Face Mask Detection

Group NS_14

Hyeyun Park 40051730

(Data Specialist, Compliance Specialist)

Junting Ye 40082557

(Training Specialist, Evaluation Specialist)

Comp 472 AA

June 22th, 2022

github.com/yjt1018439570/472-Mask-Part-2

0. Main changes

In the previous part of the project, we evaluate the AI using accuracy, precision, recall and F1-measure, and a confusion matrix. In this part, we will first improve our model and training from the first part. Then we will analyze and eliminate possible bias. And lastly, we will compare the old model and the new model using 10-fold cross-validation technique. The main changes we make since part one of the project is that we increase the size of the filter by one time. And we also increase the epochs from 10 to 20. We tried to add more convolutional blocks, but the improvement is not obvious. Before the changes in the dataset, the changes we make in the model and training process successfully increase the accuracy from about 60% to 75%. And after the size increase in the dataset, we successfully increase the number to about 80% of accuracy.

1. Dataset

This project used several sites dataset. We mainly used Kaggle to collect the dataset. In the data pre-processing step, we use ToTenser() to normalize the image data, and re-size it to 32*32 pixels. The resolution of the images in the dataset is not very consistent due to the limit resources online, they can be 7KB to 1MB. Therefore, it is better to use 32*32 pixels instead of a larger size to balance the solution different in the dataset.

The data is categorized into four classes (1) Cloth Mask, (2) N95 Mask, (3) No face Mask, (4) Surgical Mask

(1) Cloth Mask

(2) N95 Mask

(3) No face Mask

(4) Surgical Mask

Figure 1: Image categorized

The images were divided randomly into train/test/val set using 80/10/10 split.

The table below shows the total size of the dataset of each class.

	Number of Images
Cloth Mask	943
N95 Mask	952
No face Mask	992
Surgical Mask	929
Total	3,816

Table 1: Number of Images by classes

Gender Bias Dataset Division (Female)

	Number of Images
Cloth Mask	484
N95 Mask	477
No face Mask	496
Surgical Mask	479
Total	1,936

Gender Bias Dataset Division (Male)

	Number of Images
Cloth Mask	459
N95 Mask	475
No face Mask	496
Surgical Mask	450
Total	1,880

Table 2: Number of Images by bias

2. CNN Architecture

CNN is called Convolutional Neutral Network and it is one of the best machine learning models for image processing. CNN uses multi-layer perceptron to do computation and CNN uses many different filters to train the model. Therefore, it takes relatively little pre-processing on the image compare to other models.

And in the training process, we trained 20 epochs and it takes about 5 minutes to finish. The train loss graph is shown below.

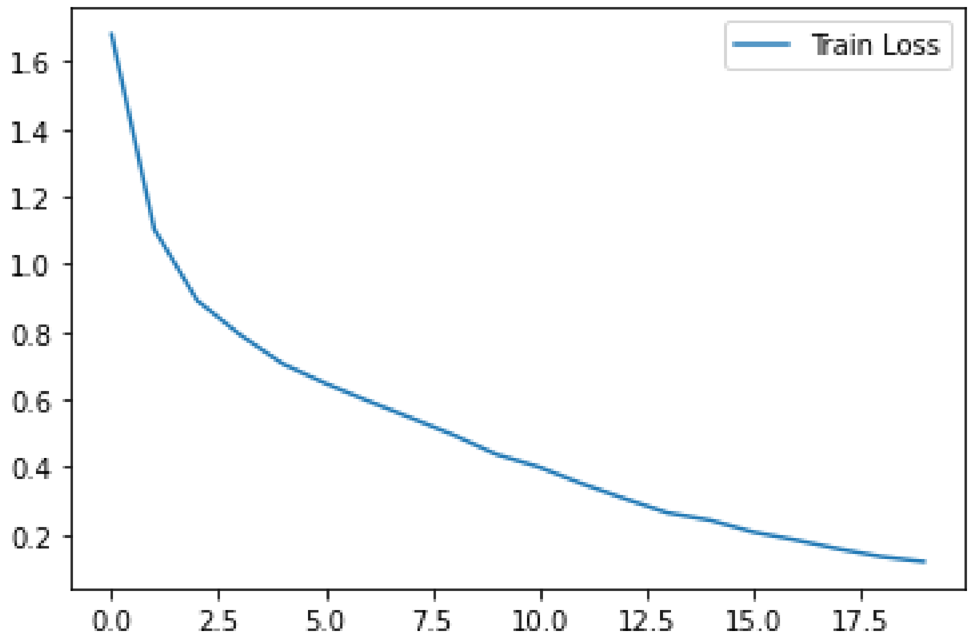

Figure 2: Epoch and Loss Output of Model

For the model in the project, the basic structure is this:

Convolution – pooling – normalization

Convolution – pooling – normalization

Convolution – pooling – normalization

Flattened - dropout – dense - dropout- dense – dropout - dense – output

Layer 1 is a Convolution layer which has 64 filters, with padding=1, The kernel size is 3*3. The activation function used is ’relu’.

Layer 2 is a Batch normalization layer.

Layer 3 is Maxpooling layer with pool size 2*2.

Layer 4 is a Convolution layer which has 128 filters, with padding=1, The kernel size is 3*3. The activation function used is ’relu’.

Layer 5 is a Batch normalization layer.

Layer 6 is Maxpooling layer with pool size 2*2.

Layer 7 is a Convolution layer which has 256 filters, with padding=1, The kernel size is 3*3. The activation function used is ’relu’.

Layer 8 is a Batch normalization layer.

Layer 9 is Maxpooling layer with pool size 2*2.

Layer 10 is a flattened layer to form a fully connected neural network or dense layer.

Layer 11 is a dropout of 0.12 is added to avoid overfitting.

Layer 12 is a dense layer useing activation function ’relu’.

Layer 13 is a dropout of 0.8 is added to avoid overfitting.

Layer 14 is a dense layer useing activation function ’relu’.

Layer 15 is a dropout of 0.6 is added to avoid overfitting.

Layer 16 is a dense layer useing activation function ’relu’.

Layer 17 is a dense layer and also the final output layer.

3. Evaluation

The table of Precision, recall, F1-measure, accuracy and confusion matrix of the new model will be shown below.

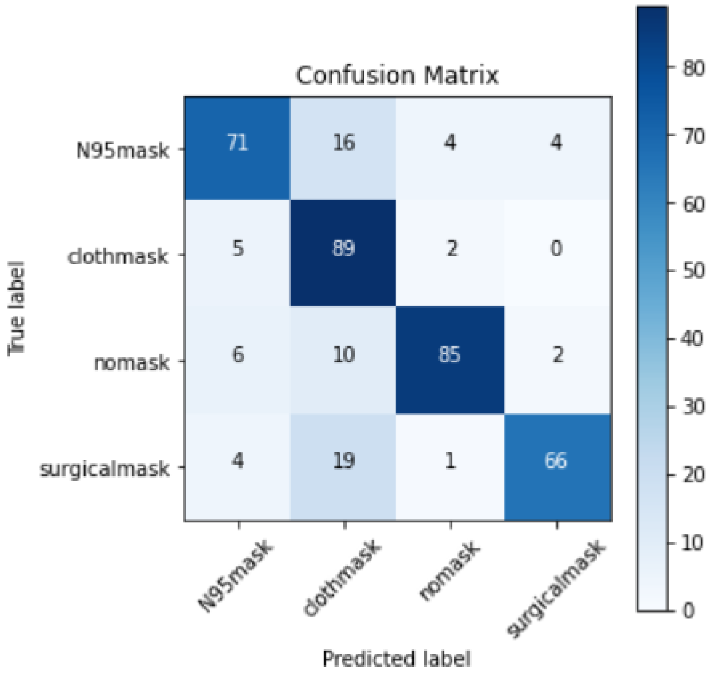

Figure 3. Confusion Matrix of the New Model

```
Accuracy:  0.810
Precision:  0.833
Recall:  0.810
F1-measure:  0.812
```

Figure 4. Result

The result of the new model and train is about 80% accuracy. Compare with the previous result, about 60%, it can be called a great improvement.

4. Bias

In this part, we analyze if the result consists of bias by gender. We evaluate the performance across all the classes in the dataset with different gender. With the old model and old dataset, we use extract about 100 images from each class with different gender (i.e., total of 400 images for female, and total of 400 images for male) as sample, and we find there is about a 4% difference in the accuracy between male and female.

4.1 Bias Evaluation for The Old Dataset

Female

```
Accuracy:  0.392
Precision:  0.437
Recall:  0.392
F1-measure:  0.407
```

Figure 5. Bias evaluation on the old model (female)

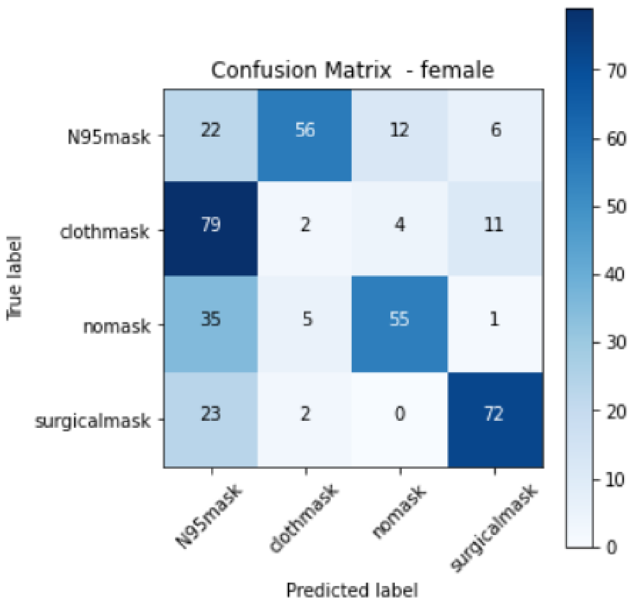

Figure 6. Confusion matrix with bias evaluation on the old model (female)

Male

```
Accuracy:   0.426
Precision:   0.477
Recall:   0.426
F1-measure:   0.444
```

Figure 7. Bias evaluation on the old model (male)

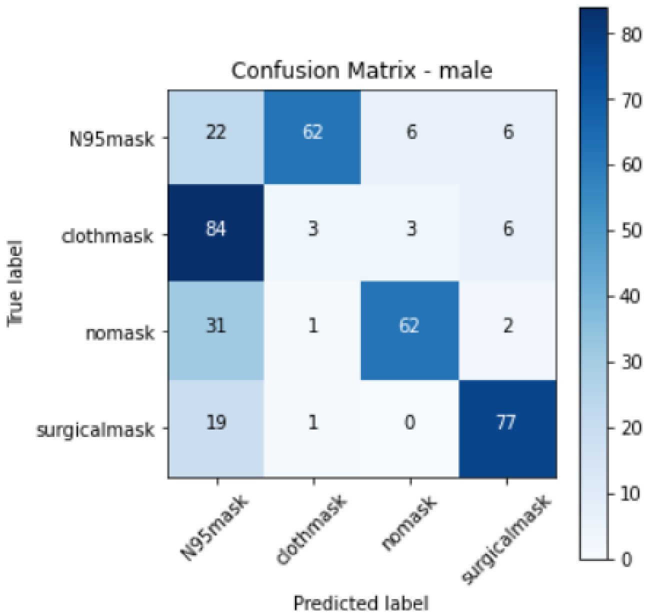

Figure 8. Confusion matrix with bias evaluation on the old model (male)

4.2 Bias Elimination

We can see from the above result, male has about 4% higher accuracy than female. We mentioned above that we increase the size of the dataset to improve accuracy. In this part, to eliminate the bias, we separated the whole dataset by gender, then we add images to make the two genders balanced in each class. By doing this, we successfully eliminate the bias, since the resulting difference between male and female has been reduced to less than 0.1%. The detail of the dataset will be mentioned below.

4.3 Bias set after re-balance

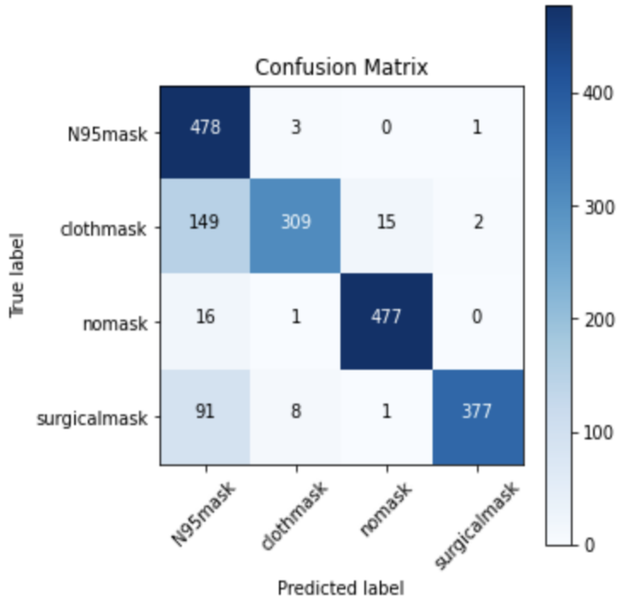

Figure 9. Confusion Matrix of New Model (Female)

```
Accuracy:  0.851
Precision:  0.893
Recall:  0.851
F1-measure:  0.853
```

Figure 10. Result of Classification (Female)

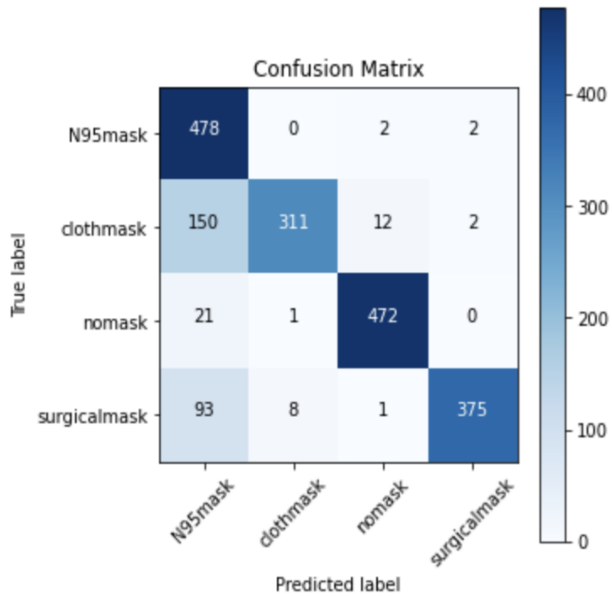

Figure 11. Confusion Matrix of New Model (Male)

```
Accuracy:  0.849
Precision:  0.894
Recall:  0.849
F1-measure:  0.851
```

Figure 12. Result of classification (Male)

We finalize the size of the dataset with a total of 1936 Female images and 1880 Male images. The accuracy of the old dataset, and old model was 39.2% vs. 42.6%, for female vs. male, respectively. With improvement in the dataset and model, the accuracy of the bias sets was improved to 85.1% vs. 84.9%.

5. K-fold cross-validation

The k-fold cross-validation randomly split the dataset into k package and perform k times of training. Each time it uses one package as test set, and uses the rest k-1 packages as train set. Then the result will be the average of the result from the k-times training.

Fold	Accuracy
fold 1	94.56140351
fold 2	93.85964912
fold 3	94.29824561
fold 4	94.73684211
fold 5	95.67251462
fold 6	96.28654971
fold 7	93.88888889
fold 8	94.12280702
fold 9	94.76608187
fold 10	93.85964912

Figure 13. Accuracy of 10-fold on new model

Fold	Accuracy
fold 1	60.74712644
fold 2	58.27586207
fold 3	54.71264368
fold 4	57.47126437
fold 5	59.88505747
fold 6	57.93103448
fold 7	52.93103448
fold 8	64.50867052
fold 9	54.85549133
fold 10	58.32369942

Figure 14. Accuracy of 10-fold on old model

	precision	recall	f1-score	support
N95mask	0.97	0.89	0.93	80
clothmask	0.92	0.98	0.95	91
nomask	0.99	0.92	0.96	92
surgicalmask	0.87	0.95	0.91	79
accuracy			0.94	342
macro avg	0.94	0.93	0.93	342
weighted avg	0.94	0.94	0.94	342

Figure 15. Aggregate statistics of 10-fold on new model

	precision	recall	f1-score	support
N95mask	0.23	0.21	0.22	42
clothmask	0.20	0.21	0.20	43
nomask	0.94	0.96	0.95	47
surgicalmask	0.95	0.93	0.94	41
accuracy			0.58	173
macro avg	0.58	0.58	0.58	173
weighted avg	0.58	0.58	0.58	173

Figure 16. Aggregate statistics of 10-fold on old model

Comparing the result, we can see that with the old model and old dataset, using classic split obtains about 60% of accuracy and using 10-fold cross validation obtains 58% of accuracy, which is very close. But for the new model and new dataset, using classic split obtains 80% of accuracy while using 10-fold cross validation obtains 94% of accuracy.

Our result shows that the 10-fold cross validation improves on the new model but not on the old model. Therefore, we conclude that a k-fold cross validation operation does not necessarily improve the accuracy, but using this operation can increase the reliability of the result.

Furthermore, the instruction of this project asks us to set the k value to 10, but we experiment on a few different k values. We find that a higher value of k tends to have higher accuracy as the outcome. And the reason is probably for higher value, the AI sees more data on the dataset. However, having the same amount of k on the same model can also produce different outcomes, we think it is because of the randomness of the split. In our dataset, there are images with very different quality, and there are also very similar images. Therefore, different splits on the dataset might give very different result outcomes.

Reference

A. Jangra. “Face mask detection ~12K images dataset,” *Kaggle*, 26 May, 2020.https://www.kaggle.com/datasets/ashishjangra27/face-mask-12k-images-dataset.

Chauhan, Shriraj. “Face Mask Detection Medical and Non-Medical Masks.” *Kaggle*, 20 Mar. 2021, https://www.kaggle.com/datasets/shrirajchauhan/face-mask-detection-medical-and-nonmedical-masks.

coffee124. “Face Mask.” *Kaggle*, 8 June 2022, https://www.kaggle.com/datasets/coffee124/facemaskn95?select=000105.jpeg.

“Confusion Matrix¶.” *Scikit*, https://scikit-learn.org/0.18/auto_examples/model_selection/plot_confusion_matrix.html.

Haddad, Jad. “How I Built a Face Mask Detector for Covid-19 Using Pytorch Lightning.” *Medium*, Towards Data Science, 9 June 2021, https://towardsdatascience.com/how-i-built-a-face-mask-detector-for-covid-19-using-pytorch-lightning-67eb3752fd61.

Jangra, Ashish. “Face Mask Detection ~12K Images Dataset.” *Kaggle*, 26 May 2020, https://www.kaggle.com/datasets/ashishjangra27/face-mask-12k-images-dataset.

J.Brownlee. “ How to Configure k-Fold Cross-Validation”, *Python Machine Learning*, July 31, 2020, https://machinelearningmastery.com/how-to-configure-k-fold-cross-validation/

Kyrie-leon. “Kyrie-Leon/Face-Mask-Classification-Dataset: 人脸口罩分类数据集（Face-Mask-Classification-Dataset，Fmcd）.” *GitHub*, https://github.com/Kyrie-leon/Face-Mask-Classification-Dataset.

Larxel. “Face Mask Detection.” *Kaggle*, 22 May 2020, https://www.kaggle.com/datasets/andrewmvd/face-mask-detection.

Mitra, Prithwiraj. “Covid Face Mask Detection Dataset.” *Kaggle*, 15 July 2020, https://www.kaggle.com/datasets/prithwirajmitra/covid-face-mask-detection-dataset.

Sumansid. “FaceMask Dataset”, *Kaggle*, June, 2020. https://www.kaggle.com/datasets/sumansid/facemask-dataset.

O.Gurav. “Face Mask Detection Dataset”,*Kaggle*,June, 2020. https://www.kaggle.com/datasets/omkargurav/face-mask-dataset.