COMP - 6721
APPLIED ARTIFICIAL INTELLIGENCE

PROJECT-1 REPORT

Submitted To: Prof. Dr. René Witte

Team Id: AK_14

Sivakumaran Malli Janardhanan (40155790)

Saghana Mahesh Sarma (40198979)

Shubhang Khattar (40163063)

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION:

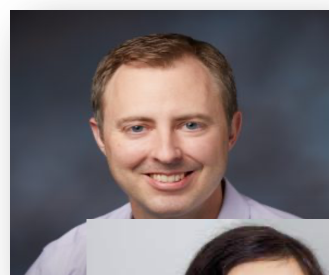
A great deal of scientific study involves the gathering and computation of measurement data. Datasets are becoming increasingly important, and they are typically viewed as the fundamental intellectual output of research. AI is an area in which datasets are used to power most of the initiatives.

For precision and accuracy, we must partition our dataset in a balanced manner. The entire dataset can be divided with 80 percent of the data used for training and the remaining 20% used for testing. The model is trained using a dataset, and the AI creates an algorithm or pattern to distinguish items during this phase. The trained model is then tested with unknown data that has been removed from the dataset to assess the model's or algorithm's performance using a variety of performance matrices.

In this experiment, we used samples for each of the three classes: person with N95 mask, N95 with valve, surgical mask, and person without mask. We chose a well-balanced dataset with a variety of factors including age, colour, gender, and so forth. Some datasets came from data repositories like kaggle, while a significant number of pictures came from Google or the internet.

The training dataset consists of the following categories:

N-95 MASK



NO MASK

CLOTH MASK

SURGICAL MASK

N-95 MASK WITH VALVE

1. DATASET:

The dataset has been preprocessed to make them all the same size (128*128). The pathways of the datasets are specified and labelled before they are analysed. For our dataset, we utilised the following five labels:

0- Cloth Mask
1- N95 Mask
2- N95 Mask with Valve
3- No Face Mask
4- Surgical Mask

We split our dataset in half, using 80 percent for training and 20 percent for testing. Following a few epochs, we use 60% of the data for training, 20% for testing, and 20% for training once more. To train and test the data, we partition the dataset in new ways each time.

Because each class's dataset is varied in size, the model may produce unexpected results. As a result, weights are supplied to the loss function, which assigns a weight to each class. The class with the fewest photographs may be given higher weights. We'll utilise the cross-entropy loss - a loss - to train our model to go along with it. To adjust the weights based on the training data, we utilise Adam optimiser.

We scaled and turned the photos into tensors with labels and normalised the dataset using the torch approach to import the data for our dataset. The data is then loaded in batches of 32 for training and 64 for testing.

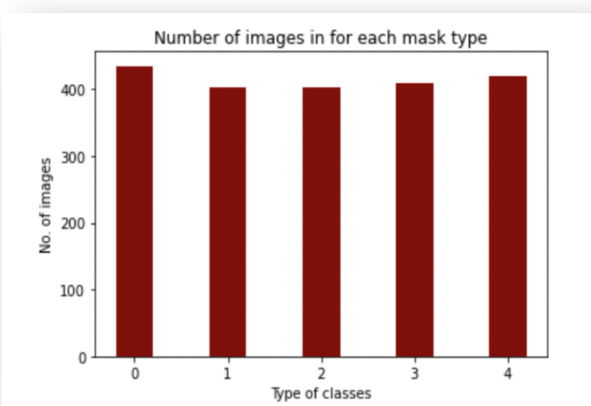


Figure 1- Images in each label

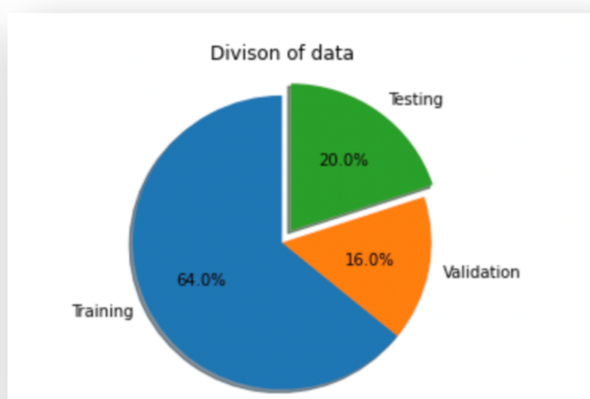


Figure 2 – Division of Data

2. CNN ARCHITECTURE:

The model was trained using a convolutional neural network. We employ three convolutional neural network layers with 32-3 x 3 filters in the first layer, 64-3 x 3 filters in the second layer, and 128-3 x 3 filters in the third layer. There are three more max pooling layers, each with a size of 2*2. Our first convolutional 2d layer with RGB channel receives our input image with channel 128*128. Activation function and maxPooling take place in our third convolutional layer, which produces output with 256 channels. The LeakyLeRu activation function is used. Finally, we have the output labels.

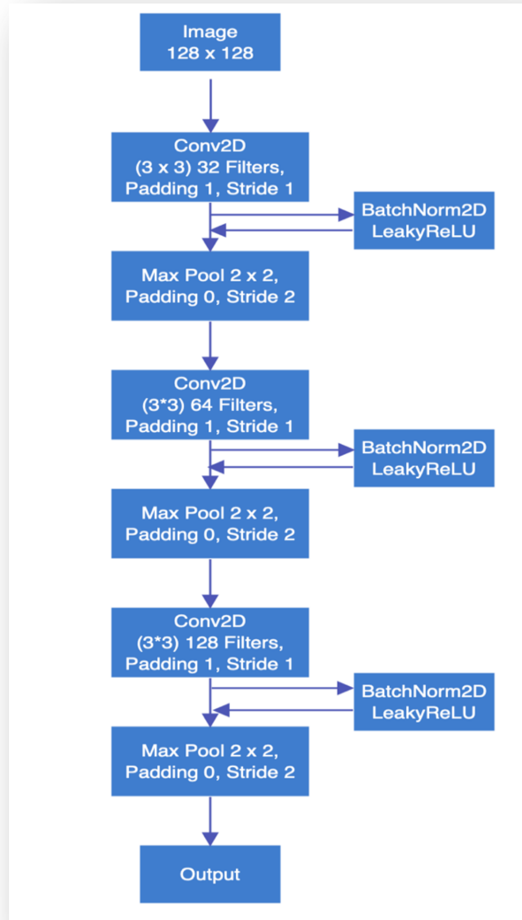


Figure 3 – CNN Architecture

The model is finished with a single unit and sigmoid activation, which is ideal for binary classification. We utilise the "use saved model" parameter to see if the dataset's accuracy and validation loss have already been calculated.

If it is untrue, we repeat the epochs and obtain the data. For each epoch, we train the data while also measuring the accuracy and validation loss on the validation data.

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 thirty epochs in this project We can state that some of the photos have been falsely labeled due to imbalance data by utilising a confusion matrix.

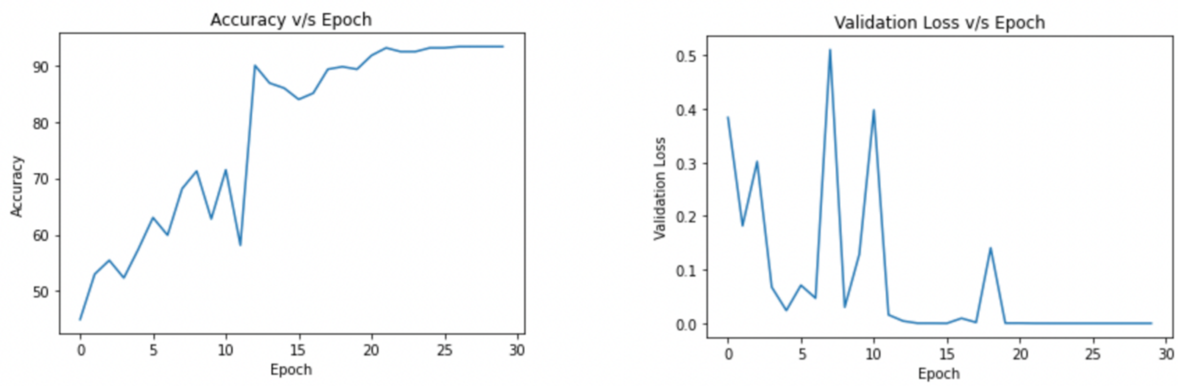


Figure 4 – Accuracy vs epoch vs validation loss

We can conclude from the above image that there are very few changes in accuracy after fifteen epochs and the validation loss is less.

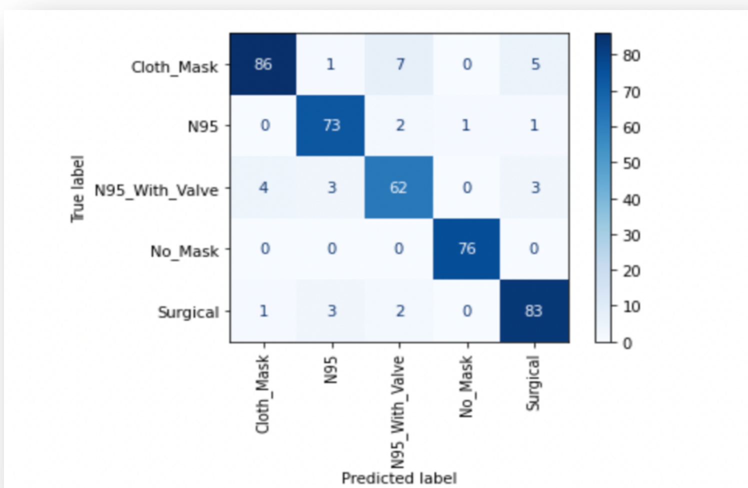


Figure 5 – True Label vs Predicted Label

In the next phase of the project, we are going to increase the data for the masked person and not a person category and also add a validation layer of K-fold cross to remove bias and improve project performance.

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://medium.com/swlh/introduction-to-cnn-image-classification-using-cnn-in-pytorch-11eefae6d83c
10. https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529