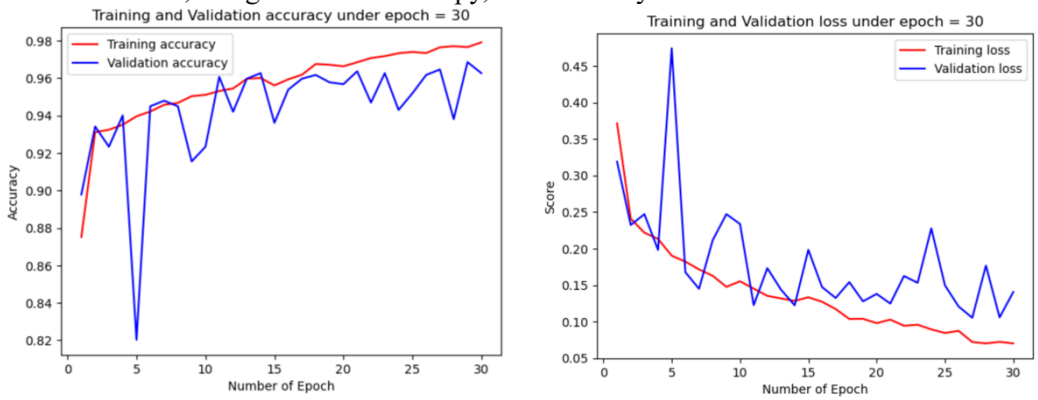



**Figure 6.** Comparison of testing accuracy under different optimizers with epoch 15.

DOI: 10.54254/2755-2721/8/20230295

According to figure 6, it is shown that that when epoch = 15 the best performance under parameter selection is optimizer: Adam, loss: categorical crossentropy, metrics: accuracy. The best accuracy reaches 0.9627. Figure 7(a) and figure 7(b) are one of the results when the epoch equals 30. The former is training and validation accuracy and the latter is training and validation loss. The chosen parameters are Adam, categorical crossentropy, and accuracy.




**(a)** **(b)**

**Figure 7.** These two pictures show the validation result when epoch is set 30. (a) represents training and validation accuracy under epoch = 30 and (b) represents training and validation loss under epoch = 30.

It is found that when the epoch equals 29, the best accuracy is 0.9715. Repeat this step and the highest values that appear in each experiment will be filled in the bar chart. Also label 1 is MSE and accuracy, label 2 is categorical crossentropy and accuracy, label 3 is MSE and categorical accuracy and label 4 is categorical crossentropy and categorical accuracy. The former is loss function and the latter is evaluation metric.

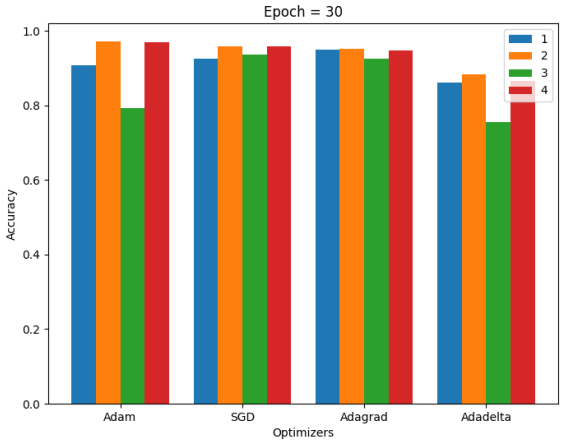



**Figure 8.** Comparison of testing accuracy under different optimizers with epoch 30.

From figure 8 shown above, it is observed that when optimizer = Adam, loss = categorical crossentropy, and metrics = accuracy, the model has the best performance with the highest accuracy when epoch is set to 30. Since the highest accuracy of epoch=30 is larger than that of epoch=15 and when epoch is set to 30, the model is more stable. So the best parameters are optimizer: Adam, loss function: categorical crossentropy, metrics: accuracy and epoch = 30. Then the model will be put into the test set for the performance.

### *3.3. Performance visualization*

Confusion matrix and heat map are used to demonstrate the degree of the accuracy of model

DOI: 10.54254/2755-2721/8/20230295

identification [12][13]. Each column of the confusion matrix represents the prediction category, and the total number of each column represents the number of data predicted for that category. Each row represents the real category of data, and the total number of data in each row represents the number of data instances of that category. The numbers in each column represent the number of real data predicted for that class.

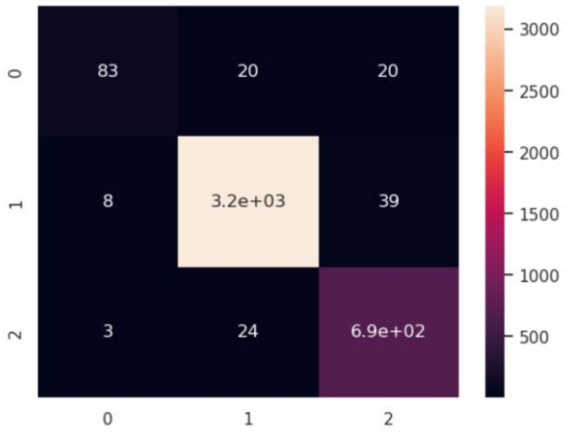



**Figure 9.** Confusion matrix with epoch 30.

The model's performance in each category may be tracked using a confusion matrix, which can also be used to determine the model's accuracy and recall rate for each category. The confusion matrix may also be used to identify categories that are challenging to distinguish between, which is useful for future improvement. As figure 9 shows, the abscissa of the confusion matrix is the predicted category and the ordinate is the true category. As the picture shows that 3200 images are identified as wearing mask, 690 images are identified as without mask and 83 images are identified as wearing mask incorrectly. The corresponding scores will be listed and charted below.

**Table 1.** Test Result.

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Mask_worn_incorrect | 0.88 | 0.67 | 0.76 | 123 |
| With_mask | 0.99 | 0.99 | 0.99 | 3232 |
| Without_mask | 0.92 | 0.96 | 0.94 | 717 |
| Accuracy | / | / | 0.97 | 4072 |
| Macro avg | 0.93 | 0.87 | 0.90 | 4072 |
| Weighted avg | 0.97 | 0.97 | 0.97 | 4072 |

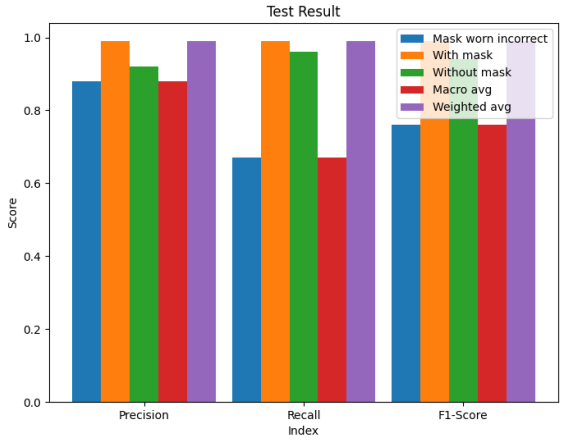


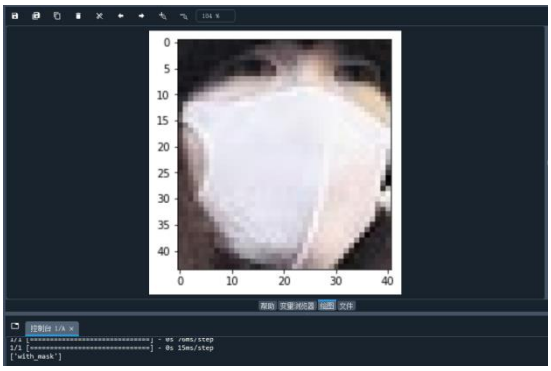

**Figure 10.** Test Result.

DOI: 10.54254/2755-2721/8/20230295

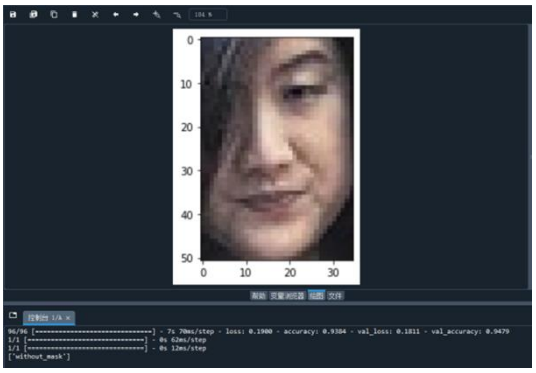
As mentioned above, the precision of class “mask_worn_incorrect”, class “with_mask” and class “without_mask” are 0.88,0.99,0.92 respectively. Recall are respectively 0.67,0.99,0.96 and F1-Score are 0.76,0.99,0.94. Support refers to the number of images belonging to each class. There are 123 images belonging to class0(incorrectly wearing mask), 3232 images belonging to class1(correctly wearing mask), and 717 images pertain to class2(not wearing mask). The accuracy is 0.97. Macro avg, in more precise terms, is the average value of all indicators for all categories. The product of the percentage of category samples in the total samples and the related index is what makes up the weighted average. The precision, recall and F1-Score of macro avg are 0.93,0.97,0.90 while the those of weighted avg are 0.97,0.97 and 0.97. Figure 10 is a bar chart drawn from table 1, it can visualize the value more intuitively. The x-coordinate is three indexes: precision, recall and F1-Score. The y-coordinate is score. The five pillars from left to right are “mask_worn_incorrect”, “with_mask”, “without_mask”, “macro avg” and “weighted avg” respectively.

### *3.4. Application*

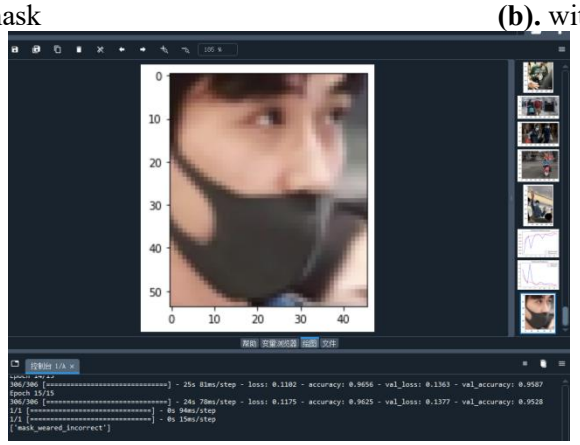
The last part of the results is the validation of the model. Class0 represents those who wear mask incorrectly and Class1 represents those with masks. Class2 corresponds to images without masks. Choose three pictures respectively corresponding to class0(incorrectly wearing mask), class1(correctly wearing mask), and class2(not wearing mask) to see whether actual results match the expected one. The images are selected from the data set, or people can post photos of themselves to validate the model.



**(a).** with_mask



**(b).** without_mask



**(c).** mask_worn_incorrect

**Figure 11.** These three pictures are the validation images. (a) is chosen from the “with_mask” class and (b) is selected from “without_mask” class. (c) belongs to “mask_worn_incorrect” class.

The validation result shows that the model can identify which class the selected picture belongs to according to figure 11 since figure 11(a) is recognized as “with_mask”, figure 11(b) is recognized as “without_mask” and figure 11(c) is identified as “mask_worn_incorrect”. Thus, it is feasible to apply this model to mask recognition.

DOI: 10.54254/2755-2721/8/20230295

## 4. Discussion

The criterion to determine the model parameter selection is accuracy under four optimizers. Adam is a gradient descent technique version which is employed to update the neural network's weight. The gradient descent algorithm used in SGD contains noise. SGD is also unlikely to converge and will instead always oscillate towards the minimal value. Another kind of gradient variation is called adagrad. Its fundamental tenet is that, in order to avoid shock, a parameter's corresponding learning rate will decrease if its gradient is consistently extremely large. Nevertheless, if a parameter's gradient is consistently extremely tiny, its learning rate will increase and it can be updated more quickly. Adadelta is a modification of Adagrad that simplifies the computation and places adaptive limitations on the learning rate. The experimental results show that Adam is more suitable to be the optimizer of this model.

Then the model is built and applied to testing after parameters determination. Though incorrect wearing of masks detection has low accuracy, the CNN model achieves 0.97 accuracy in total. Attention is also paid to F1-Score since F1-Score is progressive. F1-Score of class1 and class2 are 0.99 and 0.94 respectively. While in the identification of class0,83 of 123 images are identified correctly, so the F1-Score is 0.76. In terms of the performance of the whole model, weighted avg is selected as the final F1-Score on account of weighted average is a further reinforcement of macro average. The final F1-Score is around 0.97.

## 5. Conclusion

Based on the data analysis, it is observed that the accuracy of the model improves consistently while the loss decreases during the training process, representing that the trained model is not prone to over-fitting. Experimental results indicate that Adam is the most effective optimizer compared with SGD, Adagrad, and Adadelta. Although the CNN model achieves an accuracy of 0.76 in classifying the category of wearing masks incorrectly, it demonstrates an overall accuracy of 0.97 in classifying datasets with masks, without masks, and wearing masks incorrectly. Specifically, it achieves a high accuracy of 0.99 and 0.94 in classifying datasets with masks and without masks, respectively, representing a comparably high level of classification performance.

However, the model still needs to be optimized (various parameters, number of layers). Maybe it only performs well in this data set and the performance in other datasets is uncertain. In addition, though the model can identify those who wear masks incorrectly, it still has errors, just as the heat map shows. More attention needs to be paid to improving the accuracy of class 2 recognition. After stratified sampling, it can identify pictures of incorrectly wearing mask, but was observed relatively low classification accuracy. The next step is to extract class 2 pictures from the original data set and feed them into the model repeatedly to increase the proportion of class2 pictures in the whole data set. Alternatively, images of incorrect mask-wearing can be extracted from other data sets and added to this data set.

## References

[1] Tang Renwu,Li Chuqiao & Ye Tianxi. (2020). The damage of the new coronavirus pneumonia epidemic to China's economic development and countermeasures. Economics and Management Research(05),3-13.] doi:10.13502/j.cnki.issn1000-7636.2020.05.001.

[2] Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science (New York, N.Y.), 349(6245), 255–260. https://doi.org/10.1126/science.aaa8415

[3] Trigueros, D. S., Meng, L., & Hartnett, M. (2018). Face recognition: From traditional to deep learning methods. arXiv preprint arXiv:1811.00116.

[4] Face Mask Detection. URL: https://www.kaggle.com/datasets/andrewmvd/face-mask-detection, 2022.

[5] X.Zhang and X.Wang. (2016). Novel Survey on the Color-Image Graying Algorithm. *2016 IEEE International Conf. on Computer and Information Technology (CIT)*, pp. 750-753.

DOI: 10.54254/2755-2721/8/20230295

[6] Alnowami Majdi et al. (2022). MR image normalization dilemma and the accuracy of brain tumor classification model. *Journal of Radiation Research and Applied Sciences*, *15*(3), pp. 33-39.

[7] S.Yadav and S.Shukla. (2016). Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification. IEEE 6th International Conf. on Advanced Computing (IACC), pp. 78-83.

[8] S. I. Yudita, T. Mantoro and M. A. Ayu. (2021). Deep Face Recognition for Imperfect Human Face Images on Social Media using the CNN Method. 2021 4th International Conf. of Computer and Informatics Engineering (IC2IE), pp. 412-417.

[9] Deng Jianguo, Zhang Sulan, Zhang Jifu, Xun Yaling & Liu Aiqin.(2020). Loss function and its application in supervised learning. Big Data (01),60-80.

[10] G. Saranya, D. Sarkar, S. Ghosh, L. Basu, K. Kumaran and N. Ananthi. (2021).Face Mask Detection using CNN. 2021 10th IEEE International Conf. on Communication Systems and Network Technologies (CSNT), pp. 426-431.

[11] Parekh Disha and Dahiya Vishal. (2021). Predicting breast cancer using machine learning classifiers and enhancing the output by combining the predictions to generate optimal F1-score. Biomedical and Biotechnology Research Journal (BBRJ), 5(3), pp. 331-334.

[12] Valero-Carreras Daniel and Alcaraz Javier and Landete Mercedes. (2023). Comparing two SVM models through different metrics based on the confusion matrix. Computers and Operations Research, 152.

[13] Almonacid, C., Fitas, E., Sánchez-Covisa, J., Gutiérrez, H., & Rebollo, P. (2023). Geographical differences in the use of oral corticosteroids in patients with severe asthma in Spain: heat map based on existing databases analyses. BMC pulmonary medicine, 23(1), 3. https://doi.org/10.1186/s12890-022-02295-2