CONCORDIA UNIVERSITY

COMPUTER SCIENCE AND SOFTWARE ENGINEERING DEPARTMENT

COMP 6721- APPLIED ARTIFICIAL INTELLIGENCE

PROJECT ASSIGNMENT 2

SUBMITTED TO:

PROF. DR. RENE WITTE

SUBMITTED BY:

PAL PATEL (40197897) – Data Specialist

URVISH TANTI (40197508) – Training Specialist

SAKSHI PATEL (40194143) – Evaluation Specialist

1. Dataset

An assorted collection of data gathered from various resources to get some insights out, is called a dataset. These datasets are used to train an algorithm by finding divergent predictable patterns. It is important for AI to correctly interpret the data and result in an accurate model. For this project, we have collected data for 5 different categories. That is a Person wearing -1. No mask, 2. Community(cloth) mask, 3. Surgical mask, 4. N95 mask with a valve, and 5. FPP2 mask

1. No Mask

2. Community (Cloth) Mask

3. Surgical Mask

4. FFP2/N95 Mask

5. FFP2/ N95 Mask with a Valve

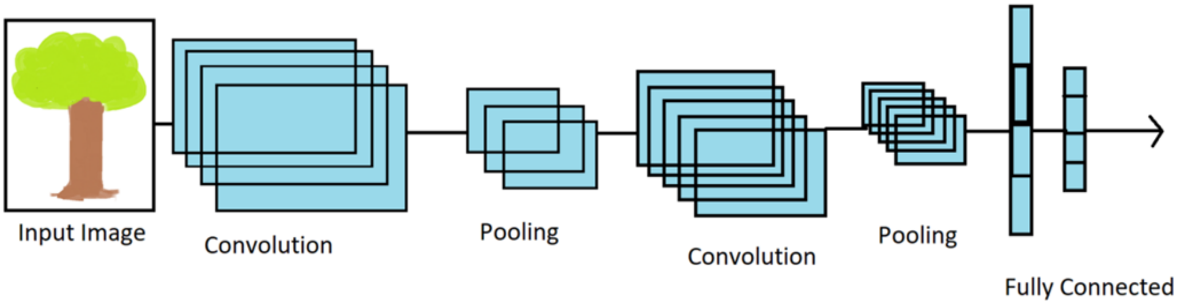
Data set statistics:

We have a total of **2982** images in our dataset. The class with **No Mask** contains **612** images, the **Community Cloth** class contains **607** images, the **Surgical Mask** class contains **592** images, and the **FFP2 Mask** class has **597** images. Lastly, the **FFP2 Mask with Valve** class contains **574** images. Thus, we have an evenly split dataset across all the classes so as to increase the training efficiency. We have split the dataset in the ratio of 70:30 for training and testing respectively.

2. CNN Architecture

Convolutional Neural Networks (CNN) is a special type of multi-layer neural network, designed to extract key features from an image as visual patterns. It requires very little pre-processing of data. We have used CNN to train our model in the given classes. Because we collected data from variegated resources, the images were in different sizes and resolutions. For this to be resolved, we converted all the images in the dataset to be of the same size. In order to change the pixel intensity, we have normalized our data using the mean and standard deviation. With the help of this, our speed of training increased drastically. Categorization of classes: {0: “no_mask”, 1: “ffp2_mask”, 2: “surgical _mask”, 3: “ffp2_with_valve”, 4: “cloth_mask”}. We then load these images using the loader method that takes up images in batches of 32. After this, the training and testing sets are loaded in the .npy file.\
CNN Details:
Total Layers - 4
Images are passed using kernel size (3*3). Stride, Padding =1.
Layer 1: Input Channels - 3, Output Channels - 6
Layer 2: Input Channels - 64, Output Channels - 128
Layer 3: Input Channels - 128, Output Channels - 256
Layer 4: Input Channels - 256, Output Channels - 512

After declaring the input and output channels for each layer, the data shall be normalized. It will be then passed onto ReLU (Activation function). After this process, MaxPooling will be run.
After this, we will have a fully connected layer to categorize the extracted features from the data into their classes. In the end, our training model would be ready.



6. CNN Architecture

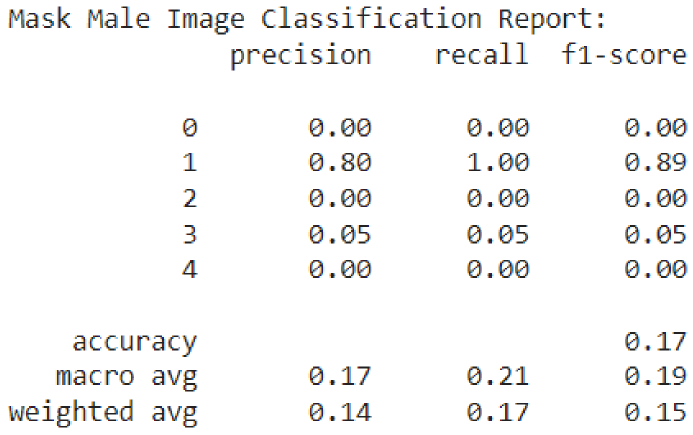
Hyperparameters:
• Optimizer: Adam’s optimizer has been used as it converges faster than gradient descent.
• Learning Rate: A learning rate of 0.0001 is used for training. After a number of trials, we concluded that the convergence happens at the 0.0001 value of the Learning Rate.
• Epochs: The model is trained on 10 epochs. (Previously we had 15 epochs, but we observed that it overfitted our model). We changed the epochs from 15 to 10 in the second build of the project.
Otherwise, the model is the same as the one in the first build of the project.

3. Bias

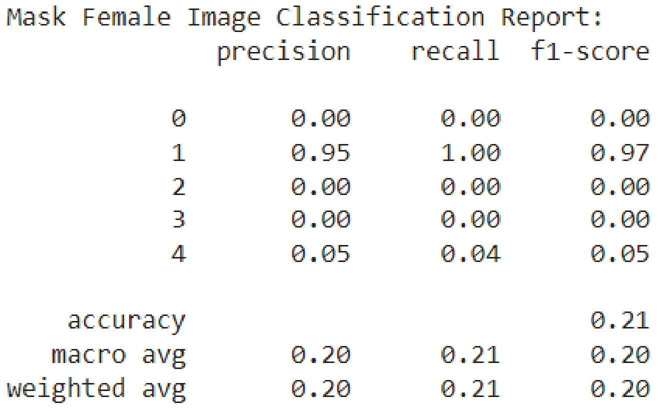
A bias is a phenomenon where the AI tends to produce outcomes that might be partialized or prejudiced towards some kind of data/images. This scenario usually occurs when the system encounters images/data that it has never seen before. Bias is that data that does not result in an accurate prediction in a situation where that data is not included in the training part.

Bias existing in datasets results in poorly trained data because the bias will lead to a skewed model that would not predict correctly for something else. Therefore, it is vital to check if there exists a bias in our model. For the same, we have considered two bases for bias to be present - Age and Gender. For the Age bifurcation, we have considered three age groups - child, young, and old. We have taken a total of approximately 375 images for testing the above-mentioned three age groups across our five classes. Similarly, based on the gender of the people wearing the masks, we have taken a total of 200 images across all the classes.

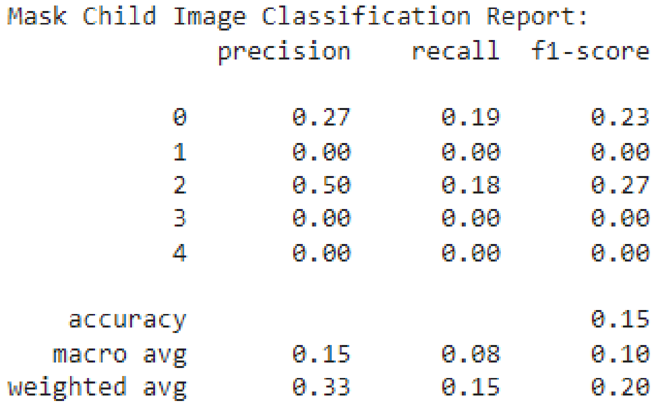
Classification report on the bias - Part 1:



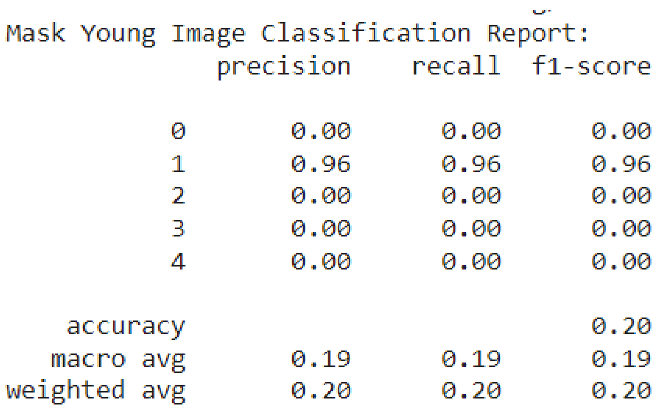
```
Mask Male Image Classification Report:
              precision    recall  f1-score

           0       0.00      0.00      0.00
           1       0.80      1.00      0.89
           2       0.00      0.00      0.00
           3       0.05      0.05      0.05
           4       0.00      0.00      0.00

    accuracy                           0.17
   macro avg       0.17      0.21      0.19
weighted avg       0.14      0.17      0.15
```



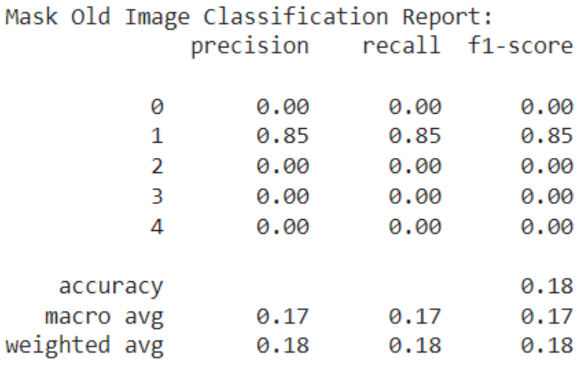
```
Mask Female Image Classification Report:
              precision    recall  f1-score

           0       0.00      0.00      0.00
           1       0.95      1.00      0.97
           2       0.00      0.00      0.00
           3       0.00      0.00      0.00
           4       0.05      0.04      0.05

    accuracy                           0.21
   macro avg       0.20      0.21      0.20
weighted avg       0.20      0.21      0.20
```



```
Mask Child Image Classification Report:
              precision    recall  f1-score

           0       0.27      0.19      0.23
           1       0.00      0.00      0.00
           2       0.50      0.18      0.27
           3       0.00      0.00      0.00
           4       0.00      0.00      0.00

    accuracy                           0.15
   macro avg       0.15      0.08      0.10
weighted avg       0.33      0.15      0.20
```

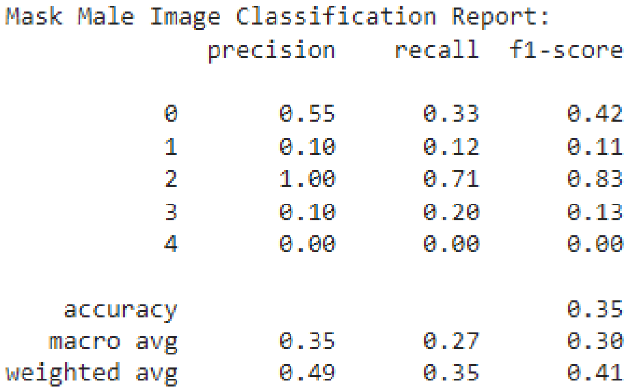


```
Mask Young Image Classification Report:
              precision    recall  f1-score

           0       0.00      0.00      0.00
           1       0.96      0.96      0.96
           2       0.00      0.00      0.00
           3       0.00      0.00      0.00
           4       0.00      0.00      0.00

    accuracy                           0.20
   macro avg       0.19      0.19      0.19
weighted avg       0.20      0.20      0.20
```

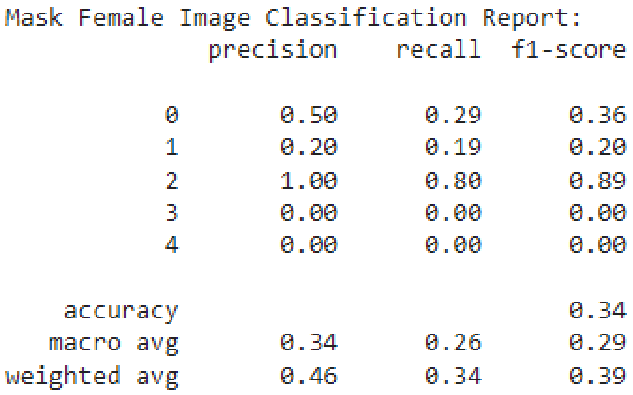


```
Mask Old Image Classification Report:
              precision    recall  f1-score

           0       0.00      0.00      0.00
           1       0.85      0.85      0.85
           2       0.00      0.00      0.00
           3       0.00      0.00      0.00
           4       0.00      0.00      0.00

    accuracy                           0.18
   macro avg       0.17      0.17      0.17
weighted avg       0.18      0.18      0.18
```

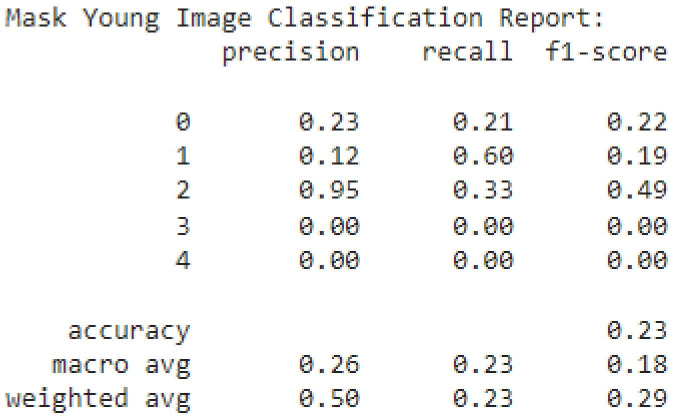
Classification report on the bias - Part 2:



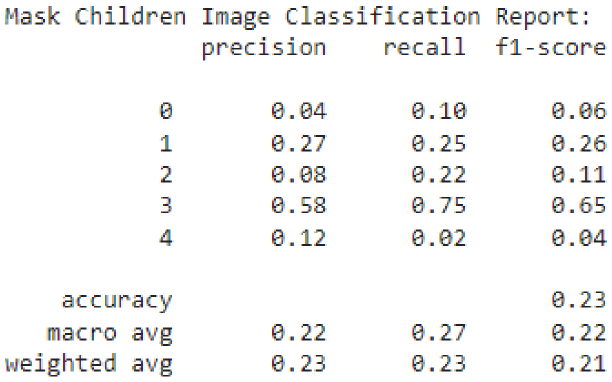
```
Mask Male Image Classification Report:
              precision    recall  f1-score

           0       0.55      0.33      0.42
           1       0.10      0.12      0.11
           2       1.00      0.71      0.83
           3       0.10      0.20      0.13
           4       0.00      0.00      0.00

    accuracy                           0.35
   macro avg       0.35      0.27      0.30
weighted avg       0.49      0.35      0.41
```



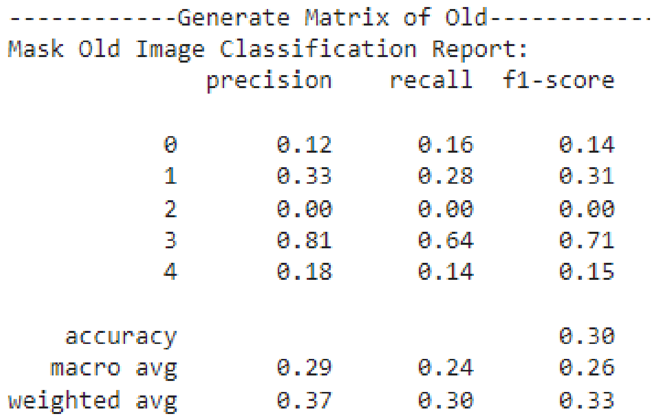
```
Mask Female Image Classification Report:
              precision    recall  f1-score

           0       0.50      0.29      0.36
           1       0.20      0.19      0.20
           2       1.00      0.80      0.89
           3       0.00      0.00      0.00
           4       0.00      0.00      0.00

    accuracy                           0.34
   macro avg       0.34      0.26      0.29
weighted avg       0.46      0.34      0.39
```



```
Mask Young Image Classification Report:
              precision    recall  f1-score

           0       0.23      0.21      0.22
           1       0.12      0.60      0.19
           2       0.95      0.33      0.49
           3       0.00      0.00      0.00
           4       0.00      0.00      0.00

    accuracy                           0.23
   macro avg       0.26      0.23      0.18
weighted avg       0.50      0.23      0.29
```



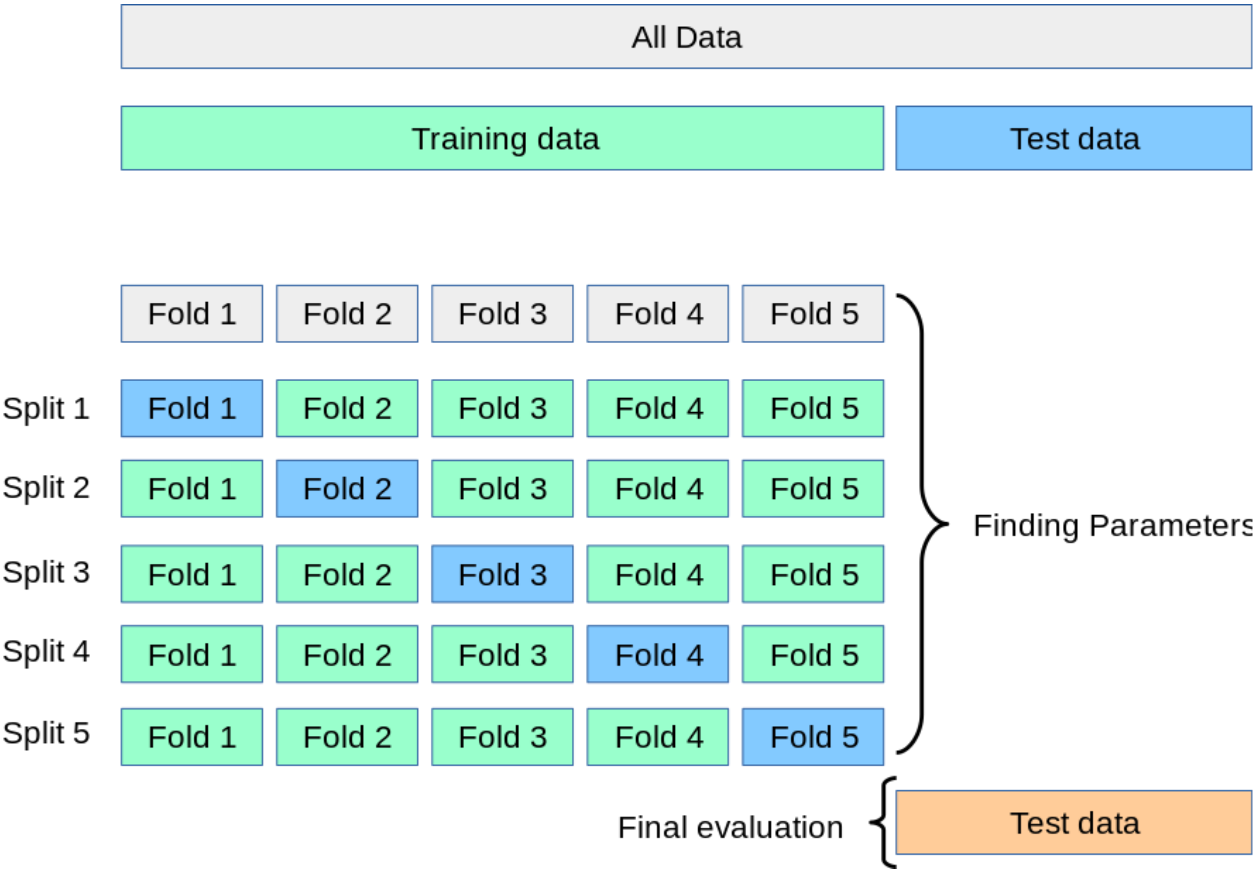
```
Mask Children Image Classification Report:
              precision    recall  f1-score

           0       0.04      0.10      0.06
           1       0.27      0.25      0.26
           2       0.08      0.22      0.11
           3       0.58      0.75      0.65
           4       0.12      0.02      0.04

    accuracy                           0.23
   macro avg       0.22      0.27      0.22
weighted avg       0.23      0.23      0.21
```



```
------------Generate Matrix of Old------------
Mask Old Image Classification Report:
              precision    recall  f1-score

           0       0.12      0.16      0.14
           1       0.33      0.28      0.31
           2       0.00      0.00      0.00
           3       0.81      0.64      0.71
           4       0.18      0.14      0.15

    accuracy                           0.30
   macro avg       0.29      0.24      0.26
weighted avg       0.37      0.30      0.33
```

4. K-fold Cross-Validation

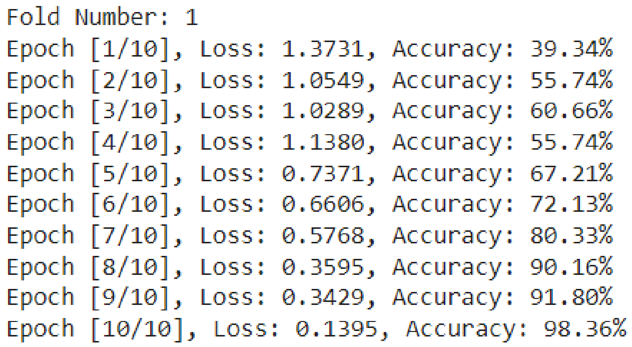
K-fold Cross-validation is **a resampling procedure used to evaluate machine learning models on a limited data sample**. The procedure has a single parameter called k which refers to the number of groups that a given data sample is to be split into. For our project, we have a total of 2982 images. These images are shuffled and split into 2 data sets one for training (2087 images) and one for testing (895 images)



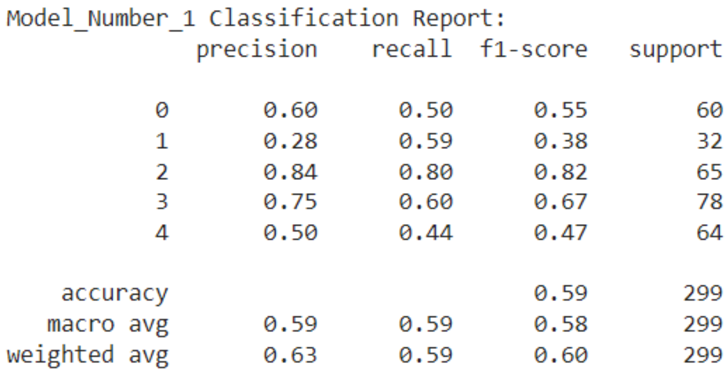
-

With the biased dataset of Part 1, we implemented K- Fold cross validation and reached the end of the last fold with an accuracy of 55%. The dataset in part 1 is biased as illustrated in the **Bias** section of the report. We have eliminated the biased part of the dataset in the build 2 of the project by balancing out the kind of images present in each class. These kinds are males, females, children, young, and the old. For the second build of the project, after removing the bias as much as possible, we implemented K- Fold for partitioning the data again which resulted in 65% accuracy at the end of the 10th fold. After the folds have been run, we have generated classification reports for different bias classes and observed that the bias percentage decreases after balancing out the dataset. The outputs of K-Fold Cross-Validation are shown below for the Part 1 and Part 2.

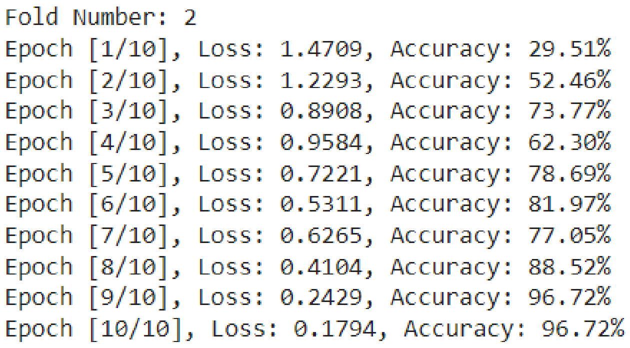
Applying K- Fold on Build 2- After Eliminating Bias



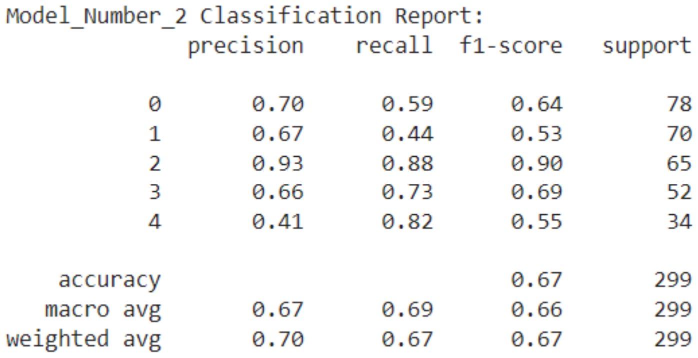
```
Fold Number: 1
Epoch [1/10], Loss: 1.3731, Accuracy: 39.34%
Epoch [2/10], Loss: 1.0549, Accuracy: 55.74%
Epoch [3/10], Loss: 1.0289, Accuracy: 60.66%
Epoch [4/10], Loss: 1.1380, Accuracy: 55.74%
Epoch [5/10], Loss: 0.7371, Accuracy: 67.21%
Epoch [6/10], Loss: 0.6606, Accuracy: 72.13%
Epoch [7/10], Loss: 0.5768, Accuracy: 80.33%
Epoch [8/10], Loss: 0.3595, Accuracy: 90.16%
Epoch [9/10], Loss: 0.3429, Accuracy: 91.80%
Epoch [10/10], Loss: 0.1395, Accuracy: 98.36%
```



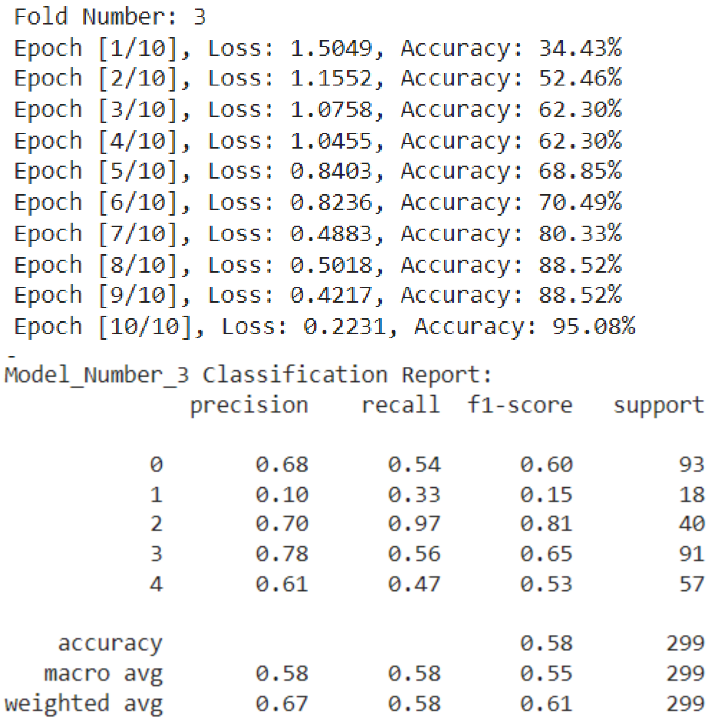
```
Model_Number_1 Classification Report:
              precision    recall  f1-score   support

           0       0.60      0.50      0.55        60
           1       0.28      0.59      0.38        32
           2       0.84      0.80      0.82        65
           3       0.75      0.60      0.67        78
           4       0.50      0.44      0.47        64

    accuracy                           0.59       299
   macro avg       0.59      0.59      0.58       299
weighted avg       0.63      0.59      0.60       299
```



```
Fold Number: 2
Epoch [1/10], Loss: 1.4709, Accuracy: 29.51%
Epoch [2/10], Loss: 1.2293, Accuracy: 52.46%
Epoch [3/10], Loss: 0.8908, Accuracy: 73.77%
Epoch [4/10], Loss: 0.9584, Accuracy: 62.30%
Epoch [5/10], Loss: 0.7221, Accuracy: 78.69%
Epoch [6/10], Loss: 0.5311, Accuracy: 81.97%
Epoch [7/10], Loss: 0.6265, Accuracy: 77.05%
Epoch [8/10], Loss: 0.4104, Accuracy: 88.52%
Epoch [9/10], Loss: 0.2429, Accuracy: 96.72%
Epoch [10/10], Loss: 0.1794, Accuracy: 96.72%
```

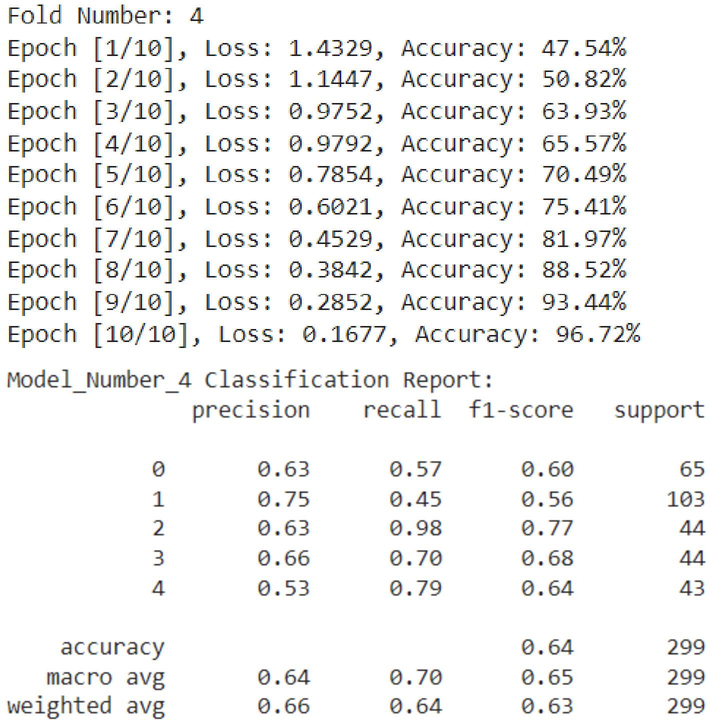


```
Model_Number_2 Classification Report:
              precision    recall  f1-score   support

           0       0.70      0.59      0.64        78
           1       0.67      0.44      0.53        70
           2       0.93      0.88      0.90        65
           3       0.66      0.73      0.69        52
           4       0.41      0.82      0.55        34

    accuracy                           0.67       299
   macro avg       0.67      0.69      0.66       299
weighted avg       0.70      0.67      0.67       299
```



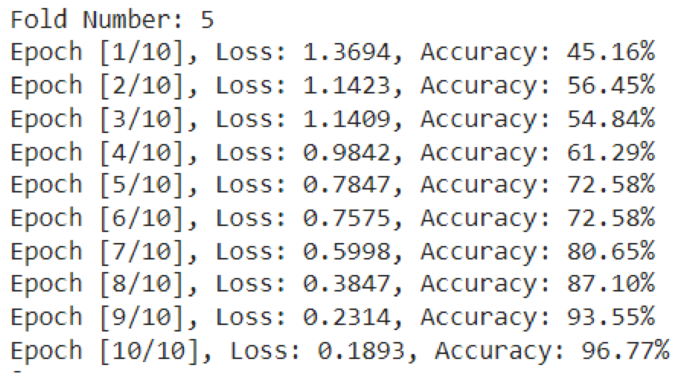
```
Fold Number: 3
Epoch [1/10], Loss: 1.5049, Accuracy: 34.43%
Epoch [2/10], Loss: 1.1552, Accuracy: 52.46%
Epoch [3/10], Loss: 1.0758, Accuracy: 62.30%
Epoch [4/10], Loss: 1.0455, Accuracy: 62.30%
Epoch [5/10], Loss: 0.8403, Accuracy: 68.85%
Epoch [6/10], Loss: 0.8236, Accuracy: 70.49%
Epoch [7/10], Loss: 0.4883, Accuracy: 80.33%
Epoch [8/10], Loss: 0.5018, Accuracy: 88.52%
Epoch [9/10], Loss: 0.4217, Accuracy: 88.52%
Epoch [10/10], Loss: 0.2231, Accuracy: 95.08%
```

```
Model_Number_3 Classification Report:
              precision    recall  f1-score   support

           0       0.68      0.54      0.60        93
           1       0.10      0.33      0.15        18
           2       0.70      0.97      0.81        40
           3       0.78      0.56      0.65        91
           4       0.61      0.47      0.53        57

    accuracy                           0.58       299
   macro avg       0.58      0.58      0.55       299
weighted avg       0.67      0.58      0.61       299
```

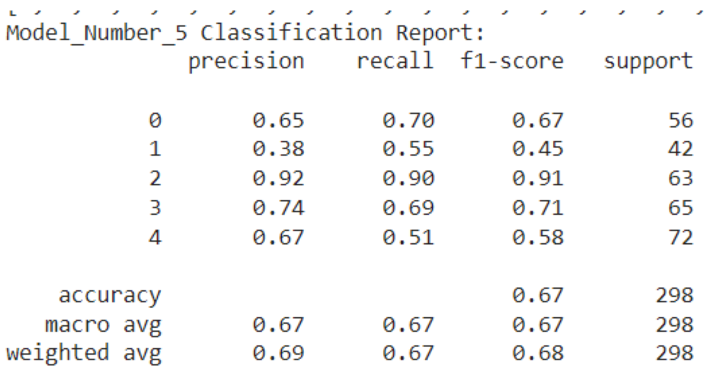


```
Fold Number: 4
Epoch [1/10], Loss: 1.4329, Accuracy: 47.54%
Epoch [2/10], Loss: 1.1447, Accuracy: 50.82%
Epoch [3/10], Loss: 0.9752, Accuracy: 63.93%
Epoch [4/10], Loss: 0.9792, Accuracy: 65.57%
Epoch [5/10], Loss: 0.7854, Accuracy: 70.49%
Epoch [6/10], Loss: 0.6021, Accuracy: 75.41%
Epoch [7/10], Loss: 0.4529, Accuracy: 81.97%
Epoch [8/10], Loss: 0.3842, Accuracy: 88.52%
Epoch [9/10], Loss: 0.2852, Accuracy: 93.44%
Epoch [10/10], Loss: 0.1677, Accuracy: 96.72%
```

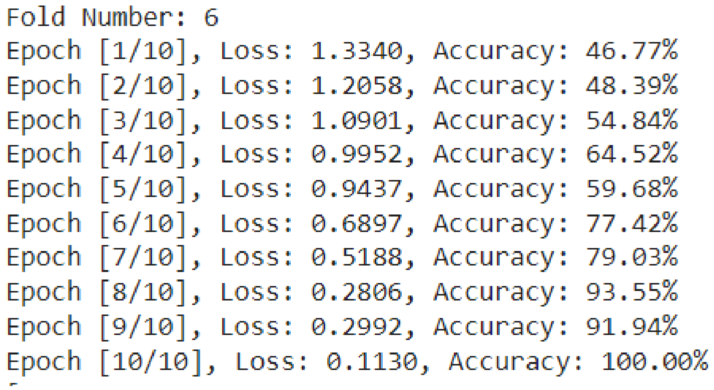
```
Model_Number_4 Classification Report:
              precision    recall  f1-score   support

           0       0.63      0.57      0.60        65
           1       0.75      0.45      0.56       103
           2       0.63      0.98      0.77        44
           3       0.66      0.70      0.68        44
           4       0.53      0.79      0.64        43

    accuracy                           0.64       299
   macro avg       0.64      0.70      0.65       299
weighted avg       0.66      0.64      0.63       299
```



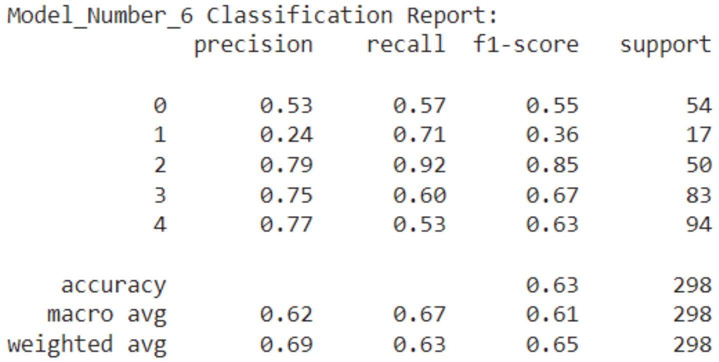
```
Fold Number: 5
Epoch [1/10], Loss: 1.3694, Accuracy: 45.16%
Epoch [2/10], Loss: 1.1423, Accuracy: 56.45%
Epoch [3/10], Loss: 1.1409, Accuracy: 54.84%
Epoch [4/10], Loss: 0.9842, Accuracy: 61.29%
Epoch [5/10], Loss: 0.7847, Accuracy: 72.58%
Epoch [6/10], Loss: 0.7575, Accuracy: 72.58%
Epoch [7/10], Loss: 0.5998, Accuracy: 80.65%
Epoch [8/10], Loss: 0.3847, Accuracy: 87.10%
Epoch [9/10], Loss: 0.2314, Accuracy: 93.55%
Epoch [10/10], Loss: 0.1893, Accuracy: 96.77%
```



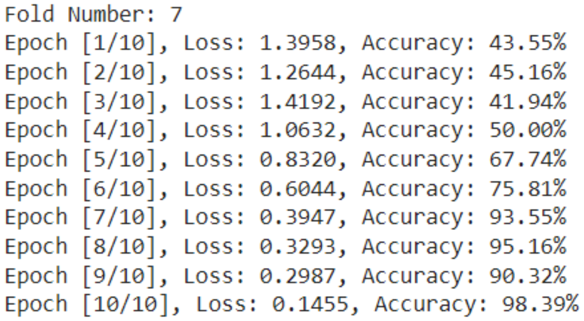
```
Model_Number_5 Classification Report:
              precision    recall  f1-score   support

           0       0.65      0.70      0.67        56
           1       0.38      0.55      0.45        42
           2       0.92      0.90      0.91        63
           3       0.74      0.69      0.71        65
           4       0.67      0.51      0.58        72

    accuracy                           0.67       298
   macro avg       0.67      0.67      0.67       298
weighted avg       0.69      0.67      0.68       298
```



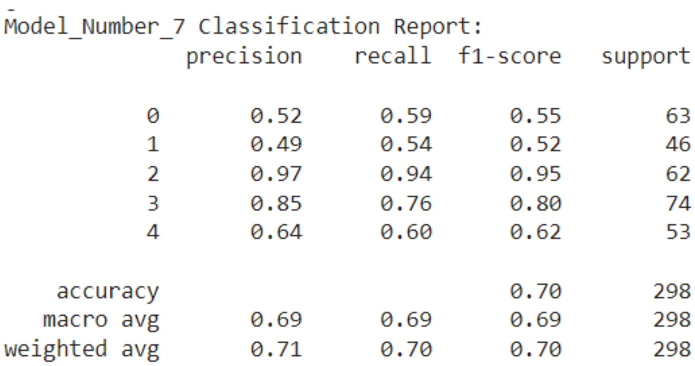
```
Fold Number: 6
Epoch [1/10], Loss: 1.3340, Accuracy: 46.77%
Epoch [2/10], Loss: 1.2058, Accuracy: 48.39%
Epoch [3/10], Loss: 1.0901, Accuracy: 54.84%
Epoch [4/10], Loss: 0.9952, Accuracy: 64.52%
Epoch [5/10], Loss: 0.9437, Accuracy: 59.68%
Epoch [6/10], Loss: 0.6897, Accuracy: 77.42%
Epoch [7/10], Loss: 0.5188, Accuracy: 79.03%
Epoch [8/10], Loss: 0.2806, Accuracy: 93.55%
Epoch [9/10], Loss: 0.2992, Accuracy: 91.94%
Epoch [10/10], Loss: 0.1130, Accuracy: 100.00%
```



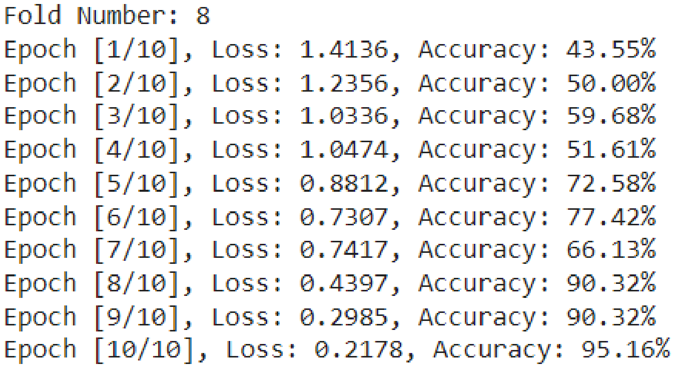
```
Model_Number_6 Classification Report:
              precision    recall  f1-score   support

           0       0.53      0.57      0.55        54
           1       0.24      0.71      0.36        17
           2       0.79      0.92      0.85        50
           3       0.75      0.60      0.67        83
           4       0.77      0.53      0.63        94

    accuracy                           0.63       298
   macro avg       0.62      0.67      0.61       298
weighted avg       0.69      0.63      0.65       298
```



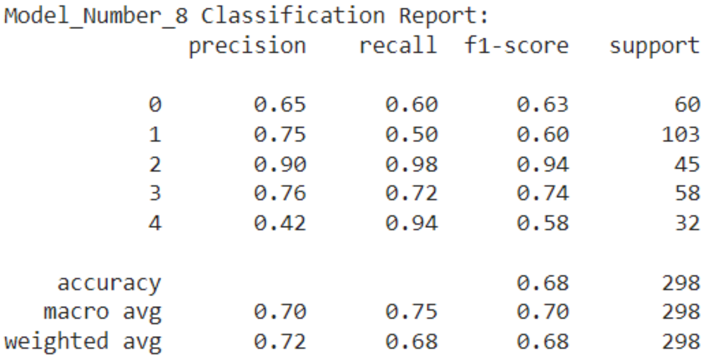
```
Fold Number: 7
Epoch [1/10], Loss: 1.3958, Accuracy: 43.55%
Epoch [2/10], Loss: 1.2644, Accuracy: 45.16%
Epoch [3/10], Loss: 1.4192, Accuracy: 41.94%
Epoch [4/10], Loss: 1.0632, Accuracy: 50.00%
Epoch [5/10], Loss: 0.8320, Accuracy: 67.74%
Epoch [6/10], Loss: 0.6044, Accuracy: 75.81%
Epoch [7/10], Loss: 0.3947, Accuracy: 93.55%
Epoch [8/10], Loss: 0.3293, Accuracy: 95.16%
Epoch [9/10], Loss: 0.2987, Accuracy: 90.32%
Epoch [10/10], Loss: 0.1455, Accuracy: 98.39%
```



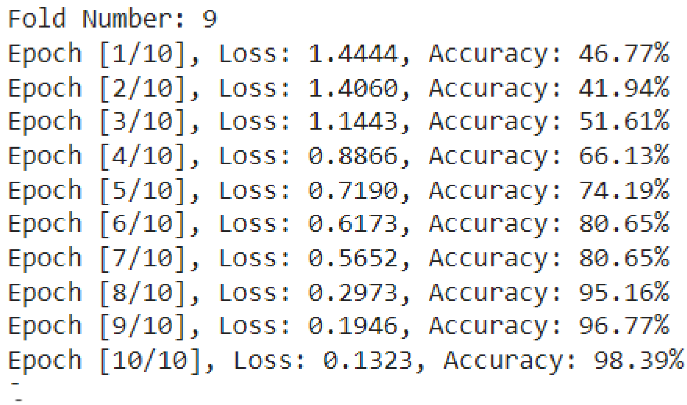
```
Model_Number_7 Classification Report:
              precision    recall  f1-score   support

           0       0.52      0.59      0.55        63
           1       0.49      0.54      0.52        46
           2       0.97      0.94      0.95        62
           3       0.85      0.76      0.80        74
           4       0.64      0.60      0.62        53

    accuracy                           0.70       298
   macro avg       0.69      0.69      0.69       298
weighted avg       0.71      0.70      0.70       298
```



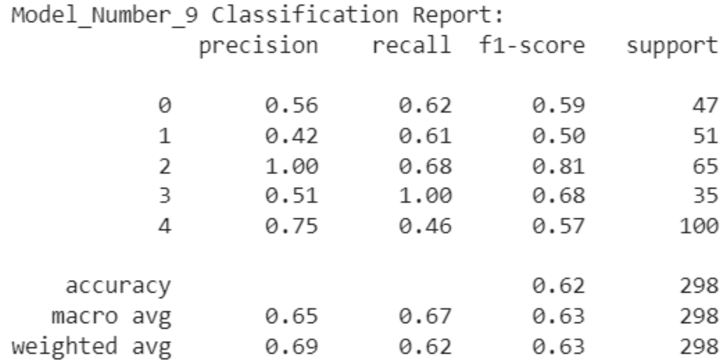
```
Fold Number: 8
Epoch [1/10], Loss: 1.4136, Accuracy: 43.55%
Epoch [2/10], Loss: 1.2356, Accuracy: 50.00%
Epoch [3/10], Loss: 1.0336, Accuracy: 59.68%
Epoch [4/10], Loss: 1.0474, Accuracy: 51.61%
Epoch [5/10], Loss: 0.8812, Accuracy: 72.58%
Epoch [6/10], Loss: 0.7307, Accuracy: 77.42%
Epoch [7/10], Loss: 0.7417, Accuracy: 66.13%
Epoch [8/10], Loss: 0.4397, Accuracy: 90.32%
Epoch [9/10], Loss: 0.2985, Accuracy: 90.32%
Epoch [10/10], Loss: 0.2178, Accuracy: 95.16%
```



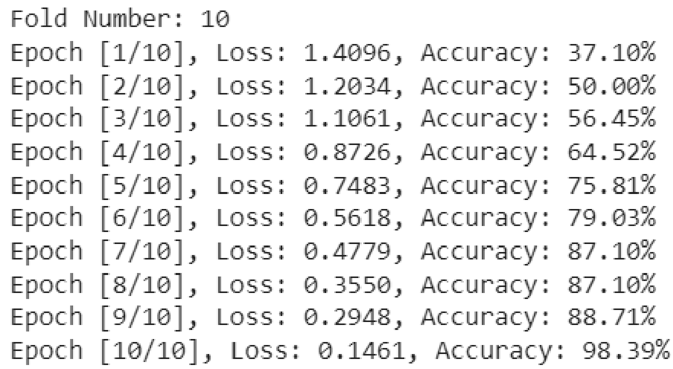
```
Model_Number_8 Classification Report:
              precision    recall  f1-score   support

           0       0.65      0.60      0.63        60
           1       0.75      0.50      0.60       103
           2       0.90      0.98      0.94        45
           3       0.76      0.72      0.74        58
           4       0.42      0.94      0.58        32

    accuracy                           0.68       298
   macro avg       0.70      0.75      0.70       298
weighted avg       0.72      0.68      0.68       298
```



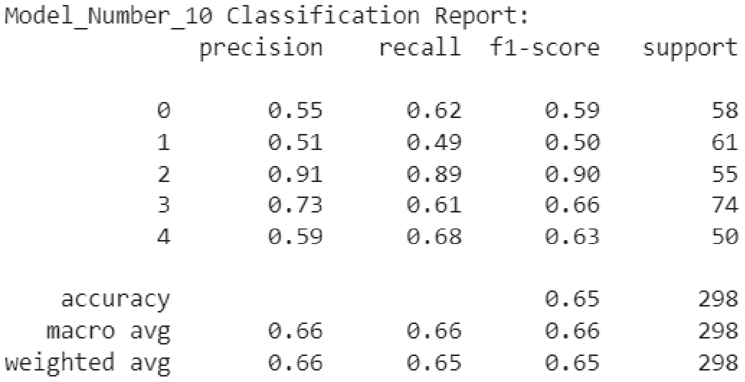
```
Fold Number: 9
Epoch [1/10], Loss: 1.4444, Accuracy: 46.77%
Epoch [2/10], Loss: 1.4060, Accuracy: 41.94%
Epoch [3/10], Loss: 1.1443, Accuracy: 51.61%
Epoch [4/10], Loss: 0.8866, Accuracy: 66.13%
Epoch [5/10], Loss: 0.7190, Accuracy: 74.19%
Epoch [6/10], Loss: 0.6173, Accuracy: 80.65%
Epoch [7/10], Loss: 0.5652, Accuracy: 80.65%
Epoch [8/10], Loss: 0.2973, Accuracy: 95.16%
Epoch [9/10], Loss: 0.1946, Accuracy: 96.77%
Epoch [10/10], Loss: 0.1323, Accuracy: 98.39%
-
```



```
Model_Number_9 Classification Report:
              precision    recall  f1-score   support

           0       0.56      0.62      0.59        47
           1       0.42      0.61      0.50        51
           2       1.00      0.68      0.81        65
           3       0.51      1.00      0.68        35
           4       0.75      0.46      0.57       100

    accuracy                           0.62       298
   macro avg       0.65      0.67      0.63       298
weighted avg       0.69      0.62      0.63       298
```

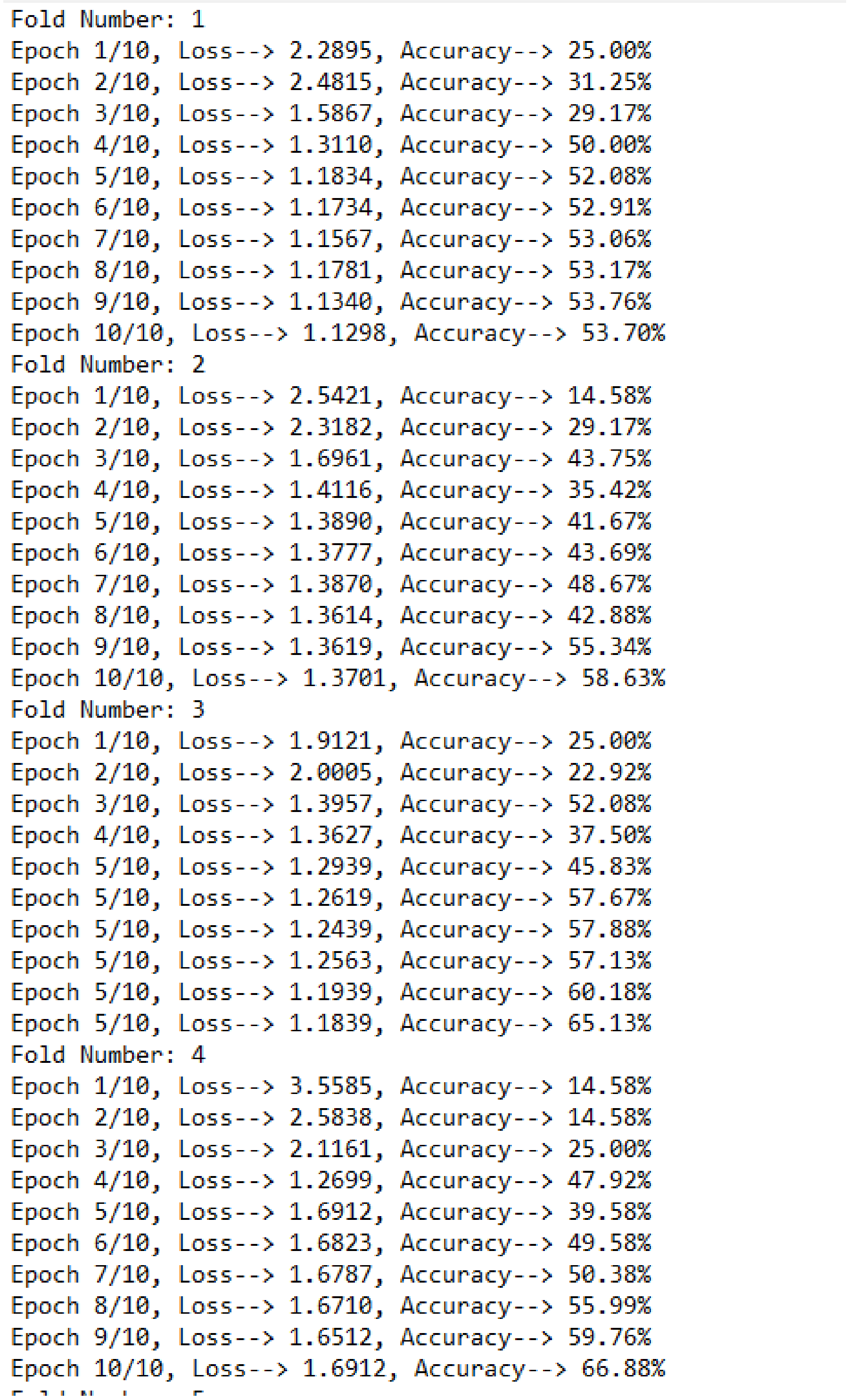


```
Fold Number: 10
Epoch [1/10], Loss: 1.4096, Accuracy: 37.10%
Epoch [2/10], Loss: 1.2034, Accuracy: 50.00%
Epoch [3/10], Loss: 1.1061, Accuracy: 56.45%
Epoch [4/10], Loss: 0.8726, Accuracy: 64.52%
Epoch [5/10], Loss: 0.7483, Accuracy: 75.81%
Epoch [6/10], Loss: 0.5618, Accuracy: 79.03%
Epoch [7/10], Loss: 0.4779, Accuracy: 87.10%
Epoch [8/10], Loss: 0.3550, Accuracy: 87.10%
Epoch [9/10], Loss: 0.2948, Accuracy: 88.71%
Epoch [10/10], Loss: 0.1461, Accuracy: 98.39%
```

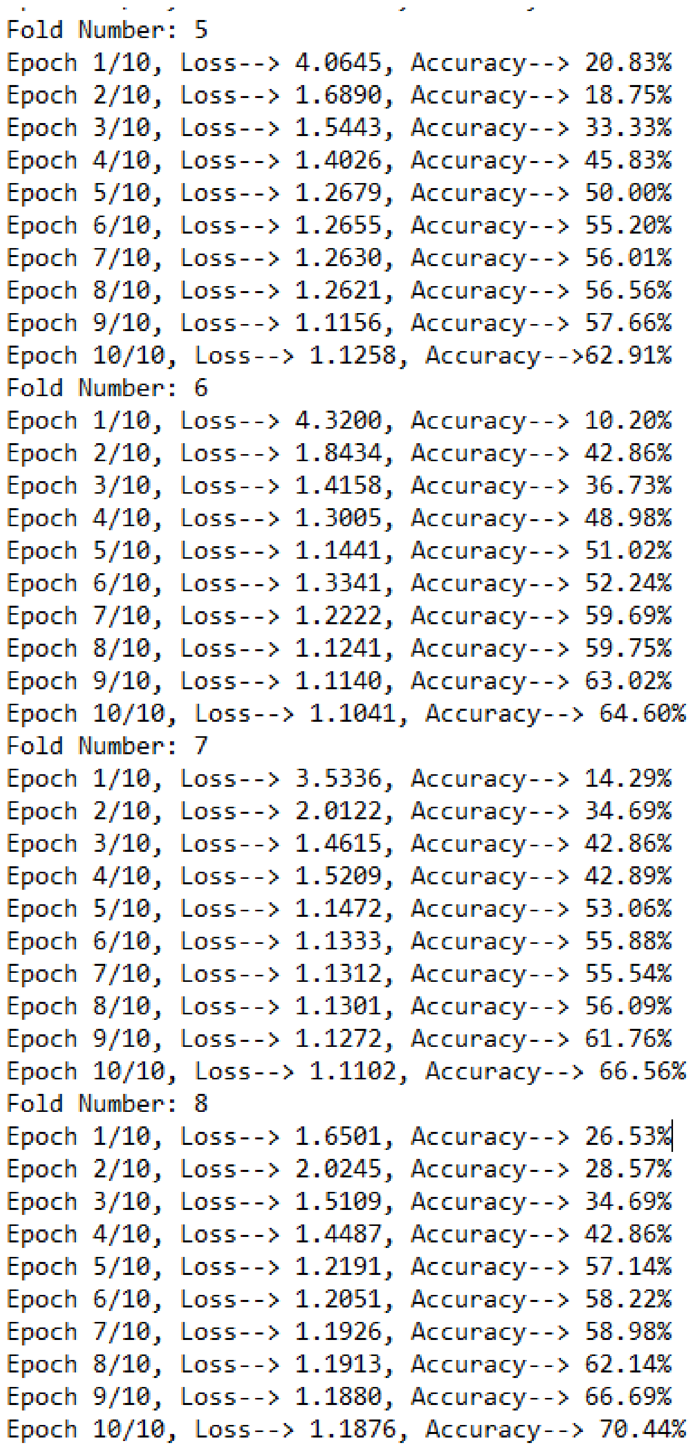


```
Model_Number_10 Classification Report:
              precision    recall  f1-score   support

           0       0.55      0.62      0.59        58
           1       0.51      0.49      0.50        61
           2       0.91      0.89      0.90        55
           3       0.73      0.61      0.66        74
           4       0.59      0.68      0.63        50

    accuracy                           0.65       298
   macro avg       0.66      0.66      0.66       298
weighted avg       0.66      0.65      0.65       298
```

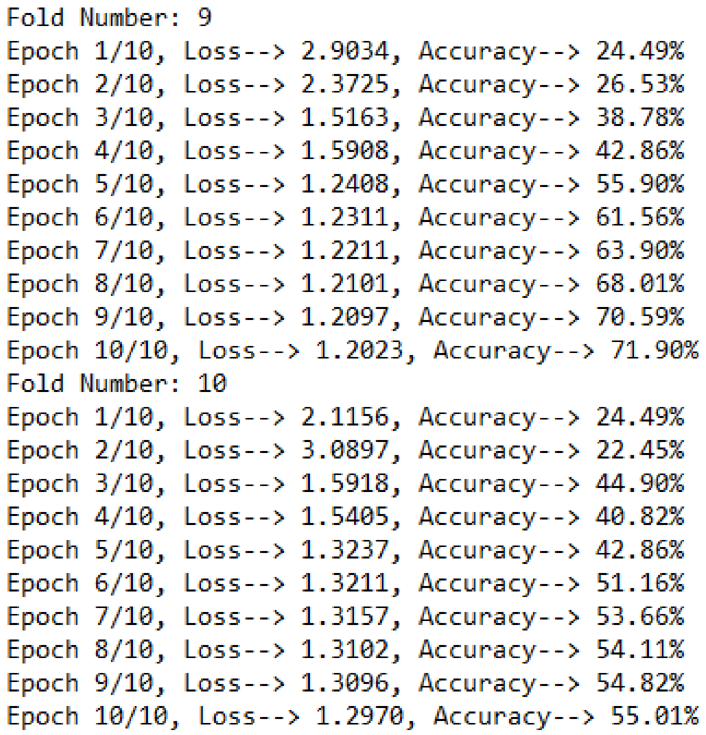
Applying K- Fold on Build 1- Before Eliminating Bias



```
Fold Number: 1
Epoch 1/10, Loss--> 2.2895, Accuracy--> 25.00%
Epoch 2/10, Loss--> 2.4815, Accuracy--> 31.25%
Epoch 3/10, Loss--> 1.5867, Accuracy--> 29.17%
Epoch 4/10, Loss--> 1.3110, Accuracy--> 50.00%
Epoch 5/10, Loss--> 1.1834, Accuracy--> 52.08%
Epoch 6/10, Loss--> 1.1734, Accuracy--> 52.91%
Epoch 7/10, Loss--> 1.1567, Accuracy--> 53.06%
Epoch 8/10, Loss--> 1.1781, Accuracy--> 53.17%
Epoch 9/10, Loss--> 1.1340, Accuracy--> 53.76%
Epoch 10/10, Loss--> 1.1298, Accuracy--> 53.70%
Fold Number: 2
Epoch 1/10, Loss--> 2.5421, Accuracy--> 14.58%
Epoch 2/10, Loss--> 2.3182, Accuracy--> 29.17%
Epoch 3/10, Loss--> 1.6961, Accuracy--> 43.75%
Epoch 4/10, Loss--> 1.4116, Accuracy--> 35.42%
Epoch 5/10, Loss--> 1.3890, Accuracy--> 41.67%
Epoch 6/10, Loss--> 1.3777, Accuracy--> 43.69%
Epoch 7/10, Loss--> 1.3870, Accuracy--> 48.67%
Epoch 8/10, Loss--> 1.3614, Accuracy--> 42.88%
Epoch 9/10, Loss--> 1.3619, Accuracy--> 55.34%
Epoch 10/10, Loss--> 1.3701, Accuracy--> 58.63%
Fold Number: 3
Epoch 1/10, Loss--> 1.9121, Accuracy--> 25.00%
Epoch 2/10, Loss--> 2.0005, Accuracy--> 22.92%
Epoch 3/10, Loss--> 1.3957, Accuracy--> 52.08%
Epoch 4/10, Loss--> 1.3627, Accuracy--> 37.50%
Epoch 5/10, Loss--> 1.2939, Accuracy--> 45.83%
Epoch 5/10, Loss--> 1.2619, Accuracy--> 57.67%
Epoch 5/10, Loss--> 1.2439, Accuracy--> 57.88%
Epoch 5/10, Loss--> 1.2563, Accuracy--> 57.13%
Epoch 5/10, Loss--> 1.1939, Accuracy--> 60.18%
Epoch 5/10, Loss--> 1.1839, Accuracy--> 65.13%
Fold Number: 4
Epoch 1/10, Loss--> 3.5585, Accuracy--> 14.58%
Epoch 2/10, Loss--> 2.5838, Accuracy--> 14.58%
Epoch 3/10, Loss--> 2.1161, Accuracy--> 25.00%
Epoch 4/10, Loss--> 1.2699, Accuracy--> 47.92%
Epoch 5/10, Loss--> 1.6912, Accuracy--> 39.58%
Epoch 6/10, Loss--> 1.6823, Accuracy--> 49.58%
Epoch 7/10, Loss--> 1.6787, Accuracy--> 50.38%
Epoch 8/10, Loss--> 1.6710, Accuracy--> 55.99%
Epoch 9/10, Loss--> 1.6512, Accuracy--> 59.76%
Epoch 10/10, Loss--> 1.6912, Accuracy--> 66.88%
```



```
Fold Number: 5
Epoch 1/10, Loss--> 4.0645, Accuracy--> 20.83%
Epoch 2/10, Loss--> 1.6890, Accuracy--> 18.75%
Epoch 3/10, Loss--> 1.5443, Accuracy--> 33.33%
Epoch 4/10, Loss--> 1.4026, Accuracy--> 45.83%
Epoch 5/10, Loss--> 1.2679, Accuracy--> 50.00%
Epoch 6/10, Loss--> 1.2655, Accuracy--> 55.20%
Epoch 7/10, Loss--> 1.2630, Accuracy--> 56.01%
Epoch 8/10, Loss--> 1.2621, Accuracy--> 56.56%
Epoch 9/10, Loss--> 1.1156, Accuracy--> 57.66%
Epoch 10/10, Loss--> 1.1258, Accuracy-->62.91%
Fold Number: 6
Epoch 1/10, Loss--> 4.3200, Accuracy--> 10.20%
Epoch 2/10, Loss--> 1.8434, Accuracy--> 42.86%
Epoch 3/10, Loss--> 1.4158, Accuracy--> 36.73%
Epoch 4/10, Loss--> 1.3005, Accuracy--> 48.98%
Epoch 5/10, Loss--> 1.1441, Accuracy--> 51.02%
Epoch 6/10, Loss--> 1.3341, Accuracy--> 52.24%
Epoch 7/10, Loss--> 1.2222, Accuracy--> 59.69%
Epoch 8/10, Loss--> 1.1241, Accuracy--> 59.75%
Epoch 9/10, Loss--> 1.1140, Accuracy--> 63.02%
Epoch 10/10, Loss--> 1.1041, Accuracy--> 64.60%
Fold Number: 7
Epoch 1/10, Loss--> 3.5336, Accuracy--> 14.29%
Epoch 2/10, Loss--> 2.0122, Accuracy--> 34.69%
Epoch 3/10, Loss--> 1.4615, Accuracy--> 42.86%
Epoch 4/10, Loss--> 1.5209, Accuracy--> 42.89%
Epoch 5/10, Loss--> 1.1472, Accuracy--> 53.06%
Epoch 6/10, Loss--> 1.1333, Accuracy--> 55.88%
Epoch 7/10, Loss--> 1.1312, Accuracy--> 55.54%
Epoch 8/10, Loss--> 1.1301, Accuracy--> 56.09%
Epoch 9/10, Loss--> 1.1272, Accuracy--> 61.76%
Epoch 10/10, Loss--> 1.1102, Accuracy--> 66.56%
Fold Number: 8
Epoch 1/10, Loss--> 1.6501, Accuracy--> 26.53%
Epoch 2/10, Loss--> 2.0245, Accuracy--> 28.57%
Epoch 3/10, Loss--> 1.5109, Accuracy--> 34.69%
Epoch 4/10, Loss--> 1.4487, Accuracy--> 42.86%
Epoch 5/10, Loss--> 1.2191, Accuracy--> 57.14%
Epoch 6/10, Loss--> 1.2051, Accuracy--> 58.22%
Epoch 7/10, Loss--> 1.1926, Accuracy--> 58.98%
Epoch 8/10, Loss--> 1.1913, Accuracy--> 62.14%
Epoch 9/10, Loss--> 1.1880, Accuracy--> 66.69%
Epoch 10/10, Loss--> 1.1876, Accuracy--> 70.44%
```



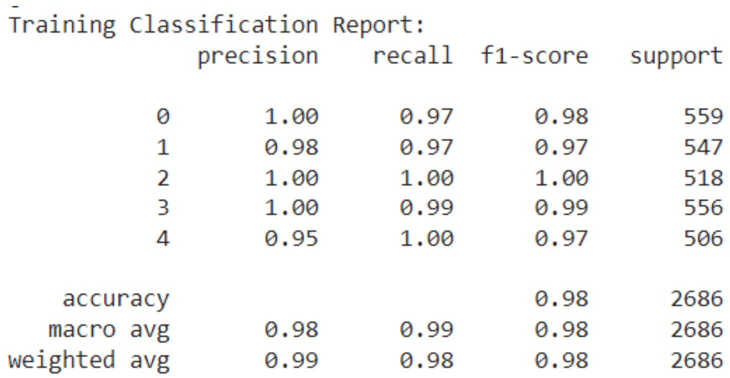
```
Fold Number: 9
Epoch 1/10, Loss--> 2.9034, Accuracy--> 24.49%
Epoch 2/10, Loss--> 2.3725, Accuracy--> 26.53%
Epoch 3/10, Loss--> 1.5163, Accuracy--> 38.78%
Epoch 4/10, Loss--> 1.5908, Accuracy--> 42.86%
Epoch 5/10, Loss--> 1.2408, Accuracy--> 55.90%
Epoch 6/10, Loss--> 1.2311, Accuracy--> 61.56%
Epoch 7/10, Loss--> 1.2211, Accuracy--> 63.90%
Epoch 8/10, Loss--> 1.2101, Accuracy--> 68.01%
Epoch 9/10, Loss--> 1.2097, Accuracy--> 70.59%
Epoch 10/10, Loss--> 1.2023, Accuracy--> 71.90%
Fold Number: 10
Epoch 1/10, Loss--> 2.1156, Accuracy--> 24.49%
Epoch 2/10, Loss--> 3.0897, Accuracy--> 22.45%
Epoch 3/10, Loss--> 1.5918, Accuracy--> 44.90%
Epoch 4/10, Loss--> 1.5405, Accuracy--> 40.82%
Epoch 5/10, Loss--> 1.3237, Accuracy--> 42.86%
Epoch 6/10, Loss--> 1.3211, Accuracy--> 51.16%
Epoch 7/10, Loss--> 1.3157, Accuracy--> 53.66%
Epoch 8/10, Loss--> 1.3102, Accuracy--> 54.11%
Epoch 9/10, Loss--> 1.3096, Accuracy--> 54.82%
Epoch 10/10, Loss--> 1.2970, Accuracy--> 55.01%
```

5. Evaluation: K- Fold Cross-Validation

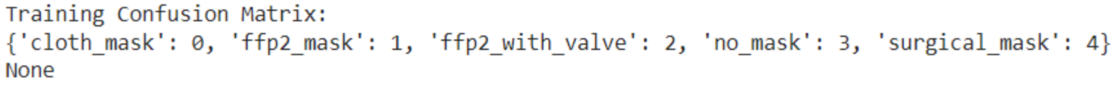
Evaluation is the way of determining how reliable our model is based on the predictions it has made on the training/testing data. This evaluation is done by giving the data input to the model and comparing the actual answers with the output the model gives us.

The performance of our code is inferred from what our model predicts and what the actual category of the image should be. Confusion metrics are given below for the training and testing parts of the project, to decide on how much the reliability of our model actually is. To determine the accuracy, we have used parameters like recall, precision, F1-score, and accuracy. Recall refers to the fraction of relevant items that an AI search returns out of the total number of relevant items in the original population. Precision is the number of true positives a model predicts. The average precision of our training is 0.96, the average recall is 0.97, the average F1- score is 0.96, and the average support is 1752.

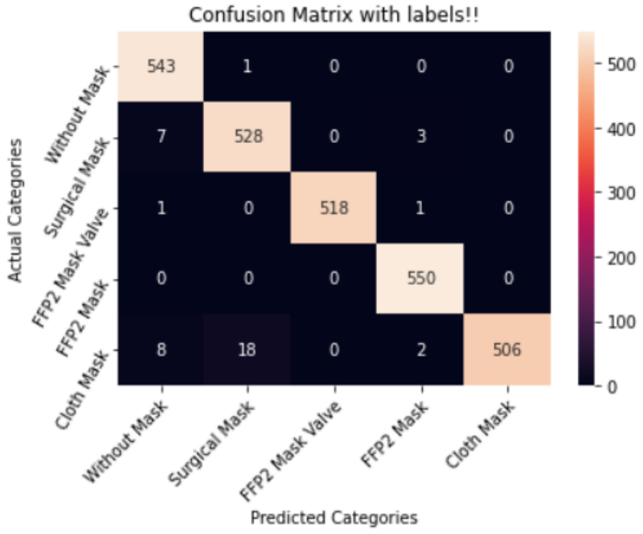
Part 1: Without K-Fold
Training Classification Report and Confusion Matrix:



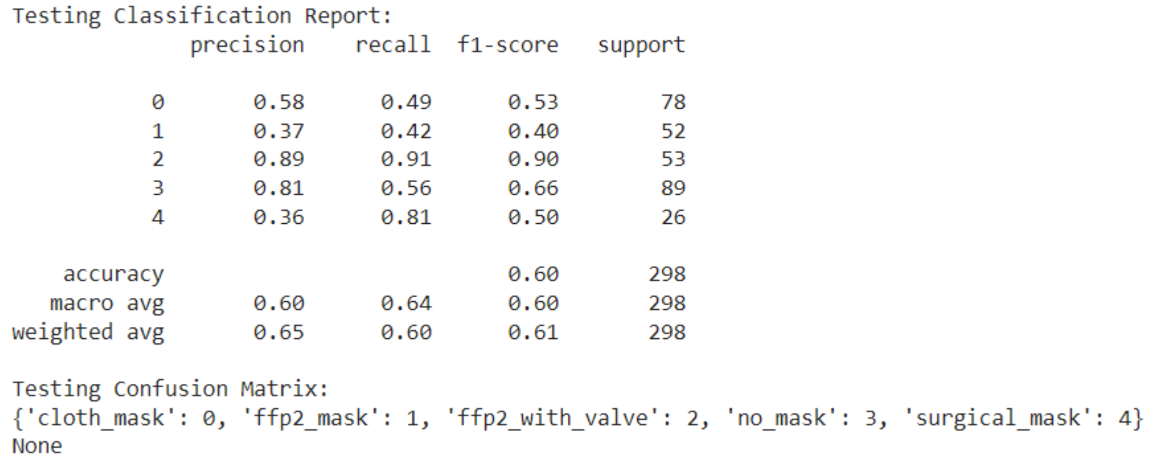
```
Training Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.97      0.98       559
           1       0.98      0.97      0.97       547
           2       1.00      1.00      1.00       518
           3       1.00      0.99      0.99       556
           4       0.95      1.00      0.97       506

    accuracy                           0.98      2686
   macro avg       0.98      0.99      0.98      2686
weighted avg       0.99      0.98      0.98      2686
```



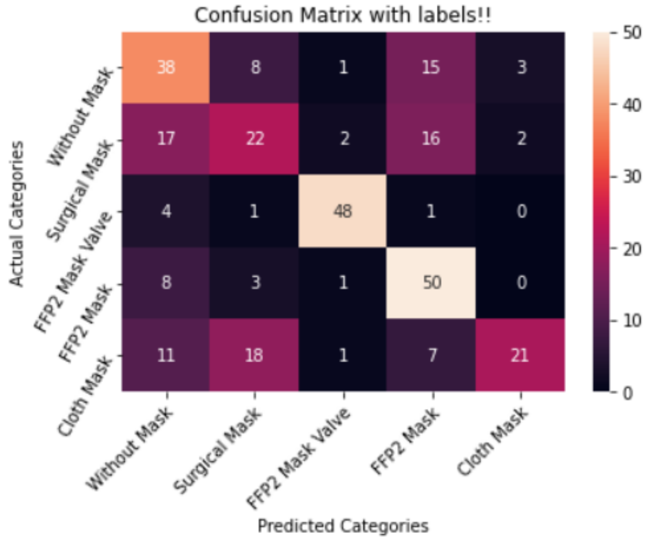
```
Training Confusion Matrix:
{'cloth_mask': 0, 'ffp2_mask': 1, 'ffp2_with_valve': 2, 'no_mask': 3, 'surgical_mask': 4}
None
```



Testing Classification Report and Confusion Matrix:

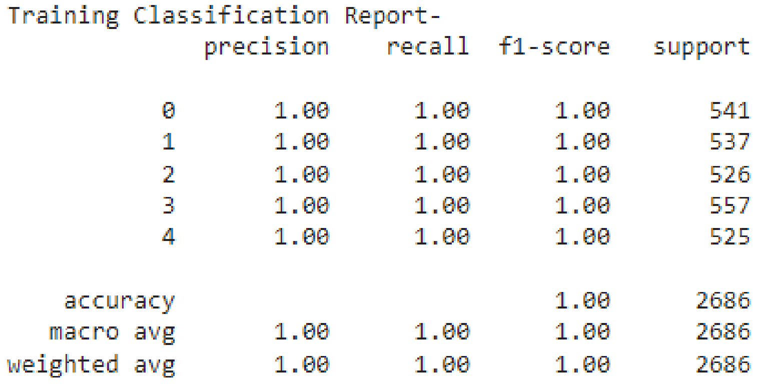


```
Testing Classification Report:
              precision    recall  f1-score   support

           0       0.58      0.49      0.53        78
           1       0.37      0.42      0.40        52
           2       0.89      0.91      0.90        53
           3       0.81      0.56      0.66        89
           4       0.36      0.81      0.50        26

    accuracy                           0.60       298
   macro avg       0.60      0.64      0.60       298
weighted avg       0.65      0.60      0.61       298

Testing Confusion Matrix:
{'cloth_mask': 0, 'ffp2_mask': 1, 'ffp2_with_valve': 2, 'no_mask': 3, 'surgical_mask': 4}
None
```

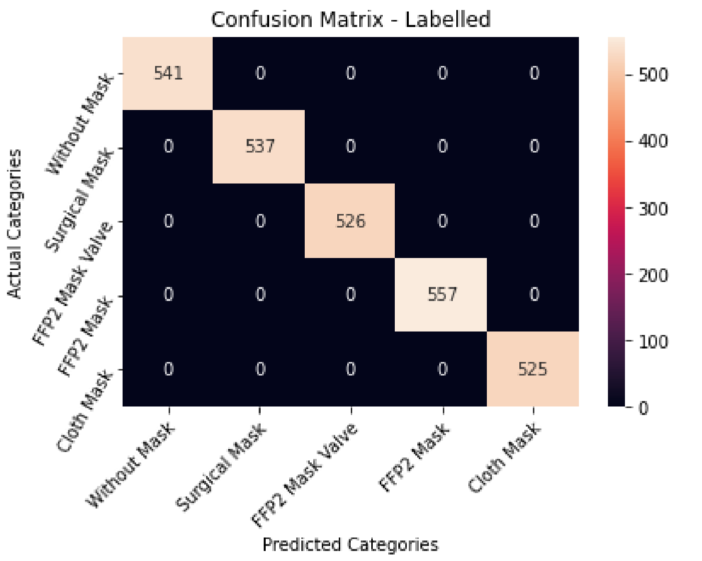


Part 2: After K- Fold Implementation

Training Classification Report and Confusion Matrix:

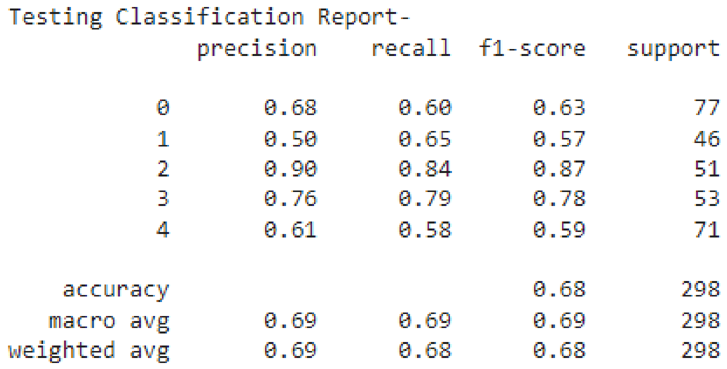


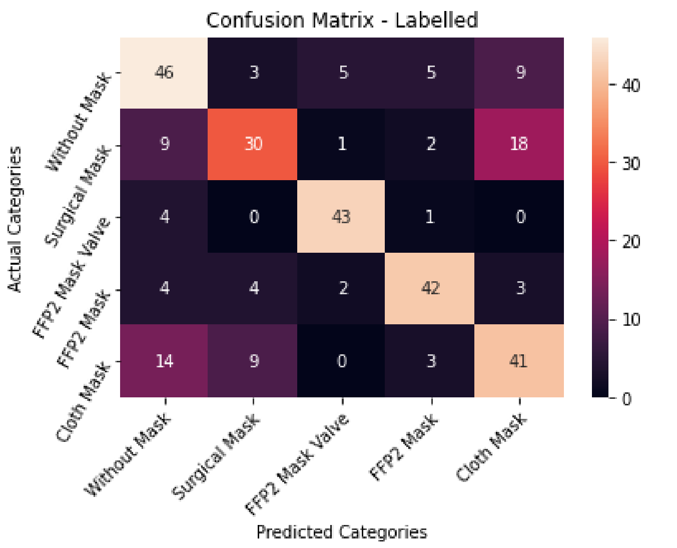
```
Training Classification Report-
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       541
           1       1.00      1.00      1.00       537
           2       1.00      1.00      1.00       526
           3       1.00      1.00      1.00       557
           4       1.00      1.00      1.00       525

    accuracy                           1.00      2686
   macro avg       1.00      1.00      1.00      2686
weighted avg       1.00      1.00      1.00      2686
```



-

Testing Classification Report and Confusion Matrix:


```
Testing Classification Report-
              precision    recall  f1-score   support

           0       0.68      0.60      0.63        77
           1       0.50      0.65      0.57        46
           2       0.90      0.84      0.87        51
           3       0.76      0.79      0.78        53
           4       0.61      0.58      0.59        71

    accuracy                           0.68       298
   macro avg       0.69      0.69      0.69       298
weighted avg       0.69      0.68      0.68       298
```



6. References:

Dataset References:

https://www.kaggle.com/datasets/a71dfe0333dcabd1827ca3d6dcfd62d43785f83d3c38321b4113339a14f780e9?select=train
https://www.kaggle.com/datasets/a71dfe0333dcabd1827ca3d6dcfd62d43785f83d3c38321b4113339a14f780e9?select=test
https://www.kaggle.com/datasets/a71dfe0333dcabd1827ca3d6dcfd62d43785f83d3c38321b4113339a14f780e9?select=valid
https://github.com/chandrikadeb7/Face-Mask-Detection/tree/master/dataset
https://github.com/balajisrinivas/Face-Mask-Detection/tree/master/dataset

https://www.google.com/search?q=people+wearing+ffp2+mask&rlz=1C1CHBFenIN8401N840&sxsrf=AOaemvKKrYHhWB4Ir1t47qNJt73xqnFn-g:1636268752963&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjtJuO2IX0AhV3m HIEHaB6CYgQ AUoAXoECAEQAw&biw=1366&bih=610&dpr=1

https://www.google.com/search?q=people+wearing+cloth+mask&tbm=isch&ved=2ahUKEwiT-a6P2IX0AhUyg3IEHQGDATMQ2-cCegQIABAA&og=people+wearing+cloth+mask&gsIcp=CgNpbWcQAzIFCAAQgAQ6BWgiEO8DECc6BggAEAcQHjoICAAQCBAHEB5Q5whyRigix10AHAAeACAAVmIAYoFkgECMTCYAQCgAQGgAQtnd3Mtd2/6LWItZ8ABAQ&sclient=img&ei=03qHYdPDFrKGytMPgYaGmAM&bih=610&biw=1366&rlz=1C1CHBFenIN8401N840

https://www.google.com/search?q=people+wearing+N95+mask&rlz=1C1CHBFenIN8401N840&source=lnms&tbm=isch&sa=X&ved=2ahUKEwi15KyFIbOAhUQheAKHUQ6BsYQAUoAXoECAEQAw&biw=1366&bih=635&dpr=1

https://www.google.com/search?q=people+wearing+ffp2+with+valve+mask&rlz=1C1CHBF_enUS965US965&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiYsfCTi5D3AhVLXM0KHeOCDzIQ_AUoAXoECAEQAw&biw=1536&bih=688&dpr=1.25

Other References

https://github.com/patelrutwik1999/Face-Mask-Detection_comp6721

https://github.com/balajisrinivas/Face-Mask-Detection

https://github.com/chandrikadeb7/Face-Mask-Detection
https://www.textiq.com/blog/how-to-measure-ai-recall-vs.-precision#:~:text=Recall%20refers%20to%20the%20fraction,a%20search%20finds%20relevant%20items.

https://machinelearningmastery.com/k-fold-cross-validation/

https://www.google.com/url?sa=i&url=http%3A%2F%2Fscikit-learn.org%2Fstable%2Fmodules%2Fcross_validation.html&psig=AOvVaw0AVDkLEz7cYg_dBDOAR3Y7&ust=1649896012049000&source=images&cd=vfe&ved=0CAsQjhxqFwoTCMiQ8t_jj_cCFQAAAAAdAAAAABAD