UNIVERSITÉ Concordia UNIVERSITY

GINA CODY SCHOOL OF ENGINEERING AND COMPUTER SCIENCE

COMP 6721 – APPLIED ARTIFICIAL INTELLIGENCE

PROJECT PHASE – I

Team ID :- NS_10

Presented to :-

Dr. René Witte

AI FACE MASK DETECTOR

Authors : -

Harshkumar Nileshkumar Patel (40165709) – Data Specialist

Chirag Hasmukhbhai Patel (40160656) – Evaluation Specialist

Harshil Jayeshbhai Patel (40163431) – Training Specialist

1. Dataset :-

The dataset for accomplishing the task of developing an AI Face Mask Detector is constructed by collecting images from various resources available online like Kaggle and several other web resources[1][2][3][4][5].The dataset contains over 2000 images which is partitioned as follows into 5 categories (class) - Cloth Mask , N95 Mask, N95 Mask with Valve, Surgical Mask, No Mask , each class contains average of 400 images.

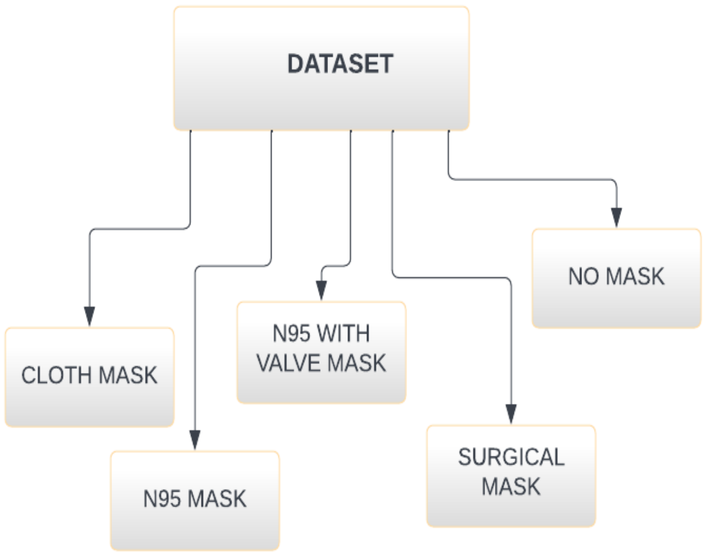


Fig 1.1 Dataset

Now it is necessary to pre-process the dataset before feeding it as input to the CNN and this task is accomplished by following several key points like compressing the images before loading them to train the model, keeping balanced number of images in each class and loading the data using the method load_data , using the raw image data will not provide the desired results therefore it is necessary to scale each image present in the dataset to a specific size and hence tensors of the same dimensions will be obtained which will be fed as input to the model.

Our Dataset contains various images of different dimensions so as a part of pre-processing all the images are scaled to a particular dimension in this case (250 X 250). Also, for uniformness, every image is converted to RGB format and then we apply normalization so that each pixel value falls in similar range.

2. CNN Architecture

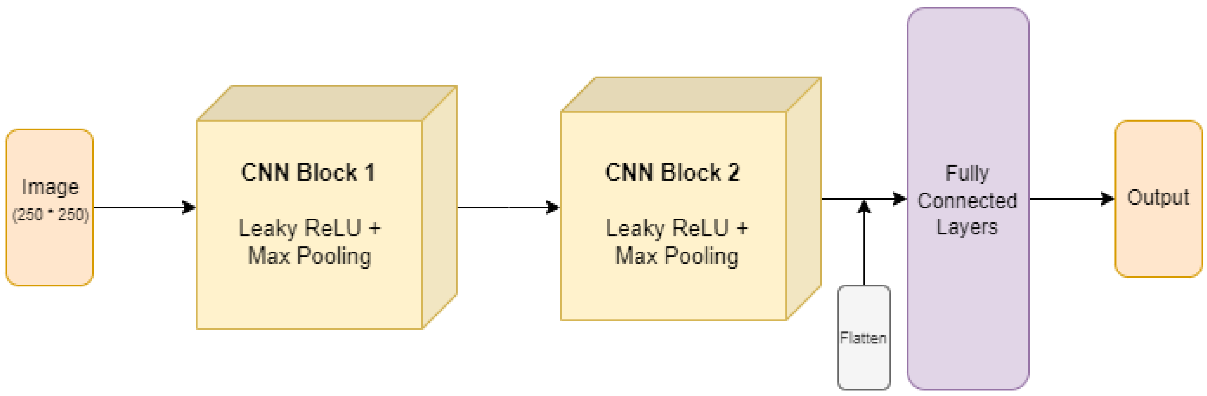


Fig 2.1 CNN Architecture

There are four types of layers for a convolutional neural network: the convolutional layer, the pooling layer, the ReLU correction layer and the fully-connected layer[6].

Convolution Layer :- The purpose of this layer to Its purpose is to detect the presence of a set of features in the images received as input. This is done by convolution filtering: the principle is to "drag" a window representing the feature on the image, and to calculate the convolution product between the feature and each portion of the scanned image. It provides a feature map, which tells us where the features are in the image.

ReLU correction layer :- The ReLU correction layer replaces all negative values received as inputs by zeros. It acts as an activation function.

Pooling Layer :- The pooling operation consists in reducing the size of the images while preserving their important characteristics. The pooling layer reduces the number of parameters and calculations in the network. This improves the efficiency of the network and avoids over-learning.

Fully Connected Layer :- This type of layer receives an input vector and produces a new output vector. To do this, it applies a linear combination and then possibly an activation function to the input values received. A flatten layer is used to flat the tensor which has 3 dimensions. The flatten layer converts the tensor to one-dimensional. Then 3 linear added to reduce the size of the tensor and learn the features[7].

In this project the CNN architecture consists of two CNN blocks and each block contains two convolution layers and one max pooling layer. Here for the purpose of optimization we tried using different optimization algorithm, but ASGD (Average Stochastic gradient descent) gave better performance , it is ordinary stochastic gradient descent that records an average of its parameter vector over time. That is, the update is the same as for ordinary stochastic gradient descent, but the algorithm also keeps track of parameter vector and when optimization is done, this averaged parameter vector takes the place of w

$$\bar{w} = \frac{1}{t}\sum_{i=0}^{t-1} w_i .$$

[8].The pre-set hyperparameter values used are as follows :-

Learning Rate:- 0.01, Epochs :- 20, Batch Size :- 25

3. Evaluation

To train the model using constructed dataset we have trained the model with 20 epochs.

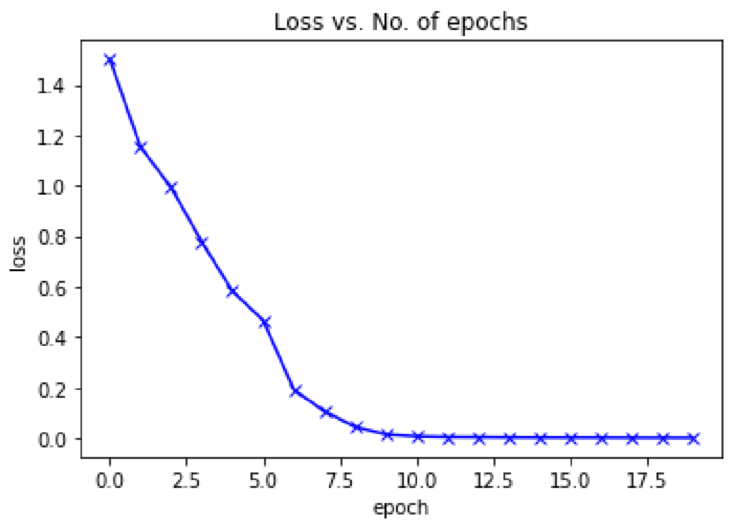
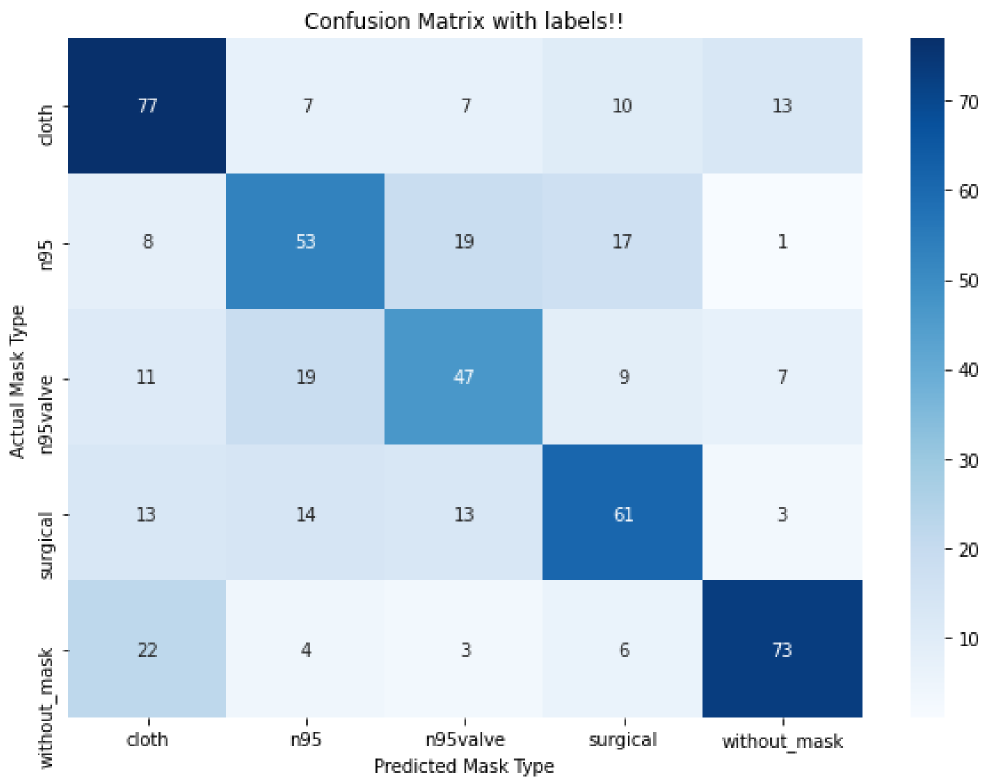


Fig 3.1 Loss vs No. of epochs

The above image shows the plot for loss vs. number of epochs and as the number of epochs are increasing the loss (training) decreases.

The confusion matrix to describe the performance of the model is as below :-



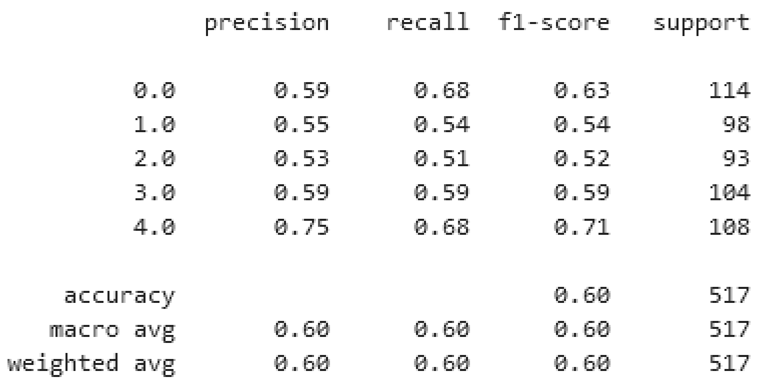
Actual \ Predicted	cloth	n95	n95valve	surgical	without_mask
cloth	77	7	7	10	13
n95	8	53	19	17	1
n95valve	11	19	47	9	7
surgical	13	14	13	61	3
without_mask	22	4	3	6	73

Fig 3.2 Confusion Matrix

As it can be seen the confusion matrix shows some of the images are not classified according to their respective classes, there are several reasons for that like data in some of the classes are imbalanced, model is not efficient enough to accurately classify images of various classes.

The accuracy of our model is **60.15%**

We evaluated the performance of the model using metrics like Recall, Precision, Accuracy and F1-Score for the testing phase.



```
              precision    recall  f1-score   support

         0.0       0.59      0.68      0.63       114
         1.0       0.55      0.54      0.54        98
         2.0       0.53      0.51      0.52        93
         3.0       0.59      0.59      0.59       104
         4.0       0.75      0.68      0.71       108

    accuracy                           0.60       517
   macro avg       0.60      0.60      0.60       517
weighted avg       0.60      0.60      0.60       517
```

Fig 3.3 Classification Report

4. Reference

1.https://www.kaggle.com/andrewmvd/face-mask-detection

2.https://www.kaggle.com/prasoonkottarathil/face-mask-lite-dataset

3.https://www.kaggle.com/datasets/ashwingupta3012/male-and-female-faces-dataset

4.https://humansintheloop.org/mask-dataset-download/?submissionGuid=90621ead-d768-4016-b823-179ee4908f34

5.https://www.kaggle.com/datasets/anglinab/facemasks

6.https://towardsdatascience.com/understand-the-architecture-of-cnn-90a25e244c7

7.https://medium.com/thecyphy/train-cnn-model-with-pytorch-21dafb918f48

8.https://en.wikipedia.org/wiki/Stochastic_gradient_descent#cite_note-23