Concordia University
Engineering and Computer Science

COMP 6721

Applied Artificial Intelligence

Project-1 Report

Submitted To: Prof. Dr. René Witte

Team Id: NS_04

Ananya Varsha (40197012) - Data Specialist

Manimaran Palani (40167543) - Evaluation Specialist

William Moses Stalin Jebadoss (40186129) – Training Specialist

1. Dataset:

The dataset used in this project is picked from different reference sites: Kaggle and Google Images [1]. The collected dataset is segregated into a dataset consisting of (1) Person without a face mask, (2) Person with a “community” (cloth) face mask, (3) Person with a “surgical” (procedural) mask, (4) Person with a “FFP2/N95/KN95”-type mask (you do not have to distinguish between them), and (5) Person with a FFP2/N95/KN95 mask that has a valve.

Following is the table representing dataset with number of images collected.

	Number of Images
No Mask	420
Cloth face mask	435
Surgical mask	410
N95 mask	402
N95 with valve	402
TOTAL	2069

Figure 1.1: Table representing Number of Images for each model

1.1 Dataset distribution for Gender Bias

	Number of Images
No Mask	157
Cloth face mask	157
Surgical mask	151
N95 mask	151
N95 with valve	162
TOTAL	778

Figure 1.2 Table representing Number of Images for Male

	Number of Images
No Mask	161
Cloth face mask	165
Surgical mask	164
N95 mask	172
N95 with valve	157
TOTAL	819

Figure 1.3 Table representing Number of Images for Female

	Number of Images
Male	778
Female	819
Total	1597

Figure 1.4 Table representing Total number of images for Gender

1.2 Dataset Distribution for Race Bias

	Number of Images
No Mask	89
Cloth face mask	92
Surgical mask	82
N95 mask	97
N95 with valve	90
TOTAL	450

Figure 1.5 Table representing Number of Images for GenZ Category

	Number of Images
No Mask	138
Cloth face mask	140
Surgical mask	142
N95 mask	133
N95 with valve	147
TOTAL	700

Figure 1.6 Table representing Number of Images for Millennial Category

	Number of Images
No Mask	111
Cloth face mask	118
Surgical mask	118
N95 mask	118
N95 with valve	117
TOTAL	582

Figure 1.7 Table representing Number of Images for Boomer Category

	Number of Images
GenZ	450
Millennial	700
Boomer	582
Total	1732

Figure 1.8 Table representing Total Number of Images for Age

2. CNN Architecture

2.1 Characterizing the model

In this case, we used a Convolutional Neural Network to train the model for certain classes. First and foremost, the provided dataset must be preprocessed to ensure that it is balanced and equal in size. Five different datasets were collected for each of the classes and labeled as ‘0’ for the person denoting no mask, ‘1’ for the person with cloth mask, ‘2’ for the person with surgical mask, ‘3’ for the person with N95 mask and ‘4’ for the person with N95 with valve mask.

Now we divide our dataset into two parts: training and testing, with training accounting for 80% of the data and testing accounting for the remaining 20%. To extract features from the images we have used 4 convolution layers with input size of [3*32*32] where 3 is the RGB channels and 32*32 is the height and width of the image. Each of these four convolution layers has a kernel size of (3,3), a stride of (1,1), and a padding of (1,1). The convolution layer is followed by the Batch Norm Layer which make neural networks faster and more stable through adding extra layers in a deep neural network. Its job is to take the outputs from the first hidden layer and normalize them before passing them on as the input of the next hidden layer. After batch normalization, there is an activation function. Here, we used ReLU as the activation function. There are two pooling layers, each after every 2 convolution layers. A filter of (2,2) and a maximum pooling with a step size of 2 were used. Padding helps hold more information around the edges of the image. Dropout functions are used to avoid overfitting the model. Finally, the output of the last pooling layer is flattened and fed as input to a fully connected layer. Neurons in the fully connected layer are connected to the activation in the previous layer. This helps to classify the data into multiple classes.

2.2 Preparing the Data

The dataset created is stored using Pickle file and torch vision is used for transformations that helps transforming the library in PyTorch. After resizing of images into (32,32), functions namely normalization and totensor is applied on the data. Split is tested using K Fold where k refers to the number of groups that a given data sample is to be split into. We have used K Fold because it results in a less biased or less optimistic estimate of the model skill than other methods.

2.3 Loading data using Data Loaders

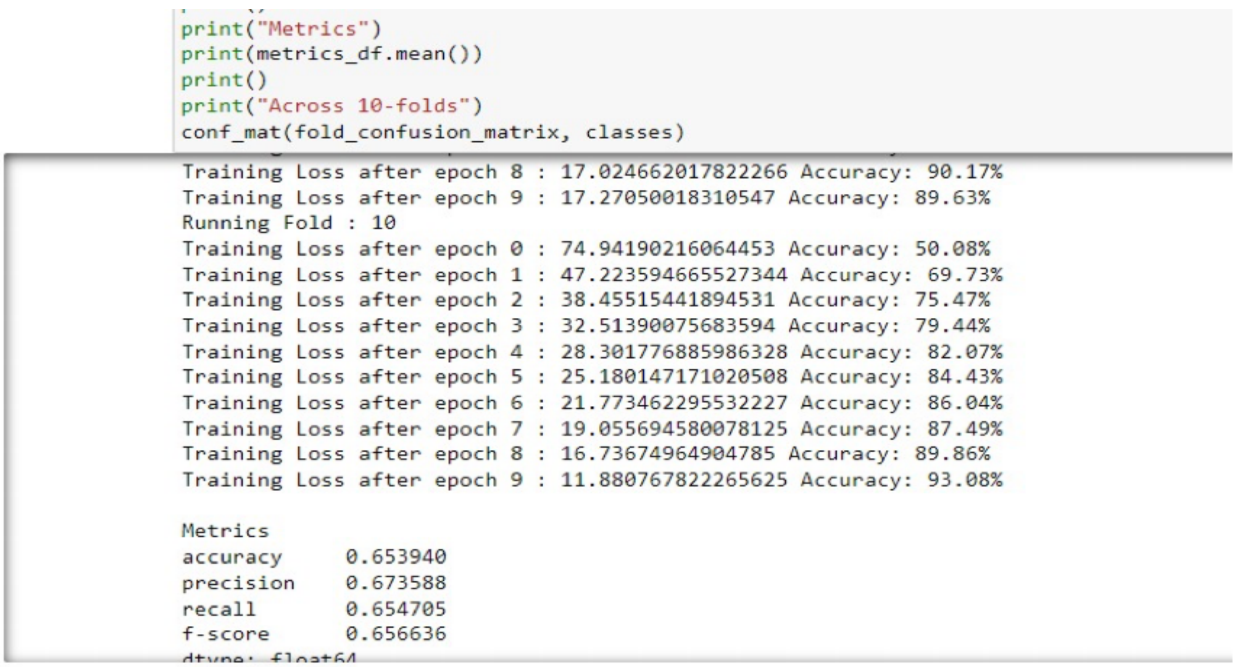
We have used Data Loaders for parallelizing the data loading process with automatic batching. With a batch size of 32, the neural network model is trained. When the Shuffle attribute is set to true, the model is better trained since it receives data that is not in a repeating pattern.

2.4 Modification using Optimizer Configuration

To modify the attributes of Neural Network, we have used the optimizer function to reduce loses. Adam optimizer is utilized, with a learning rate of 0.001.

2.5 Training the Model

Followed by optimizer configuration we train the data. The number of epochs is set to 10 and for each epoch the data is loaded using Data loader. The model is trained using face_mask_detector model as mentioned in 2.1. Entropy loss is used to calculate the loss value. Optimizer.zero_grad() function is used to set gradients to zero. Back propagation loss is calculated using backward() function. A screenshot is attached below which gives an idea about how CNN model is trained.



```
print("Metrics")
print(metrics_df.mean())
print()
print("Across 10-folds")
conf_mat(fold_confusion_matrix, classes)
```

```
Training Loss after epoch 8 : 17.024662017822266 Accuracy: 90.17%
Training Loss after epoch 9 : 17.27050018310547 Accuracy: 89.63%
Running Fold : 10
Training Loss after epoch 0 : 74.94190216064453 Accuracy: 50.08%
Training Loss after epoch 1 : 47.223594665527344 Accuracy: 69.73%
Training Loss after epoch 2 : 38.45515441894531 Accuracy: 75.47%
Training Loss after epoch 3 : 32.51390075683594 Accuracy: 79.44%
Training Loss after epoch 4 : 28.301776885986328 Accuracy: 82.07%
Training Loss after epoch 5 : 25.180147171020508 Accuracy: 84.43%
Training Loss after epoch 6 : 21.773462295532227 Accuracy: 86.04%
Training Loss after epoch 7 : 19.055694580078125 Accuracy: 87.49%
Training Loss after epoch 8 : 16.73674964904785 Accuracy: 89.86%
Training Loss after epoch 9 : 11.880767822265625 Accuracy: 93.08%

Metrics
accuracy     0.653940
precision    0.673588
recall       0.654705
f-score      0.656636
dtype: float64
```

Fig 2.1: Results after training

3. Evaluation

Following these three steps, we may assess the method or model's performance using the confusion matrix and numerous other matrices. The model is trained for ten epochs in this project. Learning occurs at a rate of 0.0001 in every epoch. For all stages, we used recall, precision, accuracy, and f1-measure to assess performance. We can state that some of the photos have been misclassified due to imbalance data by utilizing a confusion matrix. This suggests that we have a 3rd class (not a human) as a more generic one, where we may not have enough data. Which could be the model for all this muddle and misclassification.

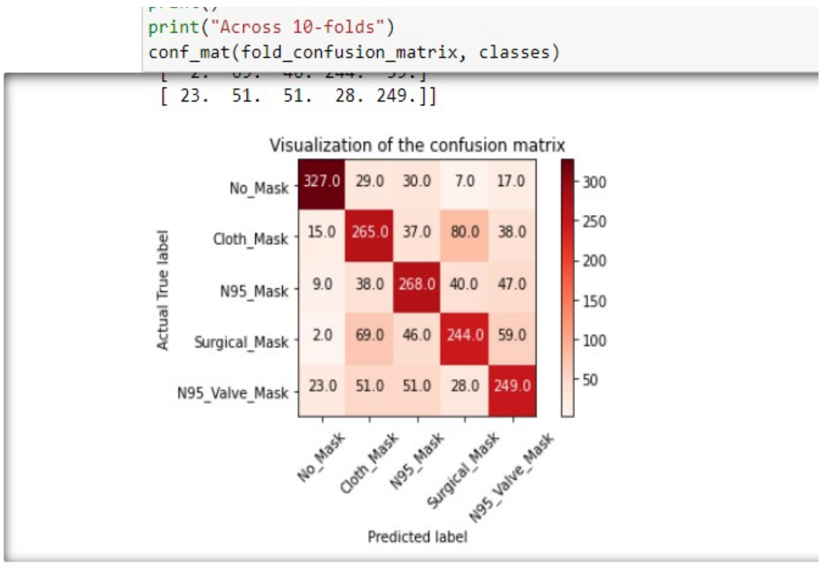


Fig 3.1: Visualization of Confusion Matrix

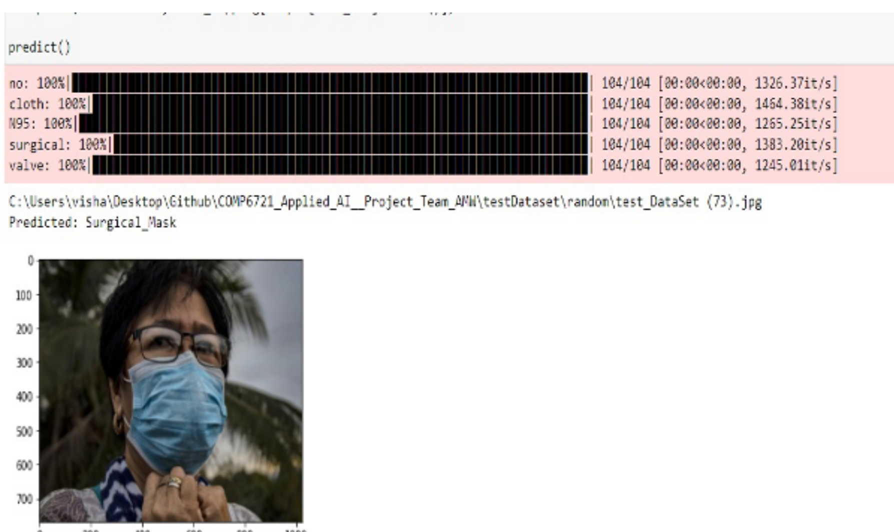


Fig 3.2: Test Result after training images

As previously stated, certain data are misclassified due to imbalanced data for each of the three classes. Humans with masks and without masks can be classified using numerous factors, however the non-human class is very broad, encompassing animals, non-living objects, statues, plants, and so on.

We will enhance the performance by adding more convolution layers as it will extract more features. Also, we can observe from the convolution matrix that the model doesn't predict well for Surgical Mask.

4 Bias Detection and Elimination

4.1 Bias Detection

4.1.1 Gender Bias

A discrepancy in the model's accuracy was discovered, after running the datasets of female masked images and male masked images separately on the dataset. The model was 63 percent accurate for males and 77 percent correct for females, demonstrating a male class selection bias in the attribute gender.

4.1.2 Age Bias

Running the dataset of young, middle, and old aged, masked images on the prior model yielded accuracies of 80%, 78 percent, and 77%, respectively, showing a slightly reduced representation for old aged, masked images with in previous complete dataset.

4.2 Bias Elimination

To eliminate the bias, additional male and female masked images have been added to the collection to create a balance between the different genders.

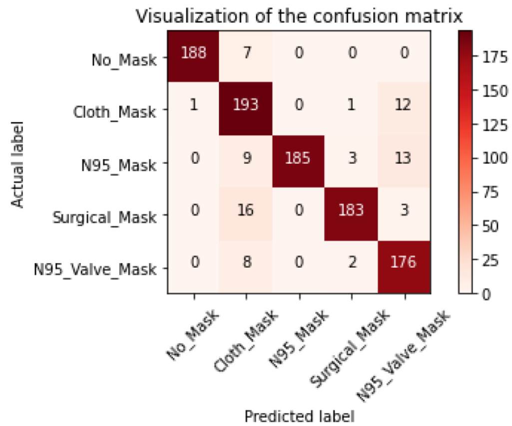
For a balanced dataset, extra photos of middle and old aged persons wearing masks were added to eliminate bias in age wise distribution.

Two more convolution layers have also been added to the model to improve accuracy. The model has been retrained using the new dataset and model, and it now performs much better.

4.3 Evaluation of Gender Bias

4.3.1 Female

When the dataset was run on old model, the resultant accuracy was 72%. The model was trained again after adding more layers and fine-tuning the hyper parameters, resulting in improved accuracy. The female masked dataset was run again on new model, where an increase in accuracy was observed. The resultant accuracy was 92%.

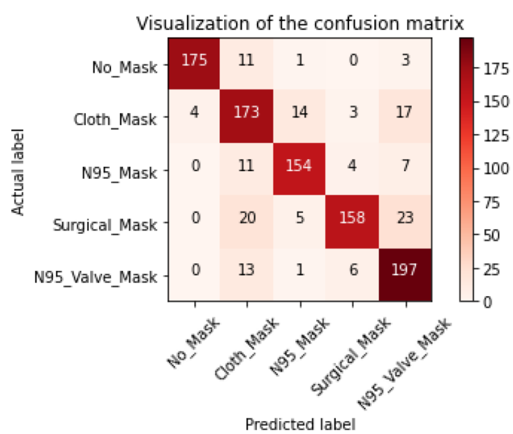


Fold results for females: (0.925, 0.9308068482519378, 0.9259198496025182, 0.9263547476674319)

Metric	Score
Accuracy	0.925
Precision	0.93080
Recall	0.92591
F1-score	0.92635

4.3.2 Male

When the dataset was run on old model, the resultant accuracy was 72%. The model was trained again after adding more layers and fine-tuning the hyper parameters, resulting in improved accuracy. The male masked dataset was run again on new model, where an increase in accuracy was observed. The resultant accuracy was 85%.



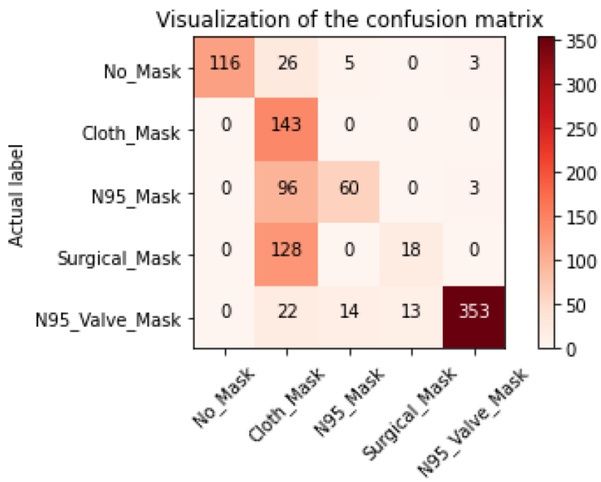
Fold results for males: (0.857, 0.8675946038998082, 0.8581564474987428, 0.860298295517733)

Metric	Score
Accuracy	0.857
Precision	0.86759
Recall	0.85815
F1-score	0.86029

4.4 Evaluation of Age Bias

4.4.1 GenZ Aged

When the dataset was run on old model, the resultant accuracy was 70%. The model was trained again after adding more layers and fine-tuning the hyper parameters, resulting in improved accuracy. The male masked dataset was run again on new model, where an increase in accuracy was observed. The resultant accuracy was 69%.

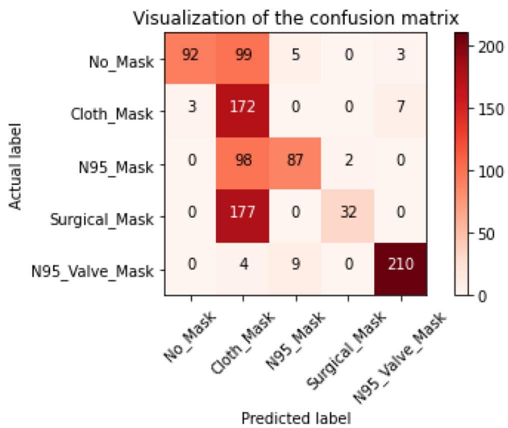


Fold results for GenZ Group:(0.69, 0.7336008107014809, 0.6304177895737133, 0.6040086881206033)

Metric	Score
Accuracy	0.69
Precision	0.73360
Recall	0.63041
F1-score	0.60400

4.4.2 Millennial Aged

When the dataset was run on old model, the resultant accuracy was 70%. The model was trained again after adding more layers and fine-tuning the hyper parameters, resulting in improved accuracy. The male masked dataset was run again on new model, where an increase in accuracy was observed. The resultant accuracy was 59%.

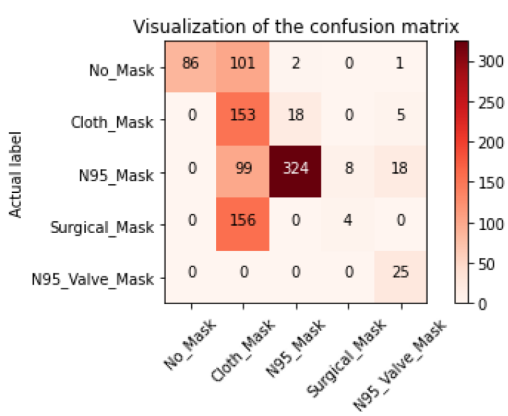


Fold results for Millennial Group: (0.593, 0.8076512778212805, 0.5934842456552898, 0.5822836223398076)

Metric	Score
Accuracy	0.593
Precision	0.80765
Recall	0.59348
F1-score	0.58228

4.4.3 Boomer Aged

When the dataset was run on old model, the resultant accuracy was 70%. The model was trained again after adding more layers and fine-tuning the hyper parameters, resulting in improved accuracy. The male masked dataset was run again on new model, where an increase in accuracy was observed. The resultant accuracy was 59%.



Fold results for Boomer Group: (0.592, 0.6171974542089875, 0.6137106648479875, 0.521848220179722)

Metric	Score
Accuracy	0.925
Precision	0.93080
Recall	0.92591
F1-score	0.92635

5 K Fold Evaluation

Here, the model is assessed using the stratified K-fold strategy on training and testing data. With the batch size of 32, the model will be reprised over 10 folds, each with 10 epochs. The evaluation on average accuracy, precision, recall and F1-score of these folds are done at the end. The visualization of confusion matrix is done using the function called plot_cm(). The model is saved as pickle file which gives us the image and its classification as a key value pair. This is then loaded using toarch.load() and trained again. The random set of images are picked, and the model is made to run on it and prediction is done on it. The outputs after training and prediction are shown below.

5.1 Old Model:

In part I of this project, we had used 4 convolution layers with the stride of (1,1) and a padding od (1,1). To ensure adequate class distribution, the model is assessed using stratified kfold evaluation rather than manual train test split. Ten folds were taken in total, with each fold including 10 epochs and a batch size of 32. Each epoch's accuracy and loss are calculated and displayed.

5.2 New Model:

Two more convolution layers with a stride of (1,1) and a padding of (1,1) have been added to improve the model's accuracy (1,1). The stratified k-fold approach with 10 folds is being used again, and the shuffle parameter is set to true to assure random shuffling. Again, ten folds were taken in total, with each fold including 10 epochs and accuracy, precision, recall, F1-neasure is calculated again.

5.3 New Model Results

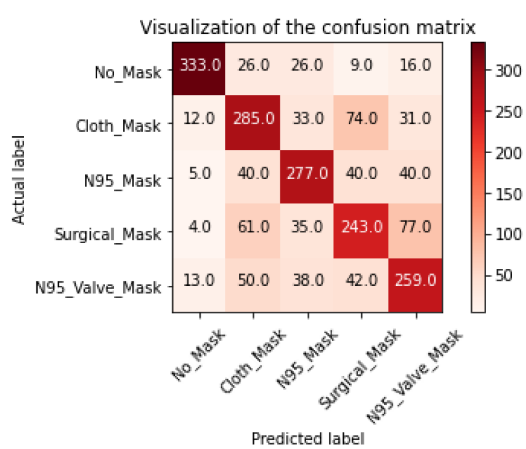


Figure 3.3 Confusion Matrix of entire dataset

Evaluation Metrics	New Model
Accuracy	0.675191
Precision	0.693819
Recall	0.675936
F1-measure	0.675427

5.4 Performance of Old Model and New Model

When compared the performance of new model against old model, it is observed that there is an increase in accuracy, precision, recall and F1-measure.

Evaluation Metrics	Old Model	New Model
Accuracy	0.653940	0.675191
Precision	0.673588	0.693819
Recall	0.654705	0.675936
F1-measure	0.656636	0.675427

References

[1] https://www.kaggle.com/omkargurav/face-mask-dataset
[2] https://www.kaggle.com/prithwirajmitra/covid-face-mask-detection-dataset
[3] https://www.kaggle.com/sumansid/facemask-dataset
[4] https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset
[5] https://www.kaggle.com/omkargurav/face-mask-dataset
[6] https://humansintheloop.org/resources/datasets/medical-mask-dataset/
[7] https://www.shutterstock.com/search/ffp2?number_of_people=1&mreleased=true
[8] https://www.gettyimages.ca/photos/ffp2-mask?assettype=image&sort=mostpopular&
[9] https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529