# Introduction

Corona Virus Disease (COVID-19) has been declared a global pandemic by the World Health Organization (WHO) in 2020 due to its rapid spread. The use of a face mask has a significant impact on the propagation of newly spread virus. Utilizing face masks successfully decreases the infection development rate of COVID-19 by 40%, according to German researchers. Furthermore, according to a report in the journal of the National Academy of Sciences (PNAS), wearing face masks is considerably more important than social distancing and home isolation regulations in limiting COVID-19 dissemination and infection [1]. During the outbreak, it is everyone's job to use face masks in public settings to reduce the spreading the virus. Not only does this necessitate individual voluntary participation, but it also necessitates specific forms of monitoring and administration. As a result, the detection of face mask and classification of mast type has become the hot topic in research domain.

Currently, the artificial intelligence and deep learning playing an important role to resist the COVID-19 virus. Numerous studies based on these technologies have been published to reduce the spread of this virus like the human respiratory patterns recognition, chest X-ray image-based detection and segmentation of city scan images. However, before the spread of COVID-19 disease, there are few studies for the recognition of face mask. Basically, the detection of face mask is related to security fields to overcome the terrorist activities and illegal behaviors. But after the outbreak of COVID-19 virus, researchers perform number of face mask detection studies for monitoring and administration purposes. In the proposed study of [42], the largest dataset of face mask was generated from real world environment and developed a model for the recognition of face mask. In a proposed study [24], author extract the face features by ResNet50 pretrained model and then classify them by decision tree, SVM and ensemble models. The mentioned studies can classify the faces with or without face mask very effectively. But in the situation like poor lightning, crowding areas that effect the performance of the above model. In addition, there are different cases of face covering like scarf, sponge mask, and cotton mask that classify as mask but badly reduce the protection of subject against virus.

Here, we proposed a light weight model for the classification of faces with or without face masks. The proposed model will light weight compared to the existing models and will perform significant result in term of accuracy and other evaluation measures.

## Related Work

Face identification has gone through a transformation from classical to deep learning as a normal object detection problem. The following are some examples of classical computer vision algorithms: Viola and Jones [40] suggested a Haar-based Adaboost face detection algorithm. The Adaboost algorithm is easy to use and can handle two-classification and multi-classification without overfitting. The approach, on the other hand, is sensitive to noise, has a slow response time, and is subject to false positives. Ma [25] et al used four types of Haar features to characterize the face relationship based on Adaboost, which lowered the detection time and missed detection rate. Unfortunately, in the presence of occlusion and profile, the model detection effectiveness suffers.

Pedro Felzenszwalb [5] created the Deformable Part-based Model (DPM) method in 2010, that solved the multi-posture and multi-angle challenge of the face using the component model approach of multi-component and pictorial structure. However, this strategy necessitates the creation of a synthetic incentive pattern for the item, a significant amount of labor, and it is not ubiquitous. To put it another way, traditional object identification methods are restricted to the effective description of picture characteristics, and can only depend on expertise to manually extract features and then create object detection classifiers based on the findings and combined with the sliding window. and with a lot of complicated steps, low precision, and poor real-time performance.

Jagadeeswari et al. [11] compared different pretrained deep learning model like the ResNet, MobileNet, and VGG16 with the combination of numerous optimizers including Adam, SGD, and ADAGRAD. He got the maximum accuracy on MobileNet model with Adam Optimizer.   Han et al. [12] used a single-shot detector to conduct object detection in a supermarket, concentrating on real-time face mask recognition (SSD). Vinitha et al. [13] employed a CNN with MobileNetV2 architecture, as well as a library of OpenCV, TensorFlow, Keras, and PyTorch, to determine whether or not persons were wearing a face mask. To identify real-time face masks, Nagrath et al. [14] employed SSDMNV2 and SSDMNV2. For face mask detection, Ge et al. [15] suggest LLE (Locally Linear Embedding) - CNNs.

# Findings

The finding of the proposed work is following:

1. The dataset of face mask detection is small and there is chance of model overfitting. The proposed work will use the Siamese Network that are designed for small amount of data.
2. The classification model for face mask in literature are very large in size. We proposed the light weight, robust customized neural Network model for face mask monitoring.

# Methodology

## General Approach

In this section we will discuss the selected dataset for our proposed study, preprocessing steps and the architecture of our developed models.

### **Dataset**

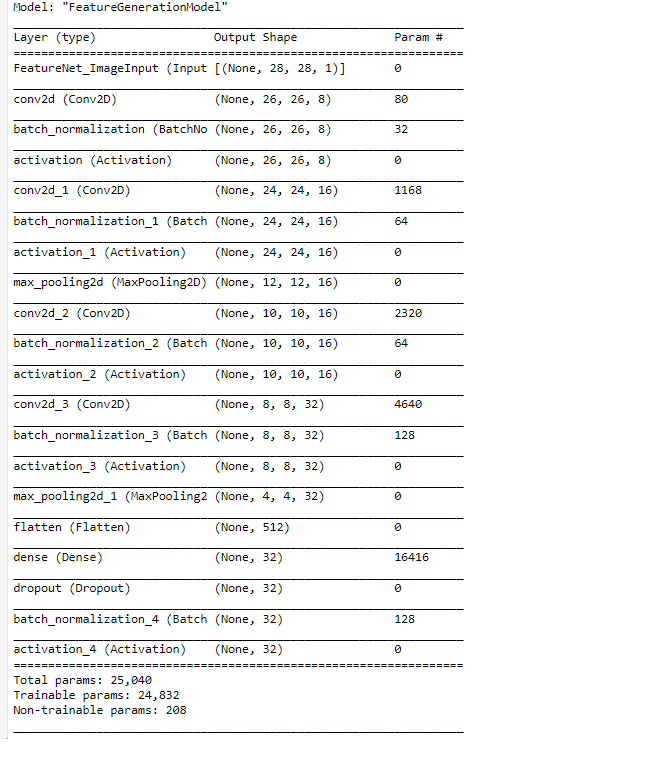
We used the Face Mask Detection dataset from Kaggle for the detection of face mask. The downloaded dataset was based on the two types of the images. The first category of the images in dataset belong to the persons who wear the face mask and the second category have the images of the persons with out face mask. Resultantly we have the two types of labels for our dataset: face mask and no face mask.

### **Preprocessing**

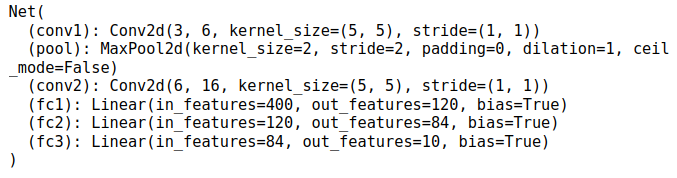
In the preprocessing phase, we resize all the images to the shape of 128X128. After resizing the all the images, we normalized the images by converting them into the range of 0-1. The labels of the images were in the categorical form. We used the label encoder for the conversion of categorical labels into numerical format. After the preprocessing of the images, the dataset was used for the training if the model.

### **Model**

Here we proposed two types of models including the Siamese Network model and Neural Network base model. Firstly, we develop a Siamese network base model that was based on the input layer, two convolutional, followed by the flatten, dense, batch normalization and output layer. In the Siamese network-based model (Model 1), each convolutional layer is followed by the batch normalization layer, and activation layer with ReLU activation function. The architecture of feature extracting portion is show in below figure.



We also develop a neural network-based model for the classification of mask faces. In our neural network model (Model 2), we used two connective convolutional layers followed by the three linear layers and one output layer. The architecture of proposed model is also shown in below figure.



### **Evaluation Measures**

For the estimation of out trained model, we used the accuracy, precision, recall and f1-score as evaluation measured. We calculate each evaluation measure by the below formulas.

Eq. 1

Eq. 2

Eq. 3

Eq. 4

## Design

In design phase, we will discuss the development environment, used libraries, programming language and all other related setting for the training of the proposed models.

### **Environment**

All the experiment s were performed by using the python programming language. The python 3.7 version was use to perform the experiments. A Conda environment was established with python version 3.7 for the training of the models. All the essentials library were installed in the developed environment.

### **Model Framework**

From the well-known framework for deep learning models, we used the PyTorch framework for the development and the training of the proposed models.

### **Libraries**

We installed different libraries in our established environment. The list of all core libraries with their version number is listed in below table.

|  |  |
| --- | --- |
| Library Name | Version |
| PyTorch | 1.7.1 |
| TensorFlow | 2.3.0 |
| Matplotlib | 3.5.2 |
| sk-learn | 1.0.1 |
| Pandas | 1.3.5 |
| Numpy | 1.19.0 |

# Experiments

In experiment section, we perform two experiments on Model 1 and Model 2 respectively. For performing the experiment, we prepare our dataset for the training and the testing of the Model 1 and Model 2.

## Train Test Split

To prepare the training and testing set, we used the train-test-split function of scikit-learn library. We split he dataset with 70% and 30% in training and testing set respectively. Train test split function split the 70% images from each class as training set and rest of the 30% as testing set. After split the data into training and testing set, we have the 6000 samples in training set and 1553 samples in testing set.

## Experiment 1

In experiment 1, we trained our Model 1 (Siamese Network based) with 6000 number of training samples. As our problem is binary classification problem, we used the binary cross entropy as loss function with Adam optimizer. We train our Model 1 with 20 epochs and 32 batch size.

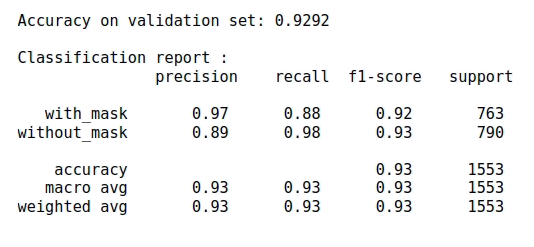
After the training of the model, we test our model with 1553 test sample and model showed the 83% testing accuracy. The graph of accuracy and loss on training and validation set is shown below:

|  |  |
| --- | --- |
|  |  |

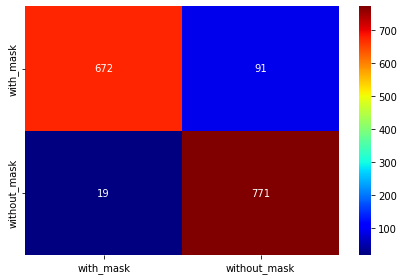
## Experiment 2

In Experiment 2, we train our model 2 with 6000 training samples. We used the 0.1% images from training set and validation set to validate the model during training. Model 2 was also train with binary cross entropy loss function with learning rate of 0.001. We used the SGD optimizer for the weight tunning of the model. The model was train with 20 epoch and 0.9 hyper parameter value of momentum.

After the training of the model, the model was tested on the 1553 samples of the testing data. Model 2 showed the 93% accuracy on test set. The complete classification report of Model 2 is shown below:



We also plot the confusion metrics of the Model 2 for testing data. Confusion Metrics showed that the trained Model 2 accurately classify the 672 and 771 samples from mask and with out mask class respectively. The confusion metrics of model 2 is shown below:



# Discussion

In the proposed study, we took the two dimensions for the classification of mask faces. We train two models from Siamese Network domain and Neural Network domain. Both models were trained using the face mask detection dataset and they showed the 83% and 93% testing accuracies respectively. Although, both models showed the significant results but the NN model followed by the Siamese model due to its nature. Siamese network models are usually used to find the similarity of image with other image and beneficial with small amount of data. They match the similarity and predict on the basis of similarity. NN model find the probability of class suing the image feature. As the signature matching and face matching are similarity problems and Siamese networks perform well on these problems but our problem is not similarity problem that’s why NN model perform over the Siamese Model.

References