COMP 6721 Applied Artificial Intelligence (Winter 2022)
Project Assignment,
Part I & II

<u>Team Members</u>

1. **Sejal Chopra** (40164708) - Data Specialist, responsible for creating, pre-processing, loading & analyzing the datasets
2. **Sagar Sanghani** (40186043) - Training Specialist, responsible for setting up and training the CNN
3. **Anushka Sharma** (40159259) - Data Evaluation, responsible for analyzing, evaluating, and applying the generated model.

Table of Contents:

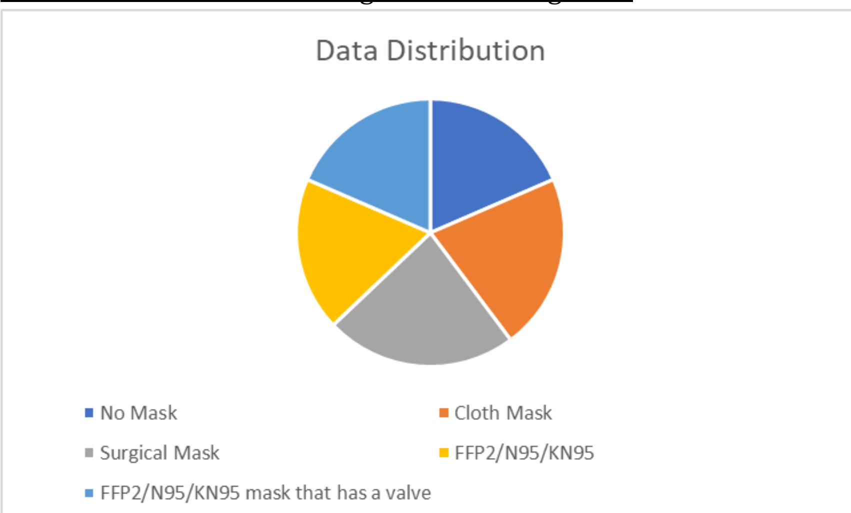
A. **Dataset Description:**

The dataset consists of images collected from various sources for the following categories of masks:

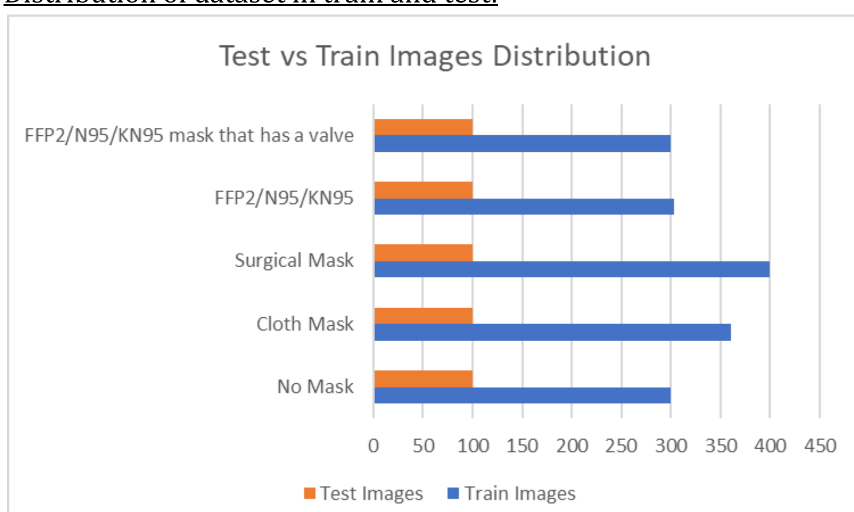
1. Person without a face mask
2. Person with a “community” (cloth) face mask
3. Person with a “surgical” (procedural) mask
4. Person with a “FFP2/N95/KN95”-type mask (you do not have to distinguish between them), and
5. Person with a FFP2/N95/KN95 mask that has a valve.

In total we have collected 2163 images across the above mentioned categories in testing and training data combined. The dataset has been divided into 1500 images for training and 500 images for testing across all classes. The data has been arranged in one folder where the different categories are distinguished by the likeliness of the image name. The images have been normalized and resized to 224x224x3. The dataset used for this project can be found here.

Distribution of dataset among different categories:



Distribution of dataset in train and test:



In order to normalize the data it is required that we find the Standard deviation and mean of our images dataset. This helps us to get consistent results when we apply our model on new unseen data. The formula used for calculating mean and standard deviation is as follows:

1. Mean : We calculate the mean by dividing the sum of the pixel values by the total count-number of pixels in the dataset computed as len(df) * image size*image size

2. Standard Deviation: We calculate the standard deviation using the equation : total_std = sqrt(psum_sq/count-total mean **2)

The resultant values are as follows:

- Mean: tensor([0.1880, 0.1771, 0.1691])
- Standard Deviation: tensor([0.3164, 0.3016, 0.2958])

B. Architecture

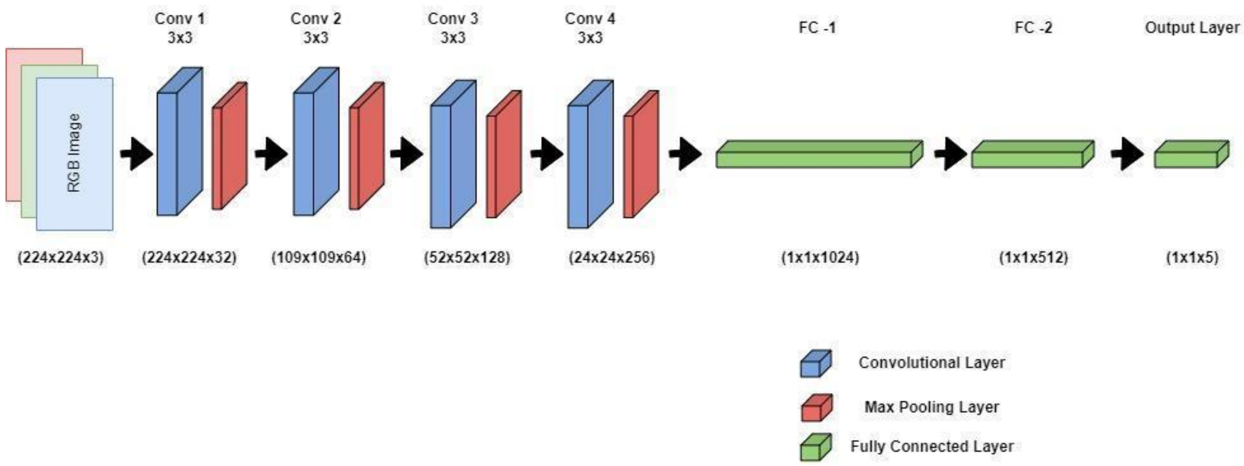
Layers:

Our CNN Model contains mainly 4 different layers:

1. Convolutional layers
2. Batch Normalization layers
3. Pooling layers
4. Fully Connected layers

Model

For the image classification of masks into different categories, we trained a model which consisted of 6 hidden layers and one output layer. Model has 4 Convolutional Layers, Batch Normalization layers, Max Pooling layers, 2 fully connected layers. The model is designed in such a way that after each Convolutional layer there is a Batch Normalization layer to speed up the training process followed by a Max Pooling layer with a filter of size 2 and stride 2. After each max pooling layer the number of features are halved. The size of the filter in each of the Convolutional layers is 3 with the stride of 1 and no padding. The activation function for all the layers is the Rectified linear unit (ReLU). After convolutions, the features are flattened and they are fed to the fully connected layers of size 1024, which is followed by another fully connected layer of size 512. This 512 units fully connected layer is then connected to the final output layer of 5 units. Figure x shows the same architecture.



CNN Model Architecture

- Conv1 - 32 Filters
- Conv2 - 64 Filters
- Conv3 - 128 Filters
- Conv4 - 256 Filters
- FC1 - 1024 Units
- FC2 - 512 Units

Training:

The size of the input images is 244x244x3. For training we used Adam Optimizer and CrossEntropyLoss as our optimizer and loss function respectively. Model was trained for 45 epochs on 80% of the entire dataset, in mini-batches of size 32. The training and test split was a random split to avoid imbalance of classes during the training phase. The learning rate was set to 0.001. After 45 epochs the training accuracy was 87%.

C. Data Evaluation

As achieved from training, we obtained a well trained model which then is used to test new images of dataset and analyze their results. The procedure of splitting the dataset for training and testing have been automatically done via python code rather than doing it manually. This test data set is then passed through our trained model and the number of correct predictions are returned along with the model's accuracy. All statistics of precision, recall, F1-Measure are being displayed using some predefined functions of the scikit library which we are showing in the Classification Report. More specifically, True Positives, False Positives, True negatives and False Negatives are used to predict the metrics of a classification report. The number of correct and incorrect predictions are summarized with count values and broken down by each class and shown in the confusion matrix.
Also a random image is passed in our trained model and results show which label the image belongs to.
The Outputs are:

a) Passing the dataset test images through trained model:

```
Testing:
Correct prediction: 382/433 and accuracy: 0.8822170900692841 and loss: 0.016994255244318936
```

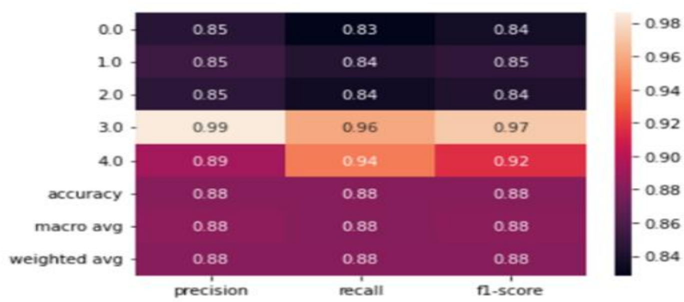
b) Accuracy:

```
Accuracy :  0.8822170900692841
```

c) Classification report (Also specific results):

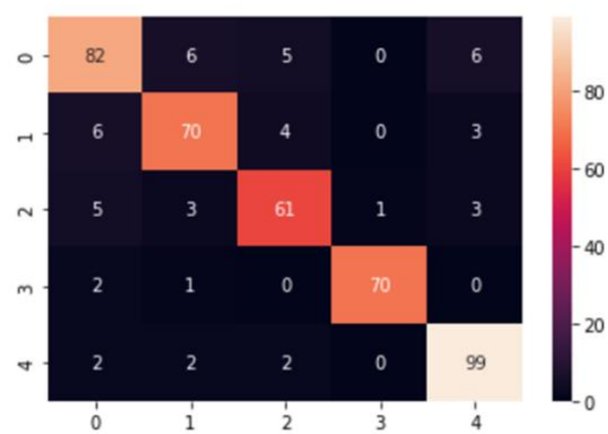
```
prec -- [0.845360824742268, 0.8536585365853658, 0.8472222222222222, 0.9859154929577465, 0.8918918918918919]
recall -- [0.8282828282828283, 0.8433734939759037, 0.8356164383561644, 0.958904109589041, 0.9428571428571428]
fscore -- [0.836734693877551, 0.8484848484848484, 0.8413793103448277, 0.9722222222222222, 0.9166666666666667]
support -- [99, 83, 73, 73, 105]
```

	precision	recall	f1-score	support
0.0	0.85	0.83	0.84	99
1.0	0.85	0.84	0.85	83
2.0	0.85	0.84	0.84	73
3.0	0.99	0.96	0.97	73
4.0	0.89	0.94	0.92	105
accuracy			0.88	433
macro avg	0.88	0.88	0.88	433
weighted avg	0.88	0.88	0.88	433



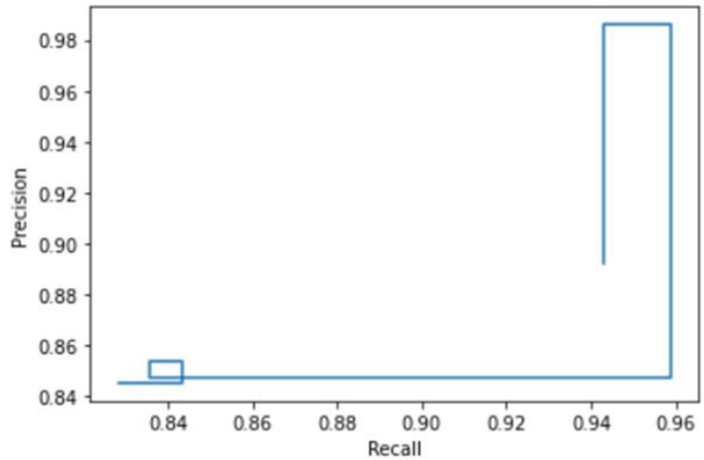
d) Confusion Matrix:

```
[[82  6  5  0  6]
 [ 6 70  4  0  3]
 [ 5  3 61  1  3]
 [ 2  1  0 70  0]
 [ 2  2  2  0 99]]
```



<Figure size 432x288 with 0 Axes>

e) Precision vs Recall graph




```
<Figure size 432x288 with 0 Axes>
```

PART II

D. K Fold Cross Validation

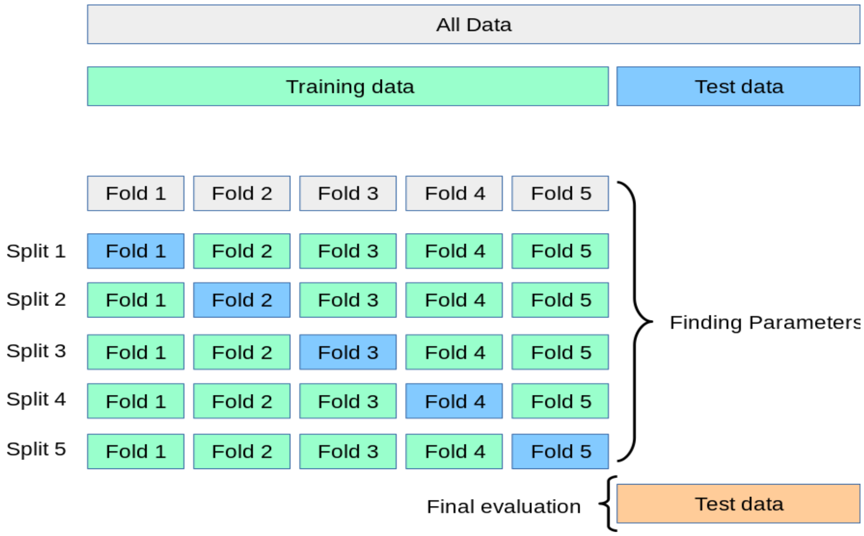
Cross - validation is a common statistical method used to estimate the skill of machine learning models. It helps to gauge the ability of a model to predict on unseen data. The K in K fold cross validation refers to the number of divisions the data is split into K=10 is a common value used for cross validation.

The general procedure for K fold Cross Validation is as follows:

1. Shuffle the dataset randomly.
2. The data is divided into K groups
3. For each unique group:
    1. Take the group as a test data set
    2. Take the remaining groups as a training data set
    3. Fit a model on the training set and evaluate it on the test set
    4. Retain the evaluation score and discard the model
4. Summarize the skill of the model using the sample of model evaluation scores

A poorly chosen value for k may might under or over estimate the ability of the model, .The choice of k is usually 5 or 10, but there is no formal rule. As k gets larger, the difference in size between the training set and the resampling subsets gets smaller. As this difference decreases, the bias of the technique becomes smaller.

The value of K that we have selected in our model is 10 and the learning rate is 0.001.



5-fold Cross Validation Example

E. BIAS DETECTION AND ELIMINATION

Our goal was to analyze our model for any two of the following biases: age, gender, or race. The bias for which we analyzed our model was "gender bias" and "age bias".The process which we followed to check for any possible gender bias was as follows. The first step involved separating our testing data into separate "male" and "female" sets which further contain mask categories.For the next step, we evaluated the performance of our model for the male dataset first, and similarly, we checked our model's performance against the female dataset. For age bias, we follow the same strategy and divide our training and test set into 3 categories which are kids, adults, and elders, and classify the total test images into these 3 different folders/categories.

Evaluation Metrics for the gender subclass –MALE and Female:

Pre Training Results

1. MALE

Testing:

Correct prediction: 21/24 and accuracy: 0.875 and loss: 0.014262354622228457

	precision	recall	f1-score	support
Cloth-Mask	0.83	1.00	0.91	5
N95-Mask	0.67	1.00	0.80	4
N95with Valve-Mask	1.00	0.75	0.86	4
No-Mask	1.00	0.83	0.91	6
Surgical-Mask	1.00	0.80	0.89	5
accuracy			0.88	24

	precision	recall	f1-score	support
macro avg	0.90	0.88	0.87	24
weighted avg	0.91	0.88	0.88	24

Confusion matrix, without normalization
[[5 0 0 0 0]
 [0 4 0 0 0]
 [0 1 3 0 0]
 [0 1 0 5 0]
 [1 0 0 0 4]]

2. FEMALE

Testing:
Correct prediction: 32/39 and accuracy: 0.8205128205128205 and loss: 0.052822151245215

	precision	recall	f1-score	support
Cloth-Mask	0.88	0.78	0.82	9
N95-Mask	1.00	0.67	0.80	9
N95with Valve-Mask	0.67	0.75	0.71	8
No-Mask	0.86	0.86	0.86	7
Surgical-Mask	0.56	0.83	0.67	6
accuracy			0.77	39
macro avg	0.79	0.78	0.77	39
weighted avg	0.81	0.77	0.78	39

Confusion matrix, without normalization
[[7 0 2 0 0]
 [0 6 1 0 2]
 [0 0 6 1 1]
 [0 0 0 6 1]
 [1 0 0 0 5]]
<Figure size 432x288 with 0 Axes>

3. KIDS

Testing:
Correct prediction: 58/60 and accuracy: 0.9666666666666667 and loss: 0.003220078969995181

	precision	recall	f1-score	support
Cloth-Mask	1.00	0.96	0.98	23
N95-Mask	0.67	1.00	0.80	2
N95with Valve-Mask	0.75	1.00	0.86	6
No-Mask	1.00	0.88	0.94	25
Surgical-Mask	0.80	1.00	0.89	4
accuracy			0.93	60
macro avg	0.84	0.97	0.89	60

```
    weighted avg       0.95      0.93      0.94        60

Confusion matrix, without normalization
[[22  0  1  0  0]
 [ 0  2  0  0  0]
 [ 0  0  6  0  0]
 [ 0  1  1 22  1]
 [ 0  0  0  0  4]]
<Figure size 432x288 with 0 Axes>
```

4. **ADULTS**

```
Testing:
Correct prediction: 149/171 and accuracy: 0.8713450292397661 and loss:
0.019520324660323517
                   precision    recall  f1-score   support

        Cloth-Mask       0.67      0.86      0.75        21
         N95-Mask       0.62      0.88      0.72        24
N95with Valve-Mask       0.70      0.84      0.76        25
          No-Mask       1.00      0.88      0.93        56
     Surgical-Mask       0.94      0.64      0.76        45

         accuracy                           0.81       171
        macro avg       0.78      0.82      0.79       171
     weighted avg       0.84      0.81      0.81       171

Confusion matrix, without normalization
[[18  1  2  0  0]
 [ 0 21  2  0  1]
 [ 1  2 21  0  1]
 [ 1  3  3 49  0]
 [ 7  7  2  0 29]]
<Figure size 432x288 with 0 Axes>
```

5. **ELDER**

```
Testing:
Correct prediction: 31/37 and accuracy: 0.8378378378378378 and loss: 0.12118556048419024
                   precision    recall  f1-score   support

        Cloth-Mask       0.73      0.89      0.80         9
         N95-Mask       0.80      0.80      0.80         5
N95with Valve-Mask       1.00      1.00      1.00         6
          No-Mask       1.00      1.00      1.00         8
     Surgical-Mask       1.00      0.78      0.88         9

         accuracy                           0.89        37
```

	precision	recall	f1-score	support
macro avg	0.91	0.89	0.90	37
weighted avg	0.91	0.89	0.89	37

Confusion matrix, without normalization
[[8 1 0 0 0]
 [1 4 0 0 0]
 [0 0 6 0 0]
 [0 0 0 8 0]
 [2 0 0 0 7]]
<Figure size 432x288 with 0 Axes>

<u>Post Training Results</u>

1. **MALE**

Testing:
Correct prediction: 22/24 and accuracy: 0.9166666666666666 and loss: 0.006568064913153648

	precision	recall	f1-score	support
Cloth-Mask	1.00	1.00	1.00	5
N95-Mask	1.00	1.00	1.00	4
N95with Valve-Mask	1.00	0.75	0.86	4
No-Mask	1.00	1.00	1.00	6
Surgical-Mask	0.83	1.00	0.91	5
accuracy			0.96	24
macro avg	0.97	0.95	0.95	24
weighted avg	0.97	0.96	0.96	24

Confusion matrix, without normalization
[[5 0 0 0 0]
 [0 4 0 0 0]
 [0 0 3 0 1]
 [0 0 0 6 0]
 [0 0 0 0 5]]

2. **FEMALE**

Testing:
Correct prediction: 36/39 and accuracy: 0.9230769230769231 and loss: 0.012245329526754526

	precision	recall	f1-score	support
Cloth-Mask	1.00	0.78	0.88	9
N95-Mask	0.88	0.78	0.82	9
N95with Valve-Mask	0.73	1.00	0.84	8
No-Mask	1.00	0.86	0.92	7
Surgical-Mask	0.86	1.00	0.92	6
accuracy			0.87	39

	precision	recall	f1-score	support
macro avg	0.89	0.88	0.88	39
weighted avg	0.89	0.87	0.87	39

Confusion matrix, without normalization
[[7 0 2 0 0]
[0 7 1 0 1]
[0 0 8 0 0]
[0 1 0 6 0]
[0 0 0 0 6]]
<Figure size 432x288 with 0 Axes>

3. KIDS

Testing:
Correct prediction: 55/60 and accuracy: 0.9166666666666666 and loss: 0.01224370946486791

	precision	recall	f1-score	support
Cloth-Mask	0.86	0.83	0.84	23
N95-Mask	1.00	0.50	0.67	2
N95with Valve-Mask	0.62	0.83	0.71	6
No-Mask	1.00	0.92	0.96	25
Surgical-Mask	0.67	1.00	0.80	4
accuracy			0.87	60
macro avg	0.83	0.82	0.80	60
weighted avg	0.89	0.87	0.87	60

Confusion matrix, without normalization
[[19 0 3 0 1]
[0 1 0 0 1]
[1 0 5 0 0]
[2 0 0 23 0]
[0 0 0 0 4]]
<Figure size 432x288 with 0 Axes>

4. ADULTS

Testing:
Correct prediction: 154/171 and accuracy: 0.9005847953216374 and loss: 0.016240646093212373

	precision	recall	f1-score	support
Cloth-Mask	0.81	1.00	0.89	21
N95-Mask	0.80	0.83	0.82	24
N95with Valve-Mask	0.77	0.96	0.86	25
No-Mask	1.00	0.82	0.90	56
Surgical-Mask	0.95	0.91	0.93	45
accuracy			0.89	171

macro avg 0.87 0.91 0.88 171
weighted avg 0.90 0.89 0.89 171

Confusion matrix, without normalization
[[21 0 0 0 0]
[2 20 2 0 0]
[0 1 24 0 0]
[2 2 4 46 2]
[1 2 1 0 41]]
<Figure size 432x288 with 0 Axes>

5. **ELDER**

Testing:
Correct prediction: 34/37 and accuracy: 0.918918918918919 and loss: 0.10431274087042422

	precision	recall	f1-score	support
Cloth-Mask	0.80	0.89	0.84	9
N95-Mask	1.00	0.60	0.75	5
N95with Valve-Mask	0.86	1.00	0.92	6
No-Mask	1.00	0.88	0.93	8
Surgical-Mask	0.90	1.00	0.95	9
accuracy			0.89	37
macro avg	0.91	0.87	0.88	37
weighted avg	0.90	0.89	0.89	37

Confusion matrix, without normalization
[[8 0 1 0 0]
[1 3 0 0 1]
[0 0 6 0 0]
[1 0 0 7 0]
[0 0 0 0 9]]
<Figure size 432x288 with 0 Axes>

Conclusion

We observe that previously we have less number of testing data due to which there was some fluctuation in the evaluation matrix i.e. accuracy, precision, recall, and f1-score. We saw biasing in the Female and Kids category. The testing data was less in size and poor in quality due to which accuracy was less promising but after adding the images and balancing the dataset among 5 different classes we eliminate biasing in some classes.

Measure	Male	Female
Accuracy	0.8958	0.8803
Precision	0.8476	0.9050
Recall	0.8311	0.8968
Fscore	0.8222	0.8962

Measure	Kid	Adult	Old
Accuracy	0.9055	0.8976	0.9324
Precision	0.8185	0.8631	0.9158
Recall	0.9114	0.8759	0.8988
Fscore	0.8432	0.8662	0.8980

Post Training Metrics

References:

The dataset was collected from the following sources:

1. https://unsplash.com/s/photos/surgical-mask
2. https://www.dreamstime.com/photos-images/cloth-mask.html?pg=6
3. https://unsplash.com/s/photos/n95-mask
4. https://unsplash.com/s/photos/cloth--mask
5. https://www.kaggle.com/c/imagerecognition1081/data
6. https://www.google.com/search?q=n95+mask+with+valve+people+images&tbm=isch&ved=2ahUKEwjF7dTigNb2AhUqr3IEHUvUA28Q2-cCegQIABAA&oq=n95+mask+with+valve+people+images&gs_lcp=CgNpbWcQAzoHCCMQ7wMQJzoFCAAQgAQ6BggAEAcQHjoGCAAQCBAeOgYIABAFEB46BAgAEBhQ2gNYoRtg3BxoAHAAeACAAV-IAagIkgECMTWYAQCgAQGqAQtnd3Mtd2l6LWltZ8ABAQ&sclient=img&ei=m843YoW0IqreytMPy6iP-AY&bih=601&biw=1280&rlz=1C1RXQR_enCA975CA977
7. https://www.google.com/search?q=images+of+people+with+n95+masks&rlz=1C1RXQR_enCA975CA977&sxsrf=APq-WBueXM495faExycd-YgC-250YBn5Rw:1647839294426&source=lnms&tbm=isch&sa=X&ved=2ahUKEwi016net9b2AhUHkokEHUo7BrgQ_AUoAXoECAEQAw&biw=1280&bih=601&dpr=1.5

References for training the model :

1. Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. 2017. arXiv: 1412.6980 [cs.LG].
2. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. 2015. arXiv:1502.03167 [cs.LG]
3. PyTorch, Training a Classifier. URL : https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html
4. https://machinelearningmastery.com/k-fold-cross-validation/

References for testing the model:

1. https://www.coursera.org/lecture/deep-neural-networks-with-pytorch/training-validation-and-test-split-pytorch-8lFxA
2. https://www.scikit-yb.org/en/latest/
3. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html#sklearn.metrics.classification_report
4. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.multilabel_confusion_matrix.html#sklearn.metrics.multilabel_confusion_matrix

5. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_curve.html#sklearn.metrics.precision_recall_curve