Approaching Transfer Learning To Identify Correctly Worn Facial Masks

by

Tausif Sadid
17101083
Md Sadik Hossain
17101387
Sudip Kumar Sengupta
17301105
Serazam Munira Khondaker
17101066

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
June 2021

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

[illegible]	Sadik Hossain
Tausif Sadid 17101083	Md Sadik Hossain 17101387
[illegible]	[illegible]
Sudip Kumar Sengupta 17301105	Serazam Munira Khondaker 17101066

Approval

The thesis/project titled “Approaching Transfer Learning To Identify Correctly Worn Facial Masks” submitted by

1. Tausif Sadid (17101083)
2. Md Sadik Hossain (17101387)
3. Sudip Kumar Sengupta (17301105)
4. Serazam Munira Khondaker (17101066)

Of Spring, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June 2, 2021.

Examining Committee:

Supervisor:
(Member)

[illegible]

Amitabha Chakrabarty, Ph.D.
Associate Professor
Department of Computer Science and Engineering
BRAC University

Co-Supervisor:
(Member)

Tanzim

Tanzim Reza
Lecturer
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam, Ph.D.
Associate Professor
Department of Computer Science and Engineering
BRAC University

[illegible]

Head of Department:
(Chair)

Sadia Hamid Kazi, Ph.D.
Associate Professor
Department of Computer Science and Engineering
BRAC University

Abstract

Climate change has made the outbreak of diseases, especially airborne and contagious diseases much more extreme. Therefore, wearing masks has become essential in protecting ourselves not only from very common airborne diseases like cold, flu but also from very dangerous diseases like COVID-19 or similar. Wearing face masks properly would significantly reduce the spread of most of these diseases. In our research, we would like to propose a way to distinguish whether people are wearing facemasks properly or not by using image data. The data-set we collected included roughly 6000 images of people wearing masks both correctly and incorrectly, among which roughly 50% of it is collected locally from Bangladesh and the rest from other countries around the globe. The idea is to detect properly masked faces from images using different Deep Neural Network architectures. Throughout our analysis, we have implemented three models, ResNet50, Inception v3, and MobileNet v2 while comparing the results when SVM is used as a classifier and when Softmax is used in place of it. The results emerged with ResNet50 with an SVM classifier bringing out the best results by reaching an accuracy of 97.41% accuracy making the model highly reliable and stand out from the rest. This research aims to provide a better understanding of all the models and their architectures while providing statistical results from our data-set when these models are being used to distinguish between pictures of a properly masked person and that of an improperly masked.

Keywords: COVID-19; Machine Learning; Deep Neural Network; Climate Change; Image Data; Face Masks; ResNet50; Inception v3; MobileNet v2; SVM; Softmax.

Acknowledgement

Firstly, all praise to the God Almighty who has bestowed us with strength and wisdom in being able to complete this thesis without any major interruption.
We would like to express our gratitude to our advisor Dr. Amitabha Chakrabarty sir and co-advisor Md Tanzim Reza for their kind support and advice in our work who have helped us with their immense support and knowledge whenever we sought help. With their expertise, guidance, and blessings we have been able to complete the thesis paper.
We are highly indebted to our family, friends, and everyone involved directly or indirectly with the thesis. Without their constant support, it may not be possible. With their kind support and prayer, we are now on the verge of our graduation.
And finally, we would like to thank our parents, without their constant love and words of encouragement this thesis would not have been possible.

Table of Contents

List of Figures

List of Equations

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AQI	Air Quality Index
CNN	Convolutional Neural Network
ILSRVRC	The ImageNet Large Scale Visual Recognition Challenge
LBP	Local Binary Pattern
ReLu	Rectified Linear Unit
ResNet	Residual Neural Network
SVM	Support Vector Machine
VGG	Visual Geometry Group

Chapter 1

Introduction

1.1 Problem Statement:

Wearing a mask is an incredibly necessary procedure that anyone could follow since this basic measure will greatly minimize the risk of airborne disease transmission. When the person wearing the mask coughs, sneezes, speaks or raises his voice, masks provide a shield to help prevent respiratory droplets from spreading into the air and onto other people. During outbreaks of airborne diseases such as the most recent COVID-19, if the public maintained the proper usage of masks the number of affected would be a significantly lower value. Not everyone is properly using a face mask; this raises the risk of such diseases, thus, with the aid of image data and machine learning techniques, we can recognize improperly worn masks and consequently alert the user. Guidelines for touch and droplet precautions for healthcare workers caring for suspected COVID-19 patients have been provided by the World Health Organization (WHO), although airborne precautions were originally recommended by the US Centers for Disease Control and Prevention (CDC). The spatial separation rule of 1- to 2-meter (about 3-6 feet) is fundamental to droplet precautions and implies that large droplets do not move further than 2 meters (about 6 feet) [1]. However, The use of masks is still a problem in society. Residents are unwilling to use masks due to many factors such as misinformation and misinterpretation, politics, values, states of mental well-being, and immunity for herds. Furthermore, individuals are getting tired of using masks [2]. Therefore with the proper use of face masks with the aid of technology, we aim to track safety violations and facilitate the use of face masks to maintain a secure working atmosphere.

1.2 Research Objectives:

The main reason for our research is to improve the quality of detecting incorrectly wearing face masks and make them correct which may bring a better life for humans. Machines are a huge part of our lives and therefore they should be used in the best possible way for us. We have chosen this area of study due to the problems mentioned in the problem statement and still lack of applications to solve that. Focusing on that, we have tried to figure out some good solutions for this by using many techniques in our proposed model. The contribution to our research is that, we have gathered as many as possible varieties(gender, age, color) of masks in our data-set in the context of Bangladesh. It covers more than 50% of our data-set.

1.3 Thesis Outline

Chapter 1 is Introduction. Here we have explained the problem statement, the research objectives and a short methodology of our work before a deeper dive into it.
Chapter 2 is Literature Review. This consists of the background study of our thesis work as well as related work we found to be of use to aid us in our research on the topic.
Chapter 3 is Workflow and Data Analysis. We go into further details on how we prepared our research, as well as explain the different layers and classifiers used in the different algorithms we implemented.
Chapter 4 is Algorithms. In this section we explain the 3 algorithms, namely, ResNet50, Inception v3 and MobileNet v2 along with their architectures and mathematical equations involved.
Chapter 5 is Results and Accuracy. We included our graphical representations of the accuracy and loss graphs of all the algorithms and classifiers we implemented in our thesis work in this section. Additionally we also added confusion matrices as well as comparisons between the algorithms.
Chapter 6 is Conclusion. In the final section of our paper we provide a brief summary of our work as well as some possible future improvements to our work we can take into consideration moving forward.

1.4 Methodology

For the analysis of the data, we chose to use ResNet50, Inception v3, and MobileNet v2 as our transfer learning models as it considerably reduces training time as well as requiring less training data in order to increase performance [3]. A portion of the data collected was randomly divided into 3 parts of train test and validation to analyze upon, among which 74.78% of the data was taken as the training data-set, 15.63% as the validation data-set, and the remaining 9.60% as the test data-set. The train data-set included 2197 correctly worn masked pictures and 2241 incorrectly worn masked, making a total of 4438 pictures. The validation data-set included 464 correctly worn masked pictures and 464 incorrectly worn masked, making a total of 928 pictures. The test data-set included 275 correctly worn masked pictures and 295 incorrectly worn masked, making a total of 570 pictures. All pictures taken were shaped to fit the transfer learning model with an input shape of 224x224 pixels. The extracted images are then passed through VGG16 models for data augmentation by passing through Dropout, Flatten and 1024 units of Dense layer before it is passed to a SoftMax layer with Relu activation. Instead of training the data on the original network, Dropout deactivates the neurons randomly at each training phase and then we train the data on the network with dropped out nodes [4]. Each node in the network is associated with a probability p in each training iteration, whether to keep it in the network or to disable it (dropout) from the network with a probability of 1-p. That implies that only p fractions of times have been modified for the weights associated with the nodes since nodes are only active p times during training in order to reduce overfitting. The Adam optimizer is then used to attain the global minimum when training the algorithm. This will assist us to get out of local minima and hit global minima if the algorithm is stuck in local minima [5].

Chapter 2

Literature Review

In one of the papers, they used a dual-stage Convolutional Neural Network (CNN) architecture that can identify masked and unmasked faces and can be combined with pre-installed CCTV cameras. The methods they used to implement this were:
1. Traditional Object Detection, where they used algorithms like Haar Cascade [6]. and HOG [7]. for object detection to detect masks.
2. Convolutional Neural Networks, where they apply Computer Vision tasks in order to extract different features of the image, and
3. Modern Object Detection Algorithms which involved Multi-Stage Detectors and Single-Stage Detectors using CNN based detection algorithms.
After running the face detection and the processing block of the architecture to ensure that the distribution and nature of training data match the expected input during the final deployment [8]. They managed a highly accurate and robust Face Mask Detection System with the help of these procedures. RetinaFace was selected as their Face Detector in Stage 1, while the NASNetMobile based model was selected as the Face Mask Classifier in Stage 2 [9]. The result demonstrated high efficiency in detecting facial masks over a wide variety of angles in photographs of many faces [10].
Another paper used a data-set containing images which were already cropped around the face, negating the need for face detection [11]. They, however, did need to correct the rotation of the face to remove the masked region efficiently. They detected 68 facial landmarks using Dlib-ml open-source library [12]. According to the eyes location, we apply a 2D rotation to make them horizontal. Then deep features were extracted using the VGG-16 face CNN descriptor from the 2D images which are trained on the ImageNet data-set [13]. VGG-16 contains 16 layers, however only the feature maps (FMs) at the last convolutional layer were considered. These features will be used in the following in the quantization stage. This work is then sent for a similarity measure using RBF Layer and Quantization layer. The final result had an astounding recognition rate of 91.3%. Due to the best features being derived from the last convolutional layer of the VGG-16 model, this high precision was achieved.
In another paper, the authors used the integration of deep transfer learning and classical machine learning algorithms to create a face mask detection model [14]. The model consists of two components, the first one is ResNet which is used as a feature extractor and the second one is the classification stage where ResNet-50 was replaced by three traditional machine learning classifiers (SVM, decision tree, and ensemble). Three data-sets were used for training and testing and SVM

showed better results in all of them. However, using deeper transfer learning models for feature extractions and neutrosophic domains in the classification would have achieved better results. A relevant paper suggested a mobile device interface to allow anyone to have a smartphone and to be able to take a photo to check that their safety mask is appropriately placed on their face or not [15]. Using various types of traditional masks and diverse acquisition conditions, different research scenarios experimented with detectors and face-feature detectors. The reliability of the built system depends on the efficacy of the face and face-feature detectors employed. The designed approach incorporates haar-like attribute descriptors to distinguish both the face and main characteristics of the face from a cell phone acquisition focused on a camera. The purpose of the Haar-like features is to code the pixel material variations in the image.

In the next paper, the authors use Haar-based feature analysis in a hybrid computing scheme and a comparative study shows their effectiveness [16]. Here, the concept of their system together employs Haar-based facial features recognition techniques. Nowadays, Haar-related analysis methods are very much practiced to conduct face analysis. But the drawback of their system is that the current number of smartphone users in the world today is 3.5 billion, which is only 45.04% of the world's population and others cannot use it because of not having a smartphone. The related work involves Automated Face Recognition in this paper [17]. A machine that can identify a face, its voice, and also its computer contact, this is referred to as Human-Computer Interaction. It also offers us a detailed survey of common methods introduced for digital image face recognition. The related study entails an algorithm that offers a modern face-recognized technique attendance management system using an algorithm called LBP-Local Binary Pattern, in addition to data processing methods, such as data mixing and equalization of histograms enhancing the system's precision [18]. Here, some of the problems that hamper the accuracy of face recognition are discussed to improve the LBP codes, hence enhancing the accuracy of the overall method of face recognition.

In a recently published journal, they talked about artificial intelligence and its subsets, such as machine learning and deep learning, as well as deep learning frameworks and a simple implementation of a face mask detection system [19]. Hence, a research paper constructed a real-time automated model in public that detects whether people are wearing facemasks and keeping social distance in public locations and reports to the appropriate authorities using computer vision and the Raspberry Pi4 [20]. Coronavirus has caused global health crises, and facial masks are a basic type of virus control. As a result, a hybrid model is created combining traditional and deep machine learning and consists of two components. The first component uses Resnet50 for feature extraction, and the second uses ensemble support vector machine (SVM) techniques and a decision tree for mask classification processing. After examination, three data-sets are used. The second component is based on YOLO v2 and is meant to detect medical face masks. Two data-sets of medical face masks have been integrated into a single data-set for this study. The ideal number of anchor boxes was estimated using mean IoU to improve the object detection process. The healthcare system is in disarray. A number of preventative precautions are being taken to minimize the transmission of viruses, one of which is the usage of facemasks. Using Closed-Circuit-Television (CCTV) cameras, a system is developed to detect people who are not wearing facemasks in smart cities [21].

In another research, they tried to make a face mask assistant [22]. For that challenge, they start by extracting four features from the micro-photos of the face mask's gray level co-occurrence matrices (GLCMs). The K Nearest Neighbor (KNN) method is then used to create a three-result detection system. The results of validation studies reveal that on the testing data-set, their system can achieve an accuracy of 82.87 percent (as determined by macro-measures). The recall of Type I 'normal usage' and Type III 'not recommended' are 92.00 percent and 92.59 percent, respectively. They intend to enhance the detecting objects to include more mask kinds in future development. This research shows that the proposed mobile microscope system can be utilized as a helper for wearing a face mask, which could help in the fight against COVID-19.
In a research paper, the authors used Principal Component Analysis to construct a traditional machine learning method for recognizing masked and unmasked faces (PCA) [23]. According to the findings, a face without a mask had a higher recognition rate in PCA. It has been discovered that extracting features from a masked face is less difficult than extracting features from an exposed face. They discovered that wearing masks affects the accuracy of mask face classification using the PCA. When you put on a mask, your accuracy drops to 70%. In another paper, a unique GAN-based network was presented for removing mask objects from facial pictures [24]. For face mask categorization, the authors suggested a hybrid deep transfer learning model with machine learning methods in [25]. The proposed paradigm has two stages. 1) ResNet50-based feature extraction 2) SVM, decision tree, and ensemble-based classification To assess the suggested methodology, three data-sets were chosen as benchmarks.
In another paper [26], RetinaFaceMask is a high-accuracy and efficient face mask detector that they proposed. The proposed RetinaFaceMask is a one-stage detector that includes a feature pyramid network for fusing high-level semantic information with various feature maps, as well as a new context attention algorithm. Face mask detection will be the emphasis of this module. They also present a revolutionary cross-class object removal method. Results of their experiment show that RetinaFaceMask produces state-of-the-art outcomes with 2.3 percent and 1.5 percent on a public face mask data-set Face and mask detection precision were both greater than the baseline result, at 11.0 percent and 5.9 percent, respectively. The recall is higher than it was at the start. They also look into the likelihood of recall being higher than baseline. In addition, they look into the idea of using RetinaFaceMask in conjunction with a web browser. MobileNet is a light-weighted neural network for embedded or mobile devices.
Again, there is another research work where their proposed model is composed of two parts. The first component is made to extract features using Resnet50. The second component uses decision trees, Support Vector Machines (SVM), and an ensemble method to classify face masks. For this study, three face-masked datasets were chosen. [14]
The authors of [27] created a new facemask-wearing condition detection method. They were able to categorize three different types of facemask use. Correct facemask wear, wrong facemask wear, and no facemask wear are the three categories. In the face detection phase, the proposed method obtained 98.70 percent accuracy. To recognize the person, Sabbir et al [28] used Principal Component Analysis (PCA) on masked and unmasked facial recognition. They discovered that wearing masks

had a significant impact on the accuracy of facial resonation utilizing the PCA. When the detected face is disguised, the recognition accuracy declines to less than 70%. PCA was also utilized in [29]. The authors suggested a method for eliminating glasses from a frontal image of a human face. The removed part was recreated using PCA reconstruction and recursive error correction.
Various researchers and analyzers have recently concentrated on gray-scale facial images [30]. Others used AdaBoost , an outstanding classifier for training purposes, while others were totally constructed on pattern identification models, containing starting information of the face model [6]. Then came the Viola-Jones Detector, a breakthrough in face identification technology that allowed for real-time face detection. It had a number of issues, including the position and brightness of the face, which made it difficult to intercept. So, in a nutshell, it didn't work in dim or dim light. As a result, researchers began looking for a new alternative model capable of detecting faces as well as masks on the face.
Many data-sets for face detection have been created in the past to establish an impression of face mask detection methods. WiderFace [31], IJB-A [32], MALF [33], and CelebA [32] are examples of recent data-sets built from Internet pictures. In these data-sets, annotations are provided for current faces, as opposed to prior ones. Large data-sets are considerably more required for better training and testing data, as well as for performing real-world applications in a much more straightforward manner. This necessitates a variety of deep learning algorithms that can read faces and masks directly from the user's data.
In this study proposes the SSDMNV2 model for face mask identification, which uses the OpenCV Deep Neural Network (DNN), TensorFlow, Keras, and MobileNetV2 architecture as an image classifier [34]. SSDMNV2 uses OpenCV DNN, which includes SSD with ResNet-10 as a backbone and is capable of detecting faces in a variety of angles. While MobileNetV2 is employed, it gives lightweight and accurate classification predictions based on whether or not a mask is used. SSDMNV2 is capable of distinguishing between photos with frontal faces with masks and images with frontal faces without masks.

Chapter 3

Workflow and Data Analysis

3.1 Workflow

This Chapter details the way we proceed with our work. In 3.1, we have described our workflow in the form of a flowchart. Our work started with collecting as many images as possible from people wearing masks in our country and collecting a huge data set of people wearing masks. After collecting the images we have to resize all images into a single size of 224x224 pixels. It begins the process of Pre-processing. Pre-processing also includes the process of labeling the images and classifying them. We have used two algorithms as classifier layer, Softmax and Support Vector Machine(SVM). We are using VGG16 model to pre-train and use transfer learning. The next part is training using 3 different algorithms of CNN architecture, Resnet50, Inception, Mobilenet. The hardware specifications we used for running our models included a Ryzen 2200g processor, Nvidia gtx 1660 as GPU, 8gb of RAM, and a storage of 1TB HDD and 256 GB SSD. Finally, after running the models and obtaining the results, we compare the results we get from the 3 different algorithms and the 2 different classifiers and analyze the data furthermore by comparing these results with similar studies.

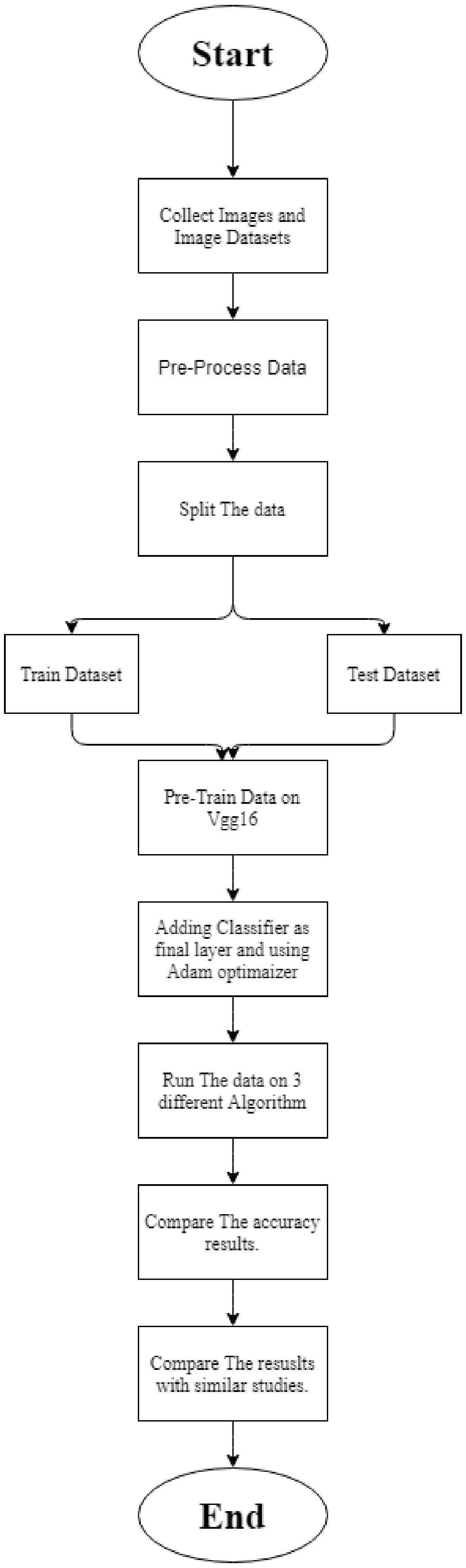

Start
Collect Images and Image Datasets
Pre-Process Data
Split The data
Train Dataset
Test Dataset
Pre-Train Data on Vgg16
Adding Classifier as final layer and using Adam optimaizer
Run The data on 3 different Algorithm
Compare The accuracy results.
Compare The resuslts with similar studies.
End

Figure 3.1: Workflow Diagram.

3.2 Data and Analysis

3.2.1 Data Collection & Statistics:

Data preparation is one of the most significant and difficult steps in any learning based research work. The reason is that each data-set is different and highly specific to the project. To work on the proposed model, we have made a data-set of people living both in Bangladesh as well as all around the world. Currently, almost everyone is using face masks to protect themselves from COVID-19 and we have collected around 6000 pictures of people wearing various types, shapes, material face masks that are used in the paper. We have a data-set of exactly 6108 pictures right now but in future, we have a plan to import data from outside sources as well so that our data-set can cover almost all types of masks throughout the world. There is also a mix of men and women wearing face masks correctly and incorrectly. We collected data of different age groups for more accurate results from every perspective.
It is to be mentioned that,we had collected almost 3000(3041 exactly) data and all of data is from Bangladesh. Among them there are 1538 correctly masked data(3.2) and 1503 incorrectly masked data(3.3). Again there are 792 women and 746 men have correctly masked and 773 women and 732 men are incorrectly masked. Again from total calculations, there are almost 3071 correctly(3.4) and 3037 incorrectly masked pictures(5.33) which we are currently working with. Among them, 1413 are men and 1658 are women wearing masks correctly as well as 1416 men and 1621 women are wearing masks incorrectly. In this data-set, Details on different age groups people wearing masks correctly and incorrectly are shown below.

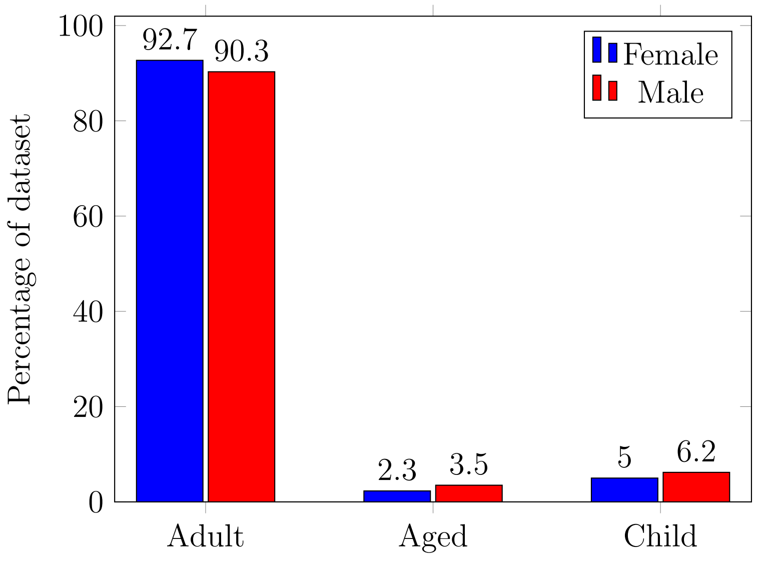


Figure 3.2: Correctly Worn Mask(Local data only)

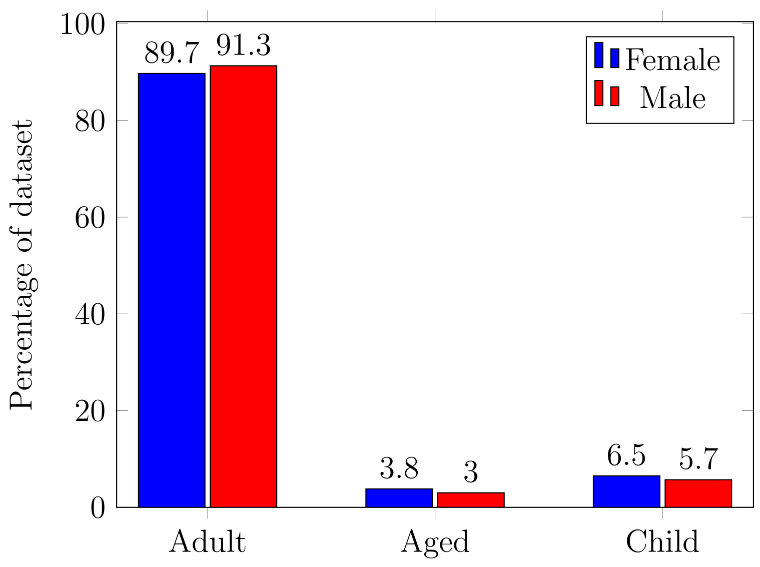


Figure 3.3: Incorrectly Worn Mask(Local Data only)

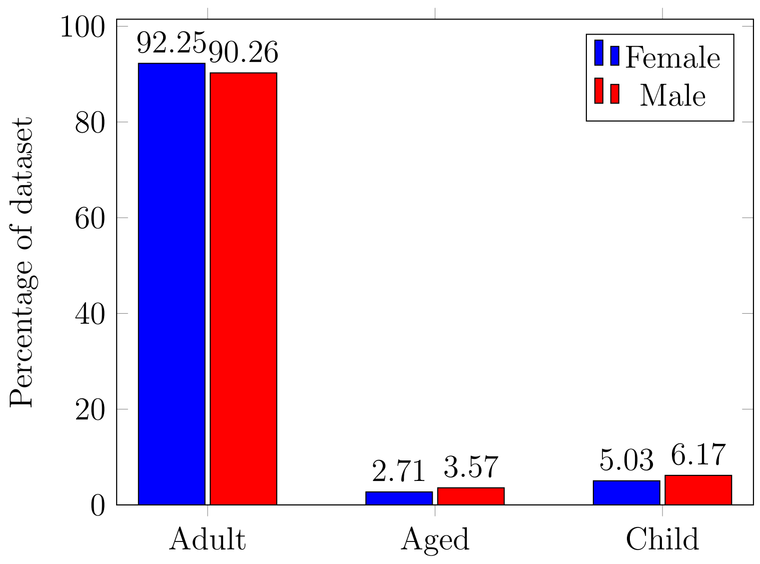


Figure 3.4: Correctly Worn Mask

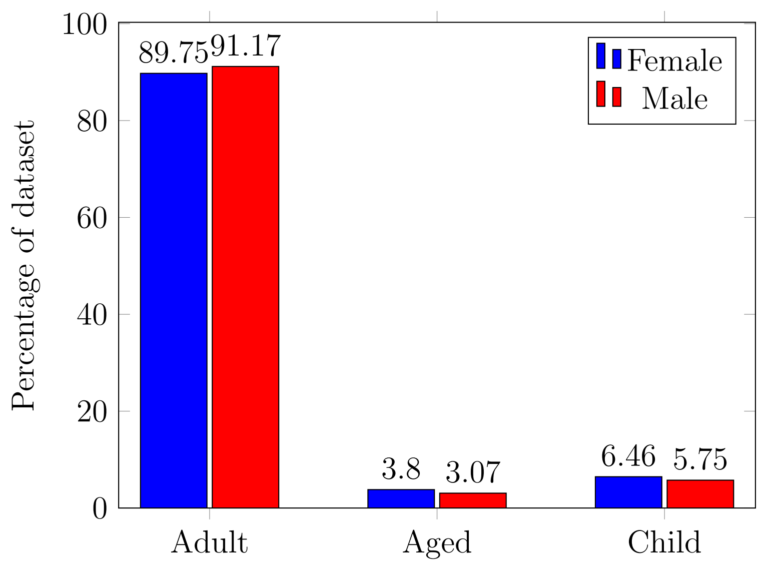


Figure 3.5: Incorrectly Worn Mask

3.3 Preprocessing:

We have resized all the images here for image training and testing.Image resizing means changing the scaling of image.It helps to pick less pixels , that helps in reducing the training time for the desired experiment.The more pixels that we need to use the more difficult it be will for the model as the pixels are our input for the model.This also helps us to rescale the images later for the purposes of zooming or shrinking the image. We have used ImageDataGenerator for our test train and validate data sets. In the training-
· rescale=1./255; where we are scaling the RGB value from the 0-255 to a 0-1 number
· rotation_range=30; The degree to randomly rotate the image
· shear_range=0.01;changing the angle in counter clockwise.value is given in degrees.
· zoom_range=[0.9, 1.25];This is the range for randomly zooming the images.
· horizontal_flip=True;Randomly flipping the images horizontally.
· vertical_flip=False);Randomly flipping the images vertically which we are not doing.
For the purposes of test and validate the data set who only just rescale the images to a 0-1 number.
All the inputs are taken in the size of 224x224 pixel form, with each batch consisting of 10 images.
Then the input goes into the models that we have created.
Our models before running on the 3 different algorithms are run on a pre-trained VGG16 model.

3.3.1 The Sequential layer:

This is the first of our layers in the model.Sequential is a one dimensional model that only takes input and output for a single input.It also can not be connected to other layers.It is used in our Resnet50 and mobilenetv2 model which adds different layers on to one another.

3.3.2 Global Average 2d pooling layer:

This layer which is proposed in the Network in Network (Lin & Chen, 2013) is used in our inception model [35]. It is designed for the purpose of acting at the behest of a fully connecting network in a CNN model.It generates one feature map for all of the corresponding categories of the classification task in the last layer.It takes average of each feature map and then adds the vector of those addition which are then sent to the next layer.It takes the global average pooling instead of the feature map.This makes the model much better from getting overfit than the other models used here.This also sums out all the spital information,which makes the input much more robust in terms of its spital transformation.

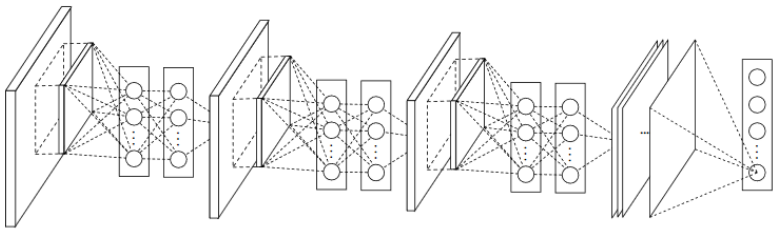


Figure 3.6: Full figure of a Network In Network architecture where three layers are staking onto a single layer of global average pooling.

3.3.3 Flatten layer:

After using a pooling layer we use the flatten layer.This is used in our models that have a matrix pooled value which is then flattened with this layer into a single column.It is then sent to the next layer.

3.3.4 Dropout layer:

This layer, acts as mask, randomly sets the value of some inputs of images to 0 which values are frequently shown in the data.This happens when the data-set is in training mode and happens in every step of training.Values are not dropped any other time unless instructed to.This action prevents overfitting.All of our models have this layer.

3.3.5 Dense layer:

This layer is known as the fully connected layer.After the flatten layer there are two dense layers.This layer does a linear operation onto the value of the inputs which are now in vector form.This two layers have activation function ReLu or one of the two classifiers in all of the models which we used here.

3.3.6 Activation Function: Residual learning(ReLu):

This activation function is used in all of our models because this activation function is very easy to use.It only makes sure that whether the value is less then 0 or not.This results in giving the derivative of this function in 0 or 1 which makes it even more easier to use. The expression for Relu is [36]-

$$f(x) = argmax(0, x), \text{ Where } f(x) = \begin{cases} 0 & \text{if, } x < 0 \\ 1 & \text{if, } x >= 0 \end{cases} \tag{3.1}$$

This ensures that the function does not suffer from the Vanishing gradient problem.This is why we used this activation function.

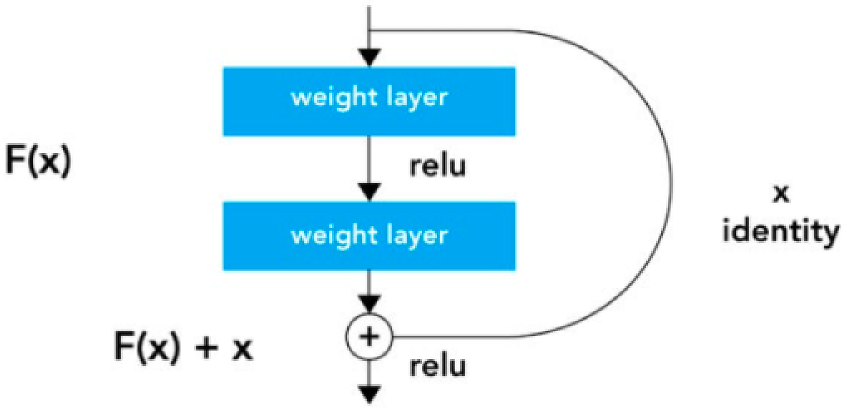


Figure 3.7: Residual learning: a building block

3.4 Classifier

3.4.1 Softmax

This is one of the classifiers we are using.Softmax is also an activation function as well.It is one of the most common forms of classifier which is used in deep learning architecture.It max all the values it receives what those values maybe and those values are turned into a sum of a number which will be always between 0 to 1.This is a probability function which turns the probability distribution of a vector distribution of a number into a real one.This is used as a one of the classifier to check the accuracy of the models.The equation for softmax is [37]-

$$f(x)_i = \frac{\exp(x_i)}{\sum_j^k \exp(x_j)} \tag{3.2}$$

Where $(x)_i$=The input vector.
x_i=The elements of the input vector.
x_j=The standard exponential function.
k=The number of multiclass classifiers.

This classifier is used as a multiclass classifier where the target class has the highest value.It then calculates the probabilities of other classes.

3.4.2 SVM

Support vector machine is another classifier that we use in all of our models.It tries to separate and group the data into a single class by maximizing the margin.It makes sure that the class it choose for the data behind a line known as the hyperline and the margin which is the closed dote to the hyper line is picked , so it can maximize the the margin.this applies for its linear functions.If there are some function that are not inside the hyperline even then it creates the best possible result for the maximizing margin which called a soft margin.For nonlinear cases SVM uses a much higher dimensional plane, which makes to classify the data for it much easier.
For linear cases the equation is given below-

$$\begin{array}{l} J(w,b,a) = \sum_{i=1}^{N} a_i + \frac{1}{2}w^T w - w^T \sum_{i=1}^{N} a_i d_i x_i - w^T \sum_{i=1}^{N} a_i d_i x_i - b\sum_{i=1}^{N} a_i d_i \\ \text{Where } a_i >= 0 \forall i \end{array} \tag{3.3}$$

For all our models in the final dense layer to classify our data of images, we use the Linear version of the SVM as one of our classifiers.

3.5 Optimizer

3.5.1 Adam

Adam optimizer (Kingma & Ba,2014) is used as an optimizer for gradient descent[25]. It's a first-order gradient-based optimization of stochastic objective functions. Gradient descent is used to make sure the cost for a function is less as possible by finding the values of function parameters. This particular algorithm is the combination of gradient descent with momentum algorithm and Root Mean Square propagation(RMSP) algorithm. Momentum used take the exponentially weighted average of the gradients and used to converse towards minima in a faster place. RMSP on the other hand used to take the exponential moving average. Adam combine this to algorithm by controlling the gradient decent so it has minimal oscillation by taking big enough steps to pass the local minima. This satisfies both of the algorithms features and reaches the global minima efficiently.
The equation for Adam is [38] -

$$W_{t+1} = W_t - m_t \left(\frac{a}{\sqrt{v_t - \varepsilon}} \right) \tag{3.4}$$

Here, W_t = weight at time t.
W_{t+1} = weight at time t + 1
m_t = Bias correlated weight
v_t = Bias correlated weight.
a = Learning rate.

$\varepsilon = +$ ve constant.

We choose this optimizer in all of our algorithms because it is very efficient and the memory needed to run this algorithm is also very low.

3.6 Sample Images

3.6.1 Sample Dataset:

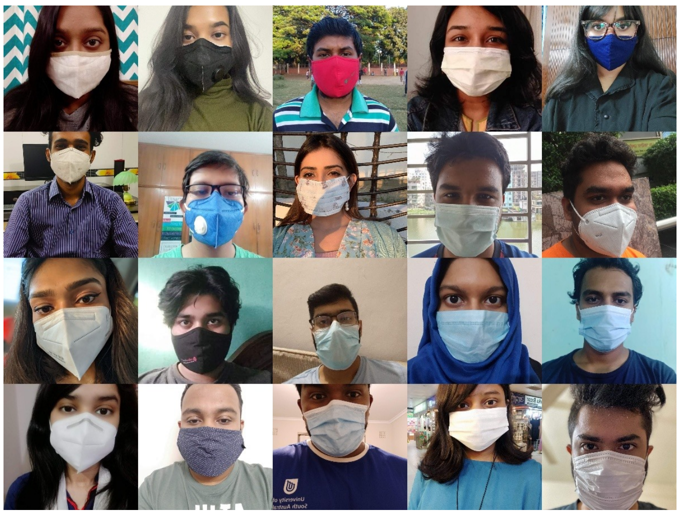


Figure 3.8: Sample dataset for correctly worn masked.

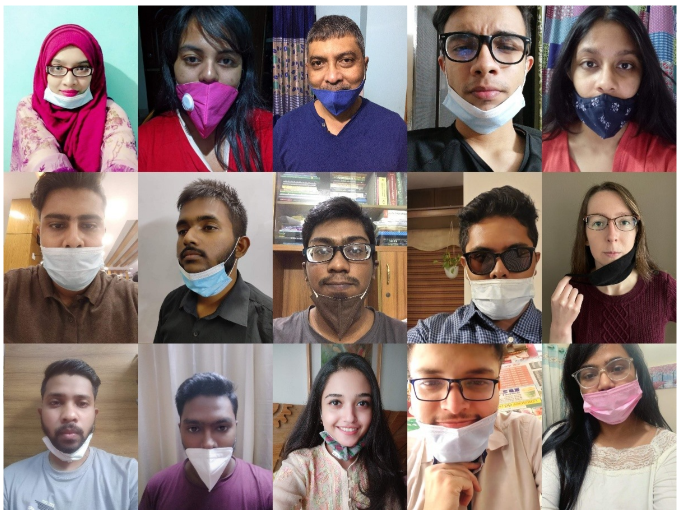


Figure 3.9: Sample dataset for incorrectly worn masked.

3.6.2 Sample Dataset after Preprocessing:

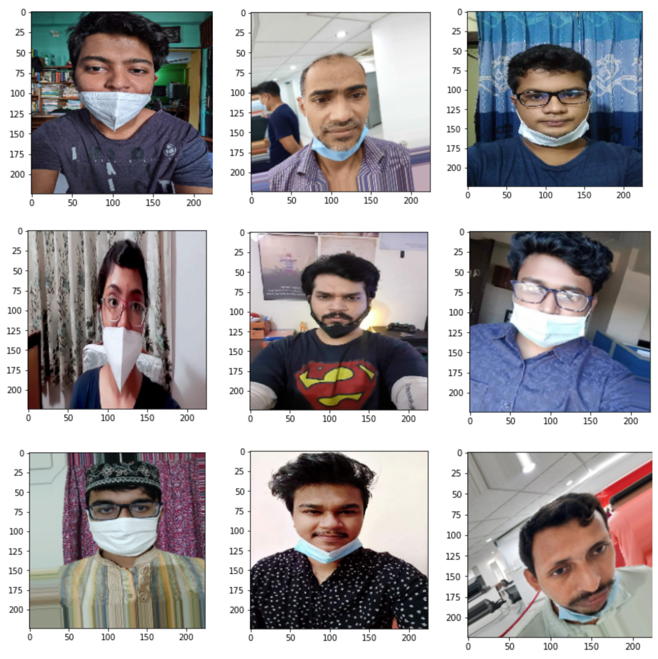


Figure 3.10: Sample dataset after preprocessing.

Chapter 4

Algorithms

Deep learning algorithms are used to learn and extract features from the training set and make accurate classifications on the test set, which is the unseen data. We are also using transfer learning for shortening the algorithm training time and the limited amount of data that we have. In recent years, many deep learning algorithms and architectures have been developed, thanks to the boom of research in the field of deep learning and image classification. We have evaluated the performances of 3 architectures on our data-set, ResNet50, MobilenetV2 and InceptionV3. We tried to compare the performance of these 3 architectures, with SVM and Softmax as the last layer. A detailed description of the architectures that we have used are given below.

4.1 Convolutional Neural Networks:

Convolutional neural networks or CNN are among the most popular types of deep learning algorithms used for pattern recognition and image classification. Many modern architectures are built on the CNN algorithm. According to Keiron O'Shea, Ryan Nash , CNNs are made up of neurons that learn to optimise themselves where each neuron continues to receive feedback and perform a task [39]. The CNN is made up of neurons that are organized in three dimensions: the input's spatial dimensionality (height and width) and depth. A CNN architecture is made by stacking three layers, which are convolutional layer, pooling layer and fully connected layers.

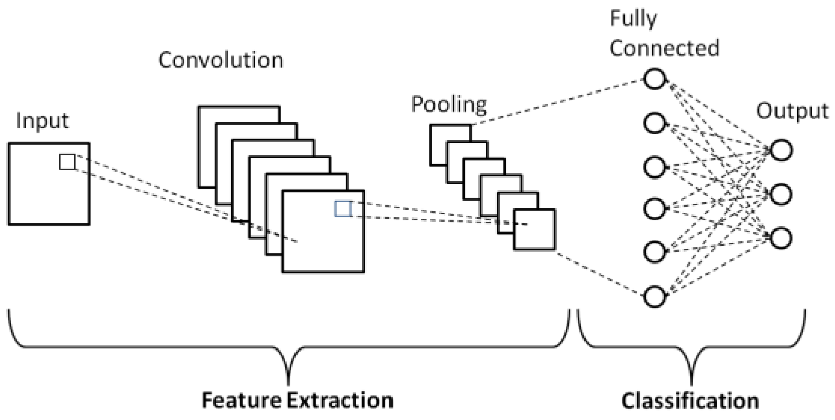


Figure 4.1: Convulational Neural Network Architecture

The output of neurons connected to local regions of the input is determined by the convolutional layer, which will calculate the scalar product between their weights and the region connected to the input volume. The input, which is an image converted to vector, is passed over a kernel or filter, which is a two-dimensional array of weights. A dot product is performed with the input data and kernel. After the kernel is systematically applied throughout the image, a two-dimensional array is generated which is called the feature map. Activation functions like Relu are used to "excite" the kernel when a specific feature occurs at a given spatial position of the input. After passing through the convolutional layer, the activation map is passed through the pooling layer. The purpose of pooling is to decrease the dimensionality and complexity of the architecture. Fully connected layers contain neurons directly connected in the two neighbouring layers without being connected to any layers beyond them.

4.2 MobileNet:

MobileNet is a lightweight model architecture which has fewer parameters and higher classification accuracy [40]. Mobilenet uses a depth-wise separable convolutional network to deepen the network and reduce parameters and computation. It is a simplified architecture that constructs lightweight deep convolutional neural networks using depthwise separable convolutions and provides an efficient model for mobile and embedded vision applications.

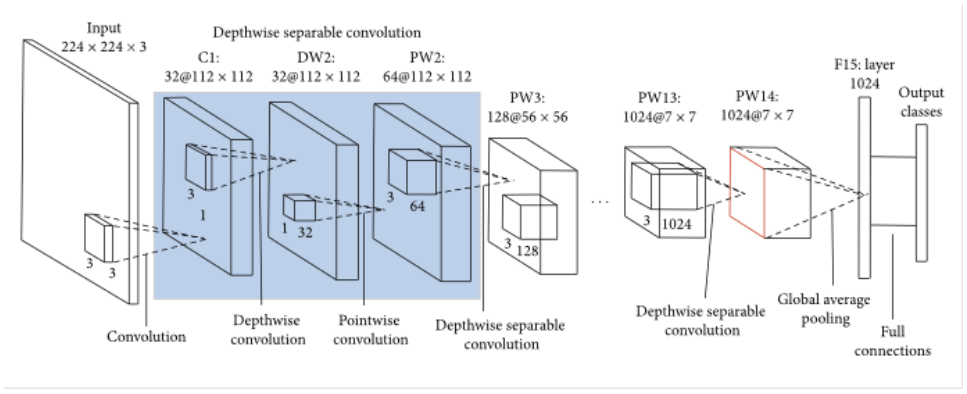


Figure 4.2: MobileNet Architecture.

Depth-wise separable convolution filters combine depth-wise and point convolution filters to produce depth-wise separable convolution filters. The point convolution filter combines the performance of depth-wise convolution linearly with 1*1 convolutions, while the depth-wise convolution filter performs a single convolution on each input channel.

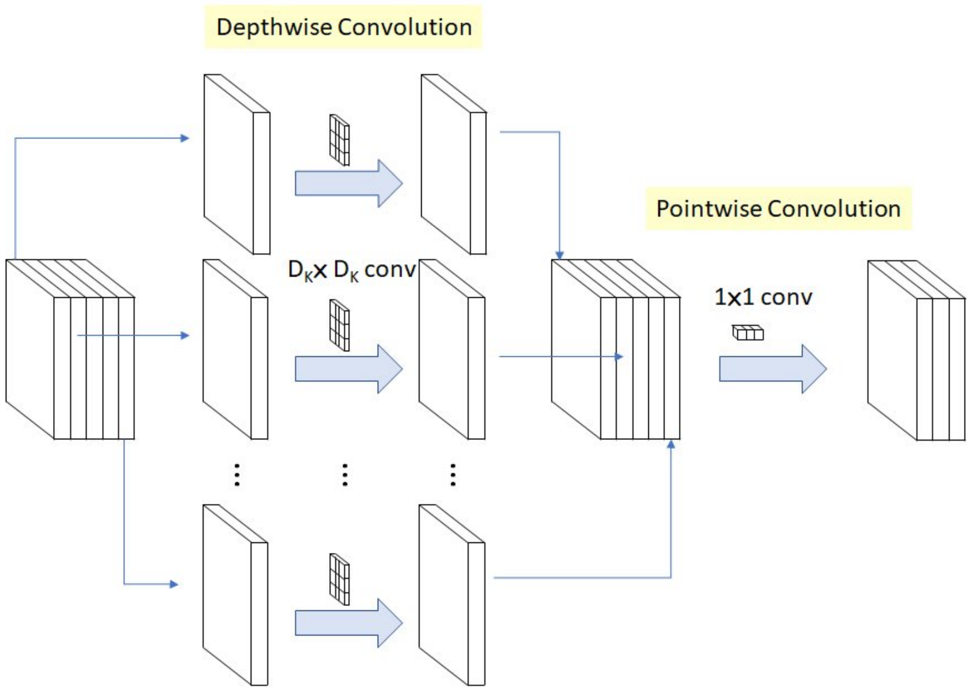


Figure 4.3: Standard convolutional layers with batch normalization and Relu.

The key difference between MobileNet and conventional CNN architecture is that instead of a single 3x3 convolution layer, the batch normalisation, and ReLU are used. As shown in the figure, Mobile Nets break the convolution into a 3x3 depth-wise convolution and a 1x1 point-wise convolution.

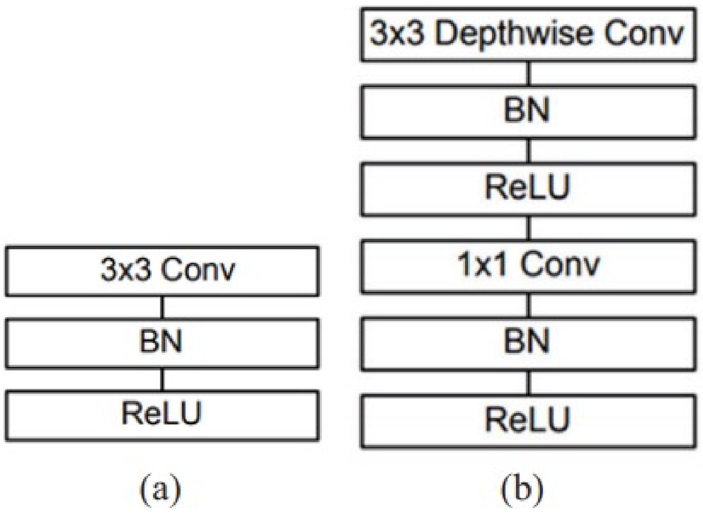


Figure 4.4: Depthwise convolutional layer with combination of depth-wise and point-wise layers followed by batch normalization and Relu.

Computational cost of standard convulation is:

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F \tag{4.1}$$

Where M is the number of input channels, N is the number of output channels, D_KxD_K is the dimension of kernels and D_FxD_F is the dimension of feature map.

Computational cost of depth-wise separable convolution is:

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F \tag{4.2}$$

Hence, the reduction in computational cost due to using mobilenet instead of CNN is:

$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D_K^2} \tag{4.3}$$

MobileNet consists of three depthwise separable convolutions, which require 8 to 9 times less computation than regular convolutions, resulting in only a minor loss of accuracy.
MobileNetV2 is a better version of MobileNetV1 which has inverted residual structure and removed non linearity and narrow layers. This model accepts a low-dimensional compressed representation as input, which is then extended to high dimension before being filtered with a lightweight depthwise convolution. After that, using a linear convolution, the features are projected back to a low-dimensional representation. MobileNetV2 contains two types of blocks, one with stride 1 and the other with stride 2. Both blocks contain 3 layers. The first layer contains 1x1 convolution with RelU6 whereas depthwise convolution takes place in the second layer. In third layer, another 1x1 convolution takes place but the difference between MobileNetV1 and MobileNetV2 in this layer is no non-linearity is imposed in the

third layer in MobileNetV2. This means RelU6 has been removed from the third layer because it has been seen that the deep networks only have the power of a linear classifier if RelU6 is applied in the third layer.

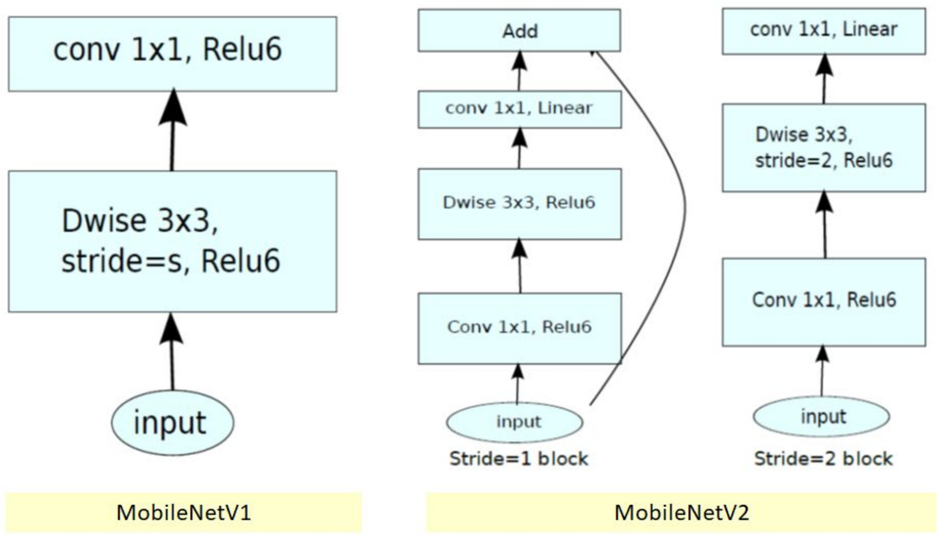


Figure 4.5: Difference between MobileNetv1 and MobileNetv2.

4.3 Inception-v3

At ILSRVRC 2014, the state-of-the-art architecture was Inception V1 (GoogleNet) [41]. The purpose of developing this architecture was to increase the use of the network's computing resources. This was accomplished by increasing the network's width (the total number of units at any given level) and depth (the total number of levels) while keeping the overall cost constant. Version 1 had 27 layers and filter sizes of 1x1, 3x3, and 5x5. It has created the lowest error on the ImageNet classification dataset, but there are some areas where the model can be improved to increase accuracy and reduce complexity.
Convolutions such as 5x5 have been used by Inception V1 to reduce the input measurements by a wide margin. The neural network's accuracy suffers as a result of this because if input dimension is reduced too significantly, the neural network is vulnerable to information loss. Due to the usage of bigger convolutions, there was a drop in computational complexity of the model. These problems were solved in Inception V2 by factorizing the 5x5 size into two 3x3 convolution operation which is 2.78 times less expensive than using a 5x5 convolution [42]. Hence, the performance of the architecture was improved. With the introduction of factorization, the number of parameters is reduced and the chance of overfitting is decreased, allowing the network to go deeper. Finally, the version we used, v3 which is 42 layers deep and includes all of the updates from v2 as well as a few new features. One of the most important is factorization of the 7x7 convolution, also known as label smoothing, which is a form of regularizing component that prevents overfitting [43].

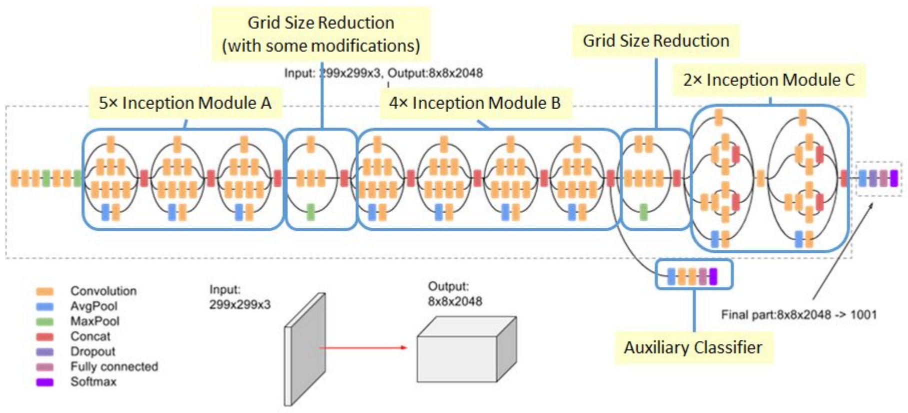


Figure 4.6: Inception v3 Architecture.

4.4 ResNet Architecture

Deeper neural networks are hard to train. As the depth of the network increases, after a certain period of time, the training accuracy gets saturated or sometimes starts to degrade because of the vanishing gradient problem. To solve this problem, residual networks (ResNet) are used [44]. The building block of a ResNet architecture is a residual block which uses ‘skip-connections’. The skip connection is the core of a residual block. We have seen that after a certain threshold, the deeper layers of the neural networks perform poorly. In ResNet, the connection of the shallower network skips one or two training layers and is added directly with the activation function so that the accuracy of the shallower network does not get degraded. In a normal neural network, the input ‘x’ gets multiplied with the weight and gets added with bias in every layer. Then, to add non-linearity, the function is passed through an activation function.

$$\begin{gathered} H(x) = f(wx + b) \\ \text{or } H(x) = f(x) \\ H(x) = f(x) + w1.x \end{gathered} \tag{4.4}$$

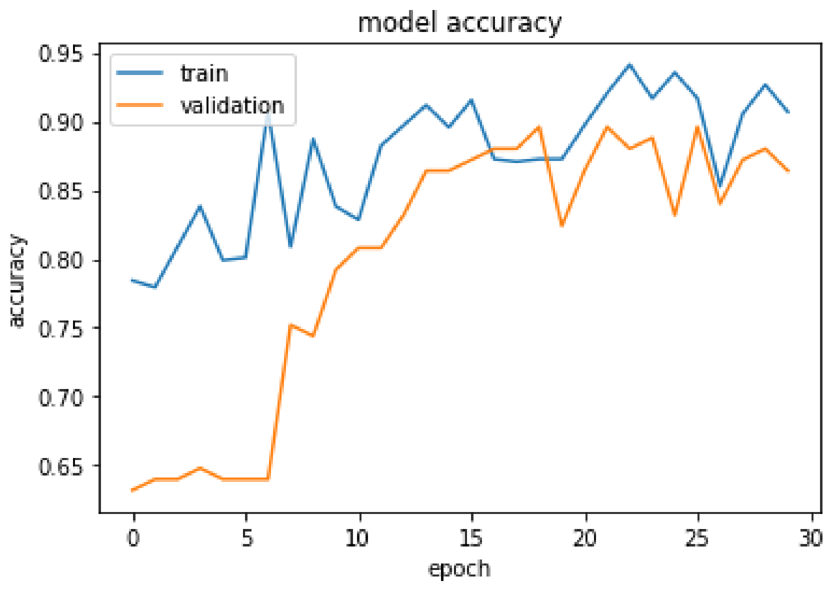
After the addition of the residual block, the input ‘x’ gets a direct connection to the activation function without getting multiplied with the weights and biases of the training layers.
Residual layers help to learn the identity function. In ResNet50 architecture, 3 convolution layers (1*1, 3*3 and 1*1) are used instead of 2 layers.

Chapter 5

Results and Accuracy

5.1 ResNet50

The first algorithm we decided to choose to run our data-set was with Resnet50. We expected this algorithm to perform the best with our data-set, therefore we decided to run this algorithm with our own data-set which consisted of only data we collected from Bangladesh as well as the other combined data-set which included data from all around the world. We also ran the model with both the data-sets through a softmax classifier and then compared it with an SVM classifier to have a better understanding of the performance of the model and the classifier.



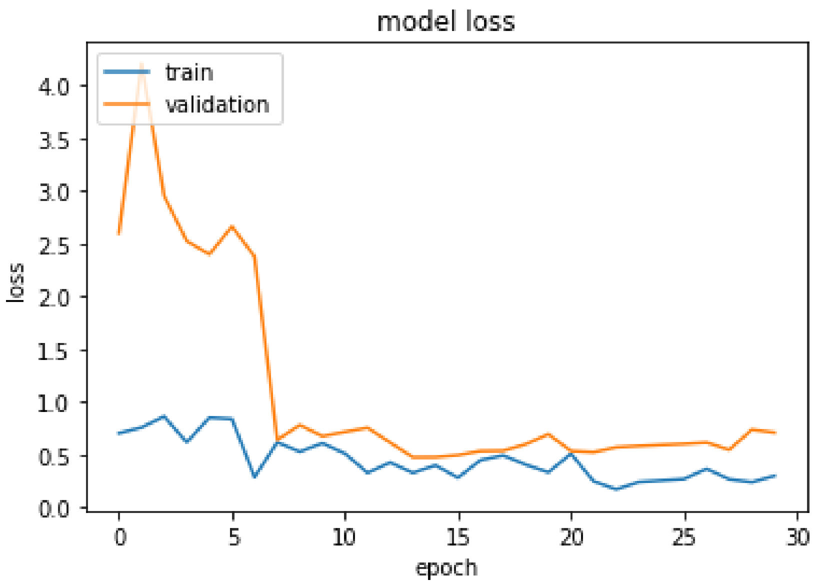


Figure 5.1: Train Validation for local dataset with softmax

In 5.1 the data shown is using the data-set we collected locally. The classifier used was softmax. We can see that the validation accuracy is roughly 86%, and the training accuracy is roughly 92%. As for the validation loss, it is roughly 0.7 while the training loss is slightly lower at 0.5.

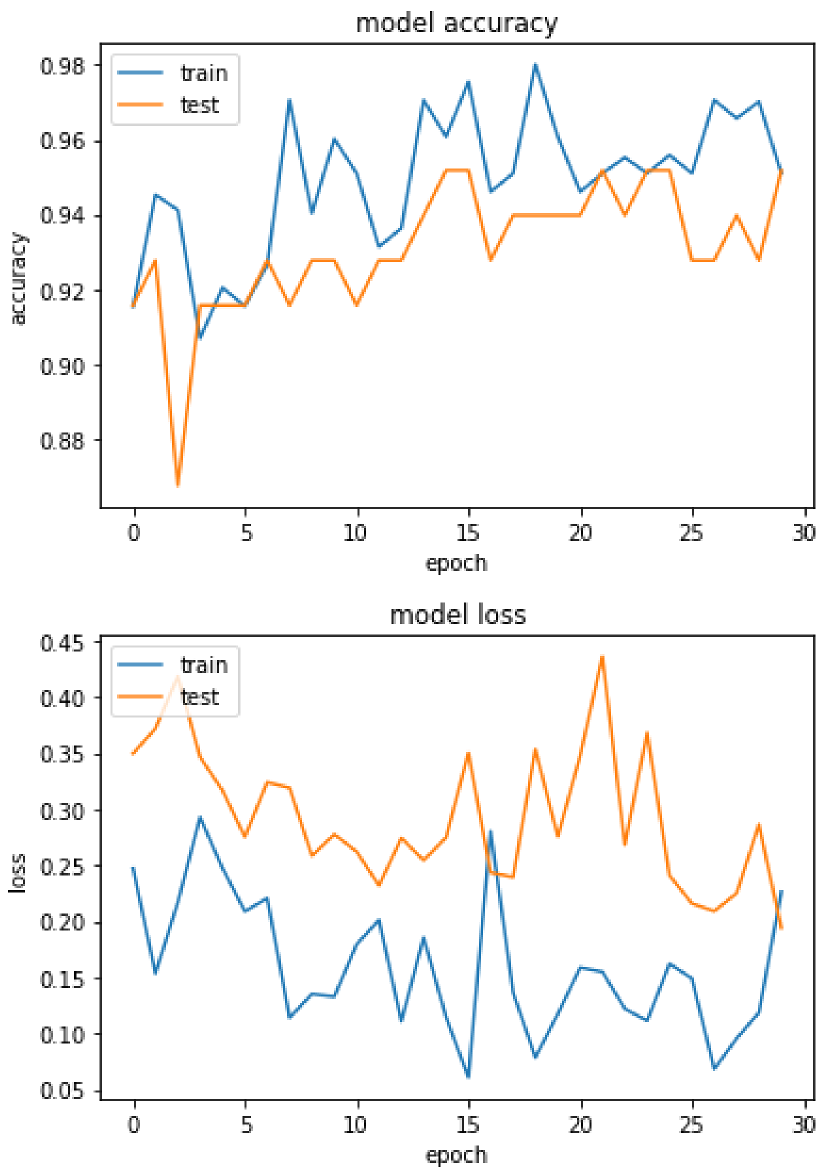
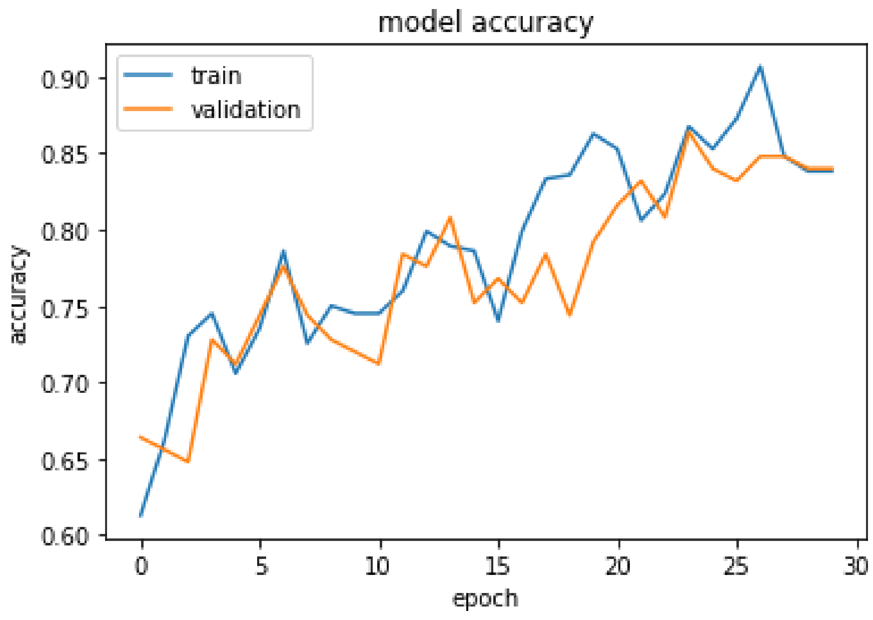


Figure 5.2: Train Test for local dataset with softmax

In 5.2 the data shown is using the data-set we collected locally. The classifier used was softmax. We can see that the training accuracy is roughly 96%, and the test accuracy is roughly 95%. As for the test loss, it is roughly .24 while the training loss is slightly lower at .20.



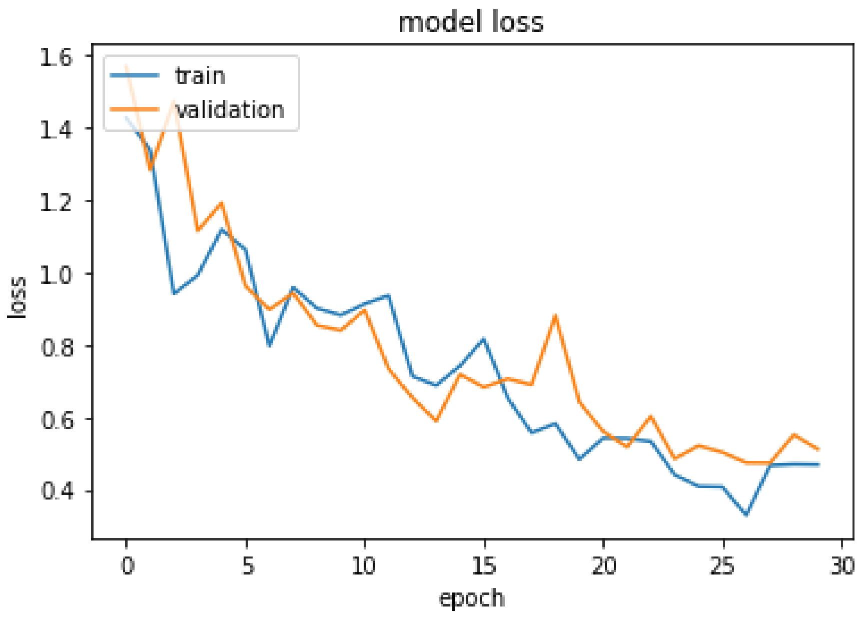


Figure 5.3: Train validation for the local dataset with SVM

In 5.3 the data shown is using the data-set we collected locally. The classifier used was SVM. We can see that the validation accuracy is roughly 85%, and the training accuracy is roughly 85%. As for the validation loss, it is roughly 0.6 while the training loss is slightly lower at 0.6.

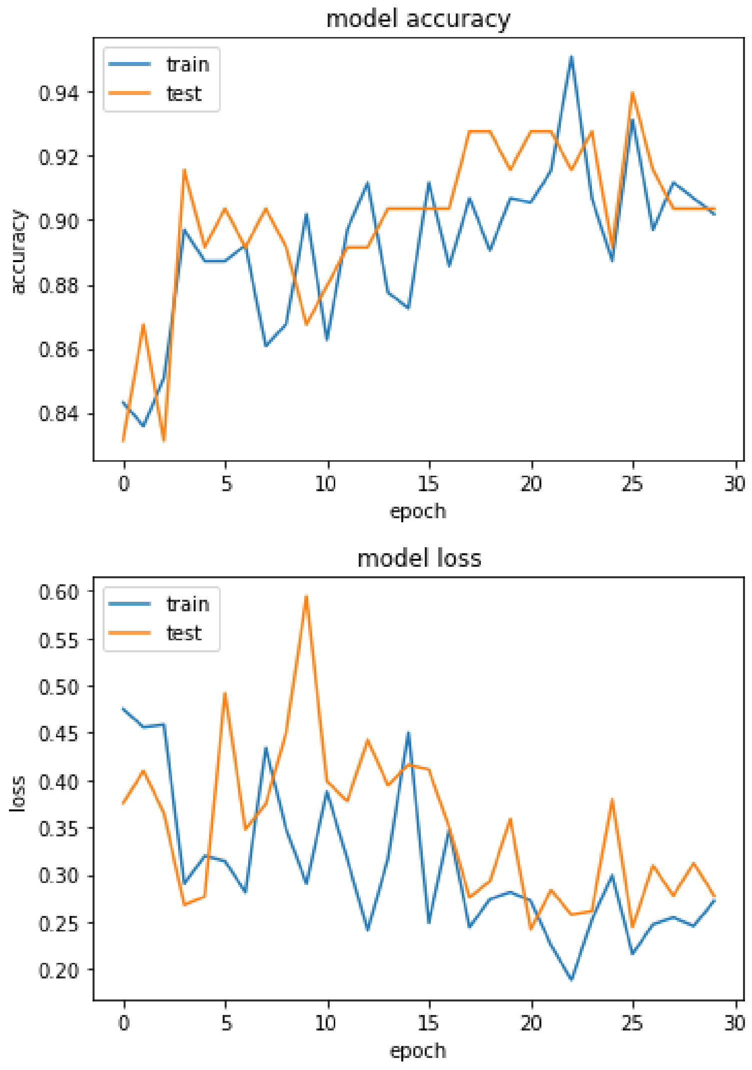
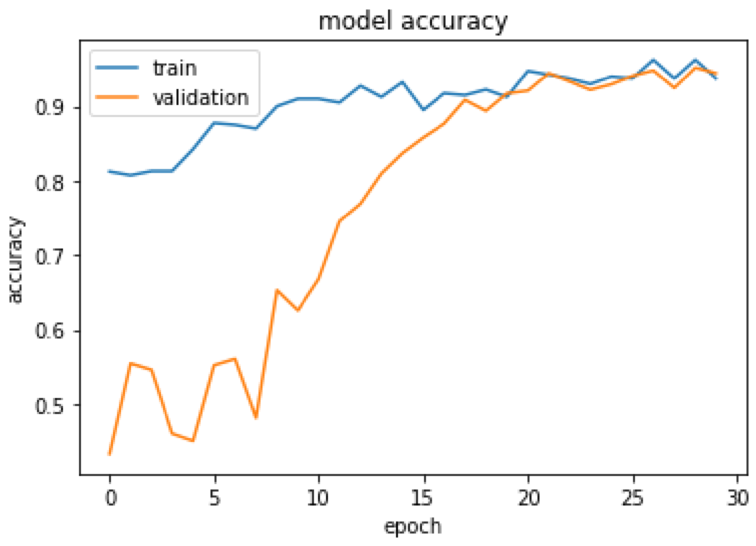


Figure 5.4: Train Test for the local dataset with SVM

In 5.4 the data shown is using the data-set we collected locally. The classifier used was SVM. We can see that the test accuracy is roughly 91%, and the training accuracy is roughly 91%. As for the test loss, it is roughly 0.3 while the training loss is similar at 0.3.



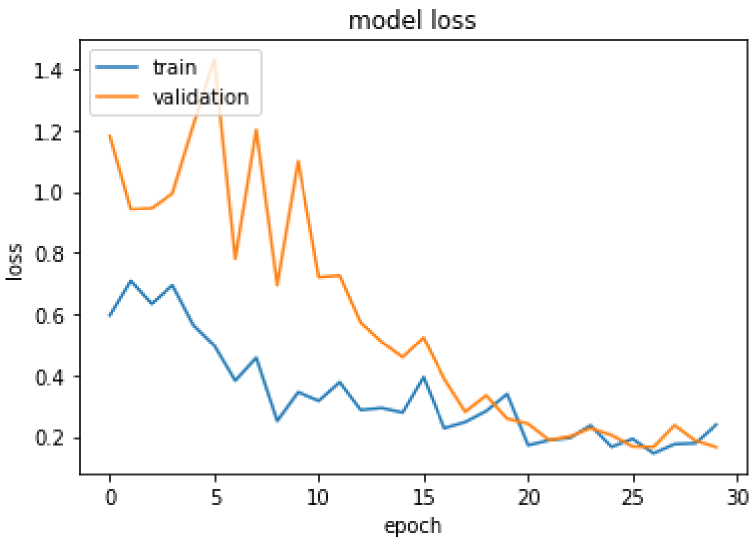


Figure 5.5: Train validation for the dataset with softmax

In 5.5 shown is using the combined data-set we collected from both Bangladesh and abroad. The classifier used was softmax. In fig 4.1.3 We can see that the training accuracy is roughly 90%, and the validation accuracy is also similar at roughly 90%. As for the test loss, it is roughly 0.3 while the training loss is also similar at 0.3.

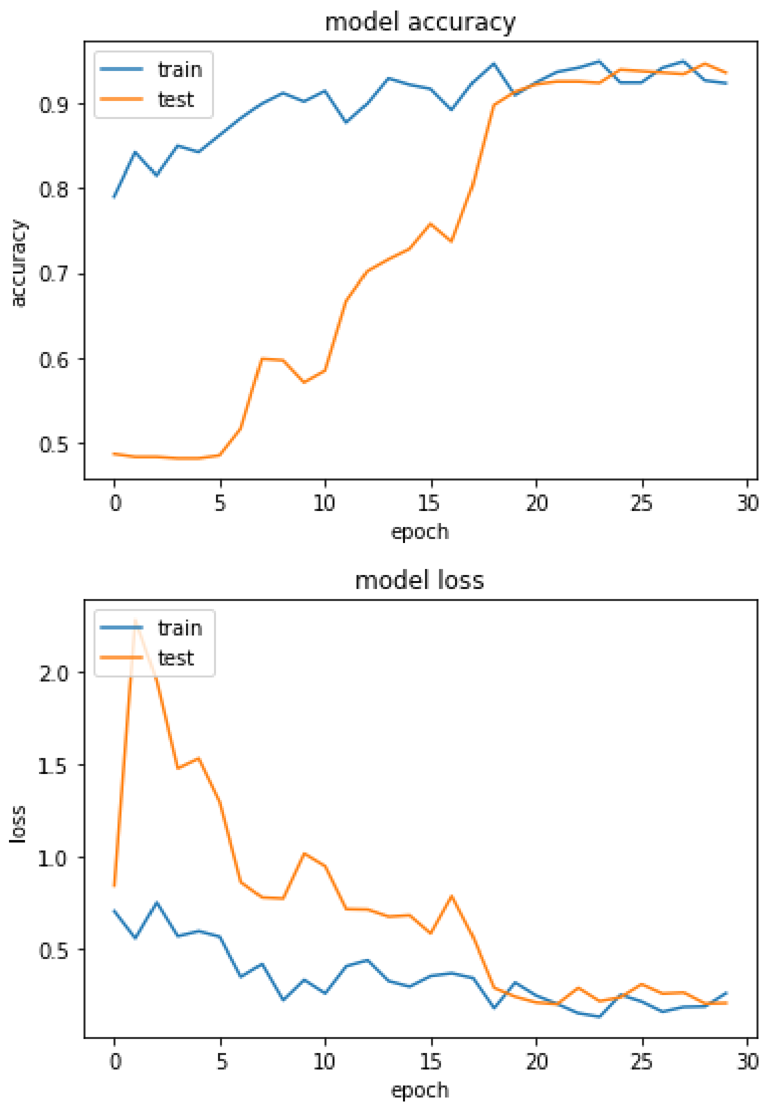


Figure 5.6: Train test for the dataset with softmax

In 5.6 We can see that the training accuracy is roughly 91%, and the test accuracy is also similar at roughly 90%. As for the test loss, it is roughly 0.30 while the training loss is also similar at 0.30.

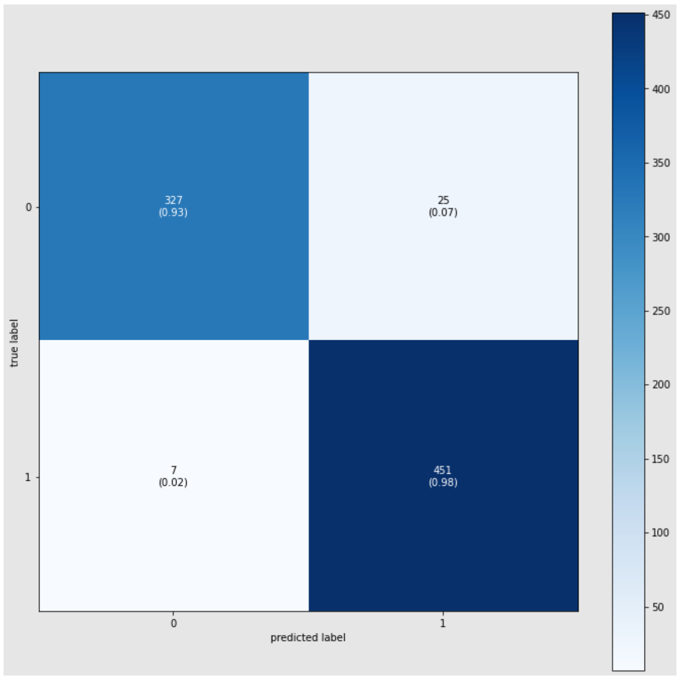


Figure 5.7: Confusion Matrix for Train Validation for the dataset with softmax

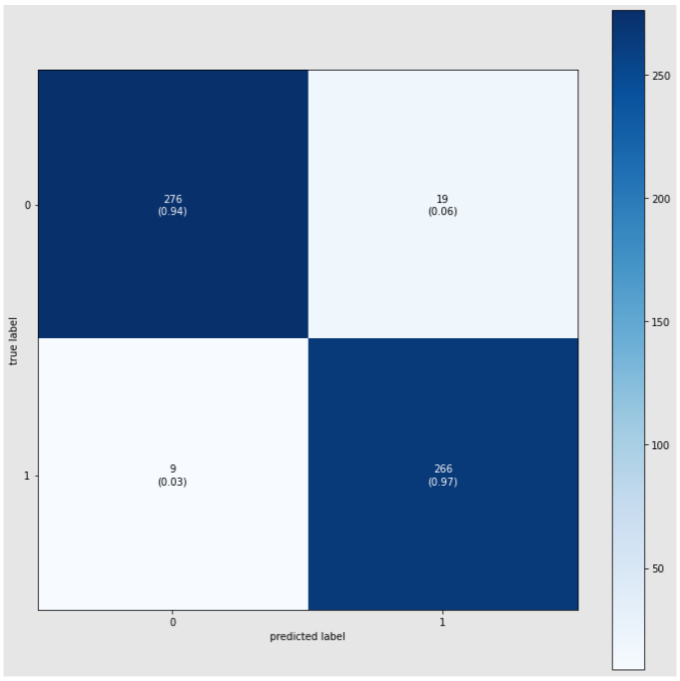


Figure 5.8: Confusion Matrix for Train Test for the dataset with softmax

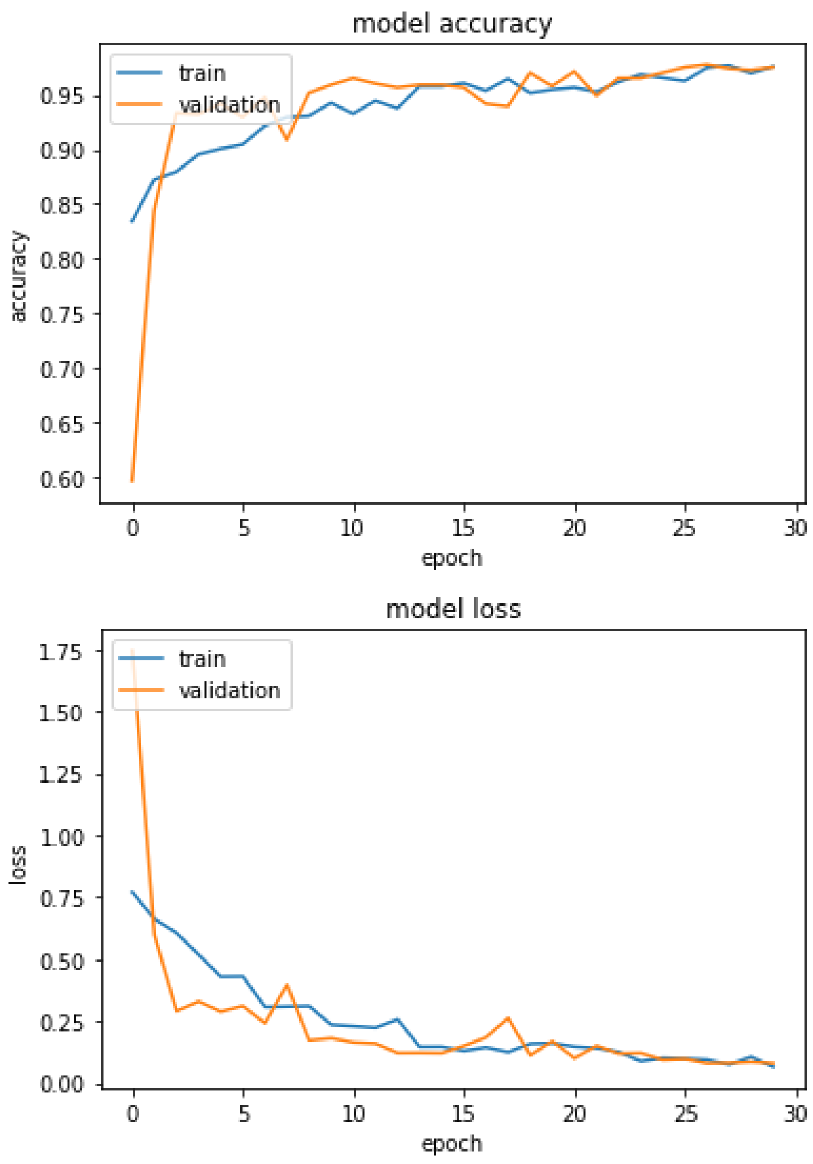


Figure 5.9: Train validation for the dataset with SVM

In 5.9 and 5.10 the data shown is using the combined data-set we collected from both Bangladesh and abroad. The classifier used was SVM. In 5.9 we can see that the training accuracy is roughly 95%, and the validation accuracy is also similar at roughly 95%. As for the test loss, it is roughly 0.10 while the training loss is also similar at 0.10.

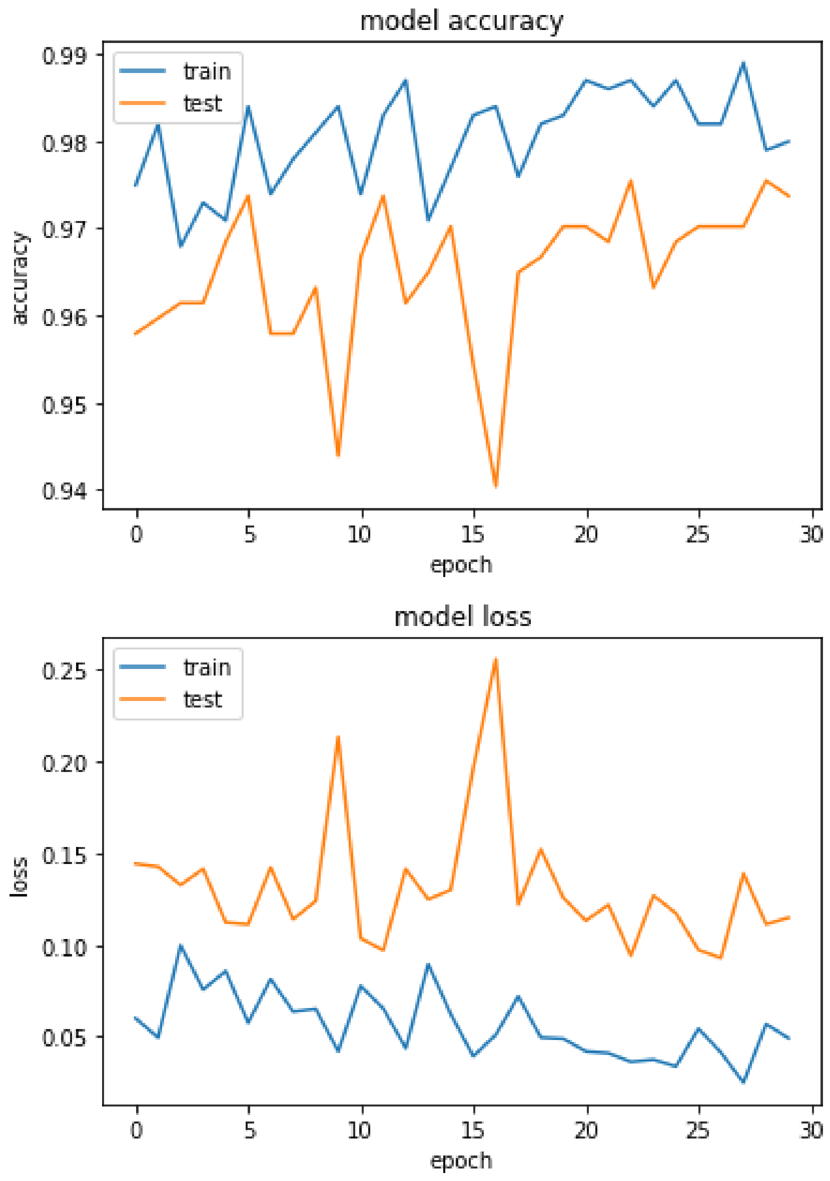


Figure 5.10: Train test for the dataset with SVM

In 5.10 We can see that the training accuracy is roughly above 98%, and the validation accuracy is also similar at roughly 97%. As for the test loss, it is roughly 0.15 while the training loss is also similar at 0.05.

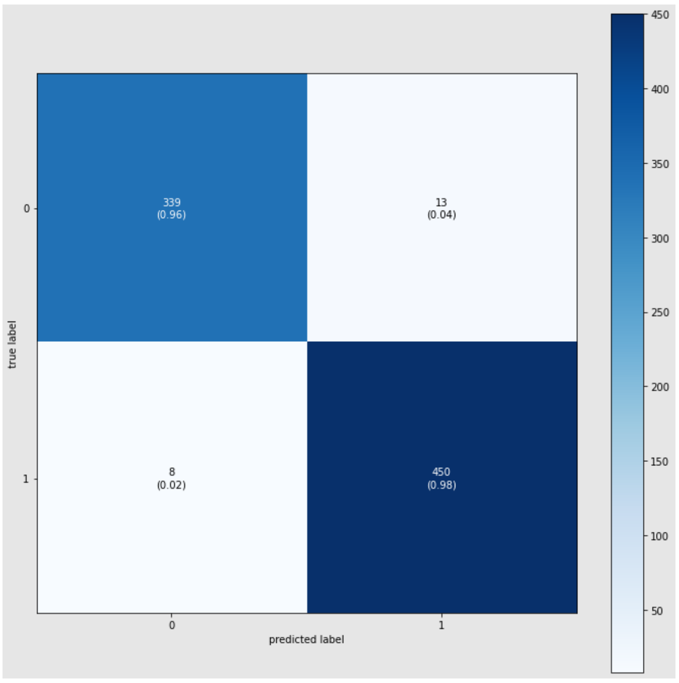


Figure 5.11: Confusion Matrix for Train Validation for the dataset with SVM

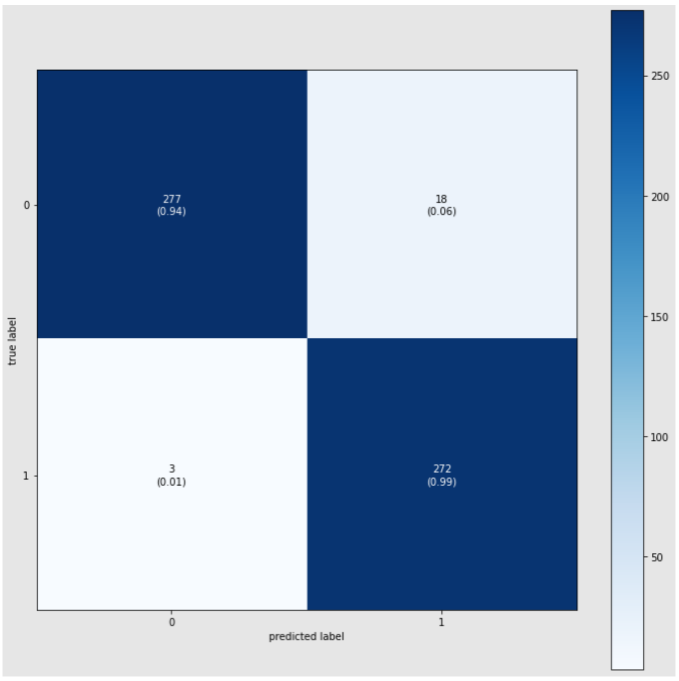


Figure 5.12: Confusion Matrix for Train Test for the dataset with SVM

From all the graphs plotted above (5.1- 5.12) we can see all of them maintain a similar shape, therefore overfitting or underfitting does not take place. We can assume the algorithm is able to properly distinguish between properly masked and improperly masked individuals with very little error. The highest accuracy we obtained was by using the combined data-set of which images were collected from both abroad and locally, and with an SVM classifier in place of softmax. Here the accuracy was an astounding 98% for the test data-set, as well as a very low loss of 0.05.

5.2 Inception v3

The next algorithm we decided to choose to run our data-set was with Inception v3. We expected this algorithm to perform similar to the results we obtained through Resnet50 and wanted to compare the results. We ran this algorithm with the combined data-set which included data from all around the world. We also ran the model through a softmax classifier and then compared it with an SVM classifier again to have a better understanding of the performance of the model and the classifier. In 5.13 and 5.14 we are using SVM as classifier, and in 5.17 and 5.18 we are using Softmax as classifier

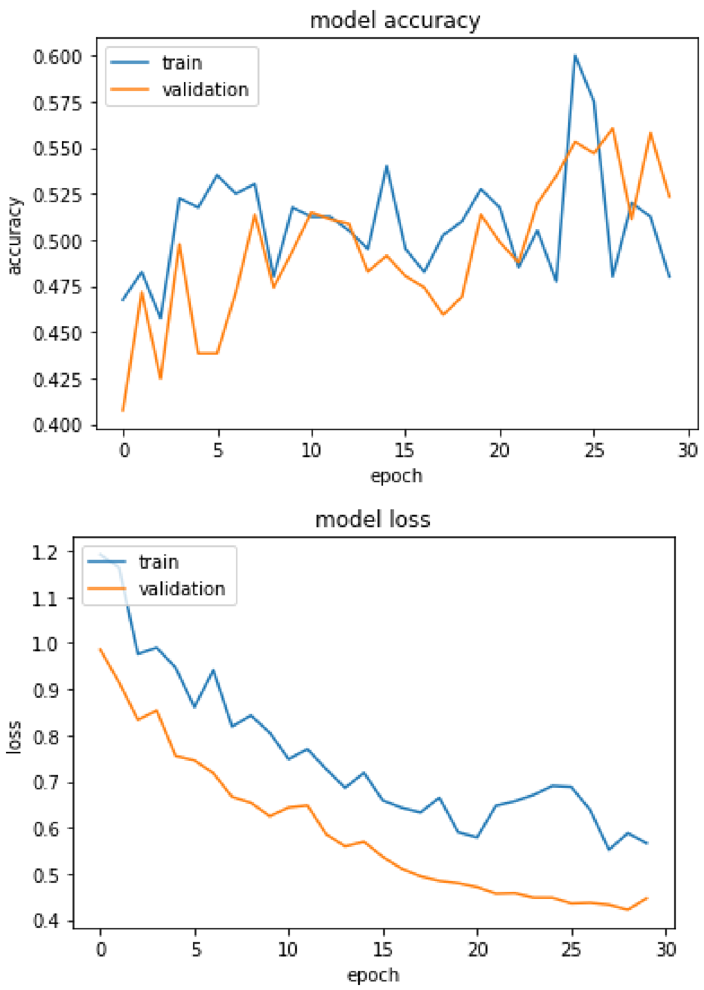
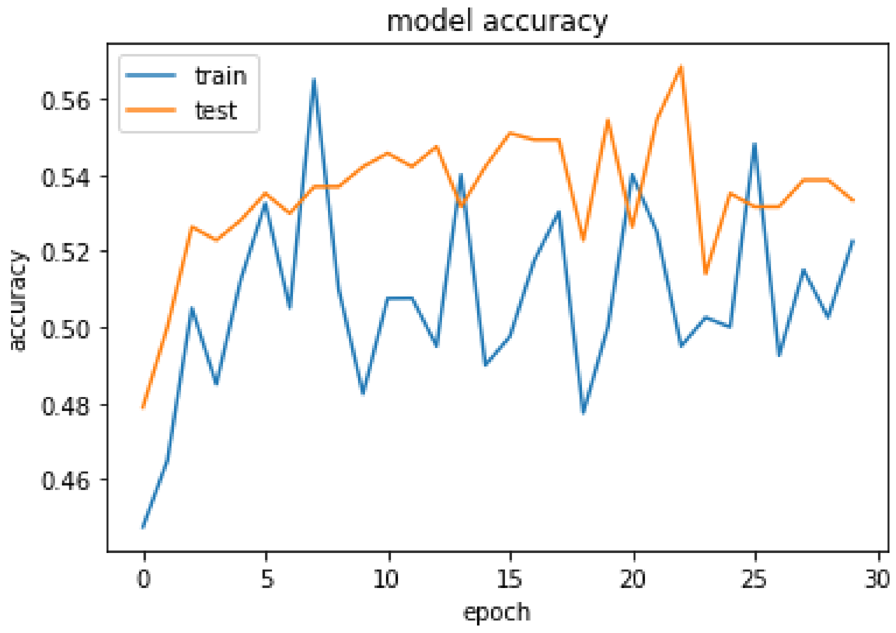


Figure 5.13: Train Validation for the dataset with svm

In 5.13 We can see that the training accuracy is roughly 50%, and the validation accuracy is also similar at roughly 53%. As for the test loss, it is roughly 0.6 while the training loss is also similar at 0.45.



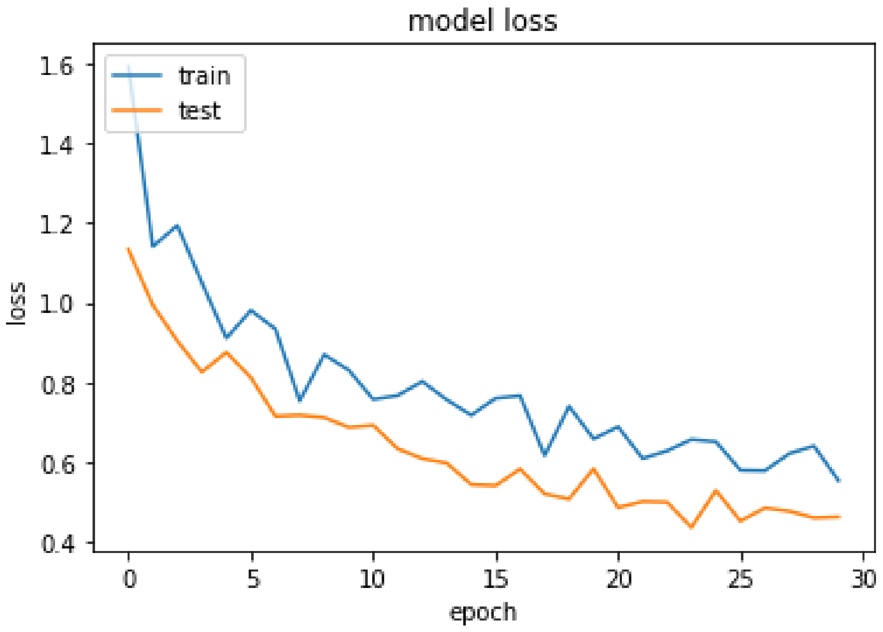


Figure 5.14: Train Test for the dataset with svm

In 5.14 We can see that the training accuracy is roughly 52%, and the test accuracy is also similar at roughly 53%. As for the test loss, it is roughly 0.5 while the training loss is also similar at 0.6.

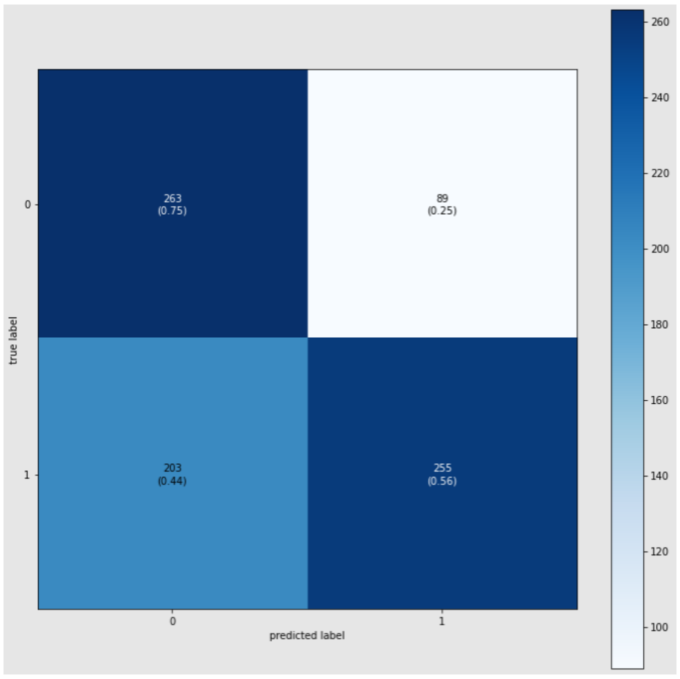


Figure 5.15: Confusion Matrix for Train Validation for the dataset with SVM

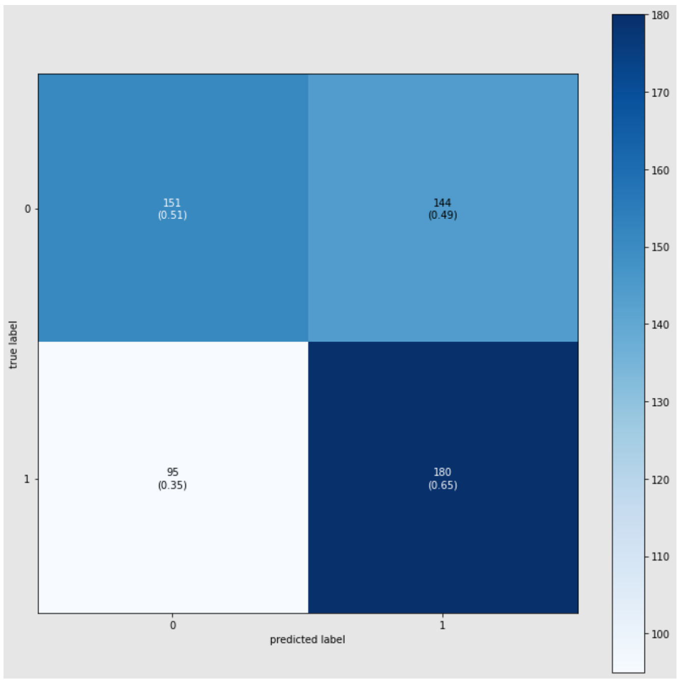
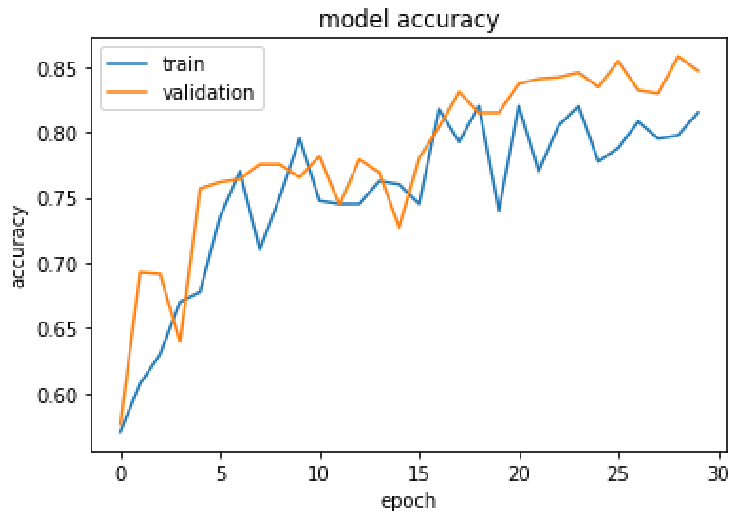


Figure 5.16: Confusion Matrix for Train Test for the dataset with SVM



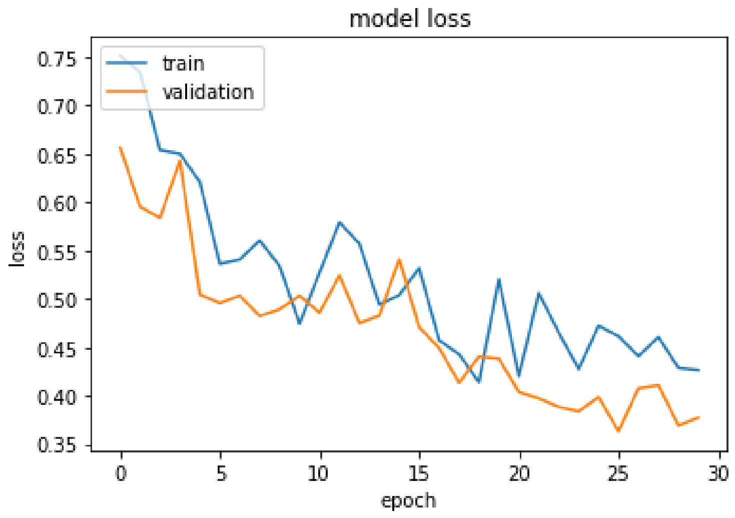
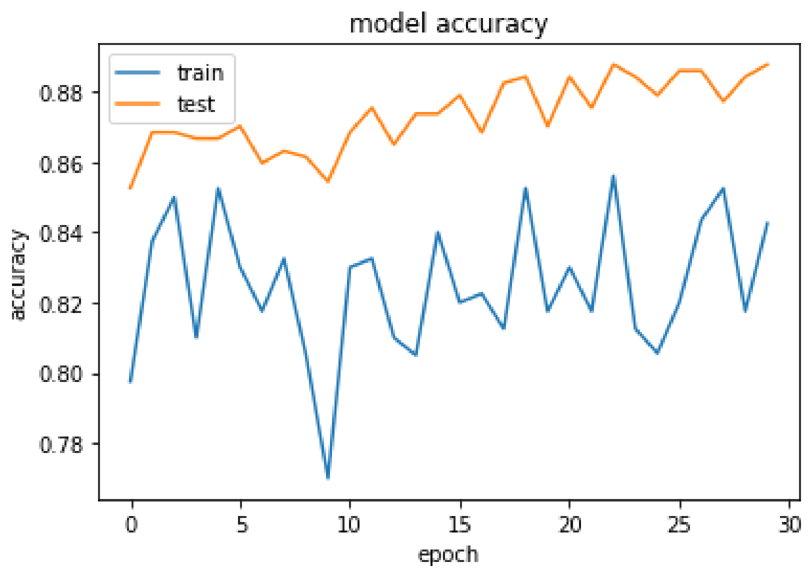


Figure 5.17: Train Validation for the dataset with softmax

In 5.17 We can see that the training accuracy is roughly 80%, and the validation accuracy is also similar at roughly 85%. As for the validation loss, it is roughly 0.4 while the training loss is also similar at 0.45.



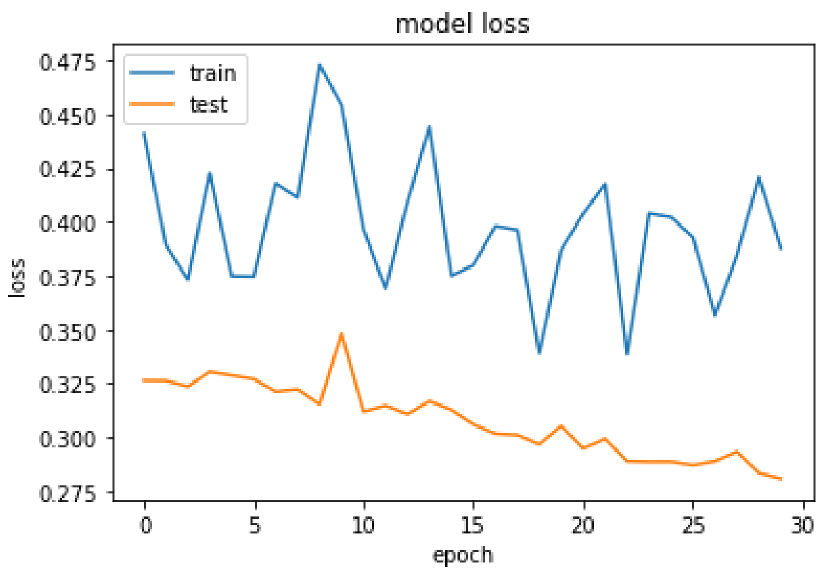


Figure 5.18: Train Test for the dataset with softmax.

In 5.18 We can see that the training accuracy is roughly 84%, and the test accuracy is also similar at roughly 88%. As for the test loss, it is roughly 0.3 while the training loss is also similar at 0.375.

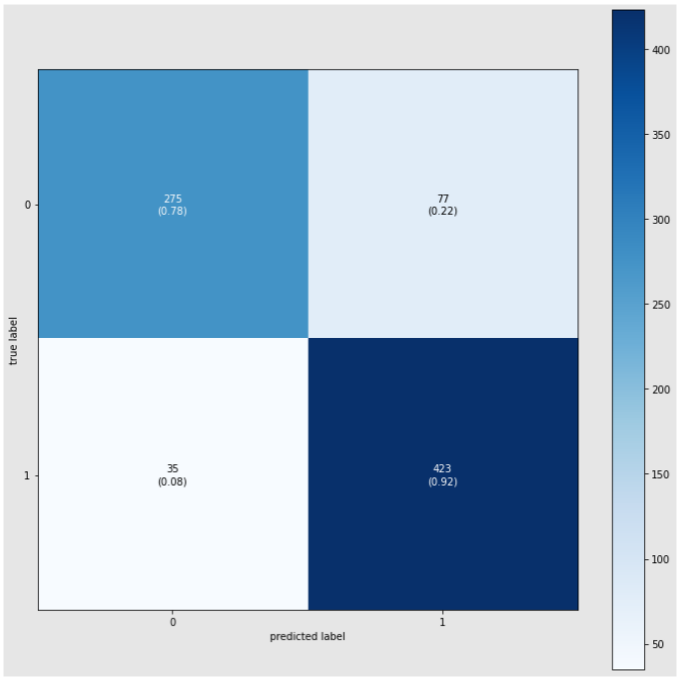


Figure 5.19: Confusion Matrix for Train Validation for the dataset with softmax

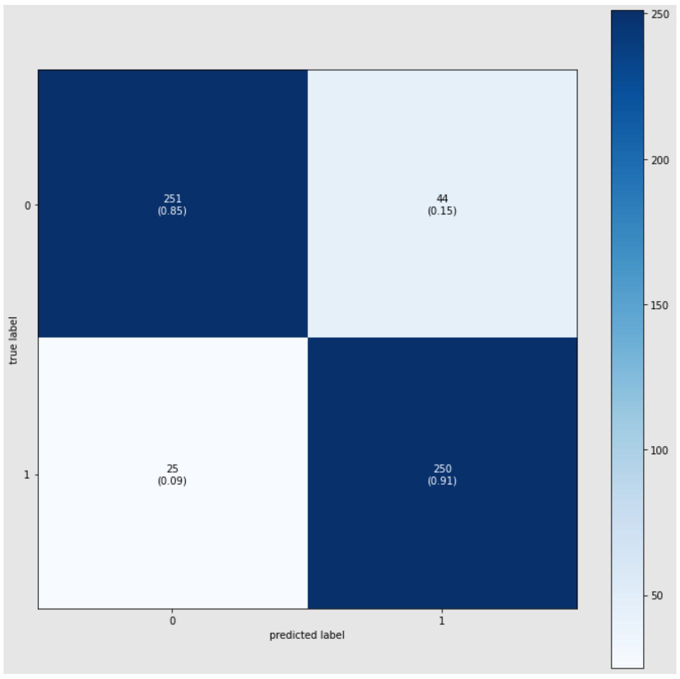


Figure 5.20: Confusion Matrix for Train Test for the dataset with softmax

From the Inceptionv3 dataset, we can deduce that our dataset works better with softmax classifier rather than SVM where it performed significantly better in terms of both accuracy and loss. The training and validation dataset for example had a difference of nearly 30%. It had 50% training accuracy with SVM whereas softmax had an 80% training accuracy. Unlike Resnet50 where the SVM classifier slightly outperformed Softmax, while using Inception v3, Softmax overpowers SVM by a huge margin.

5.3 MobileNet v2

The final algorithm we decided to run our dataset with was with Mobilenetv2. We again ran this algorithm with the combined dataset which included data from all around the world while using the SVM and Softmax classifiers one at a time. In 5.21 and 5.22 we are using Softmax as classifier, and in 5.25 and 5.26 we are using SVM as classfier.

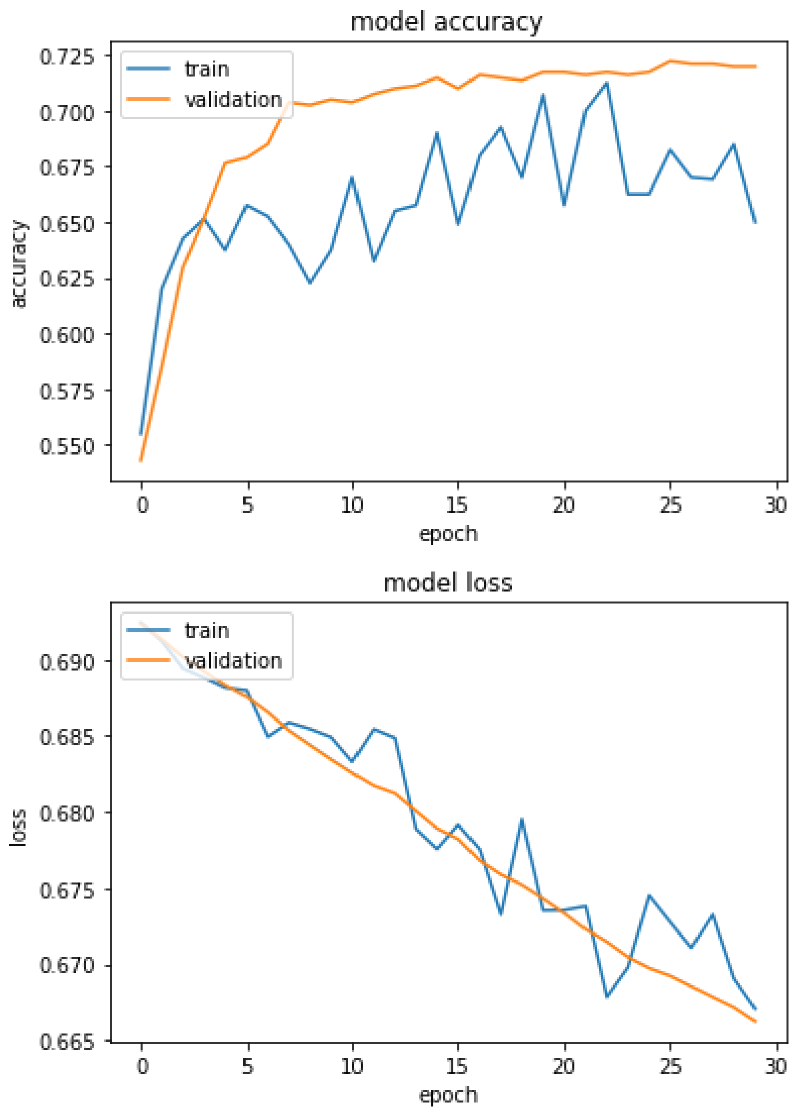
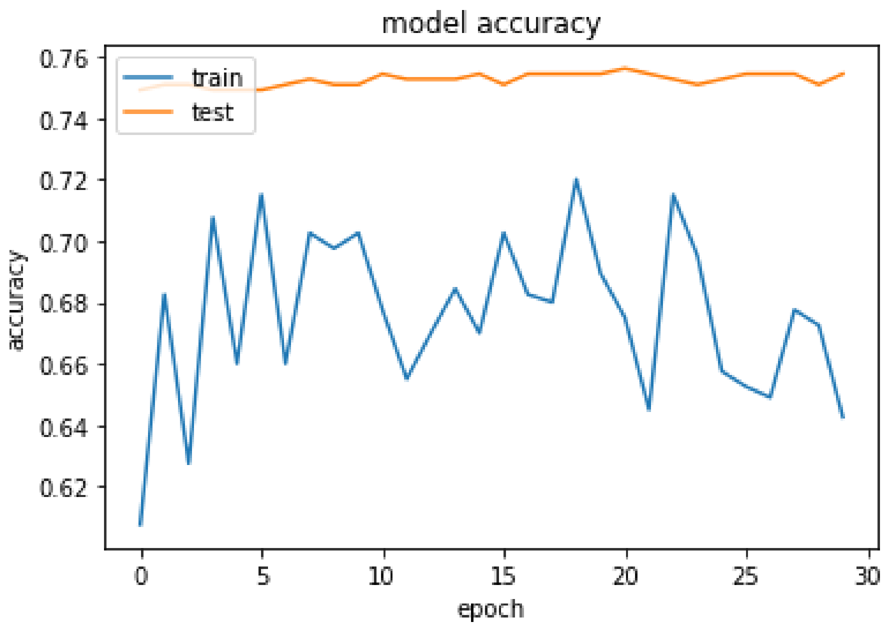


Figure 5.21: Train Validation for the dataset with softmax.

In 5.21 We can see that the validation accuracy is roughly 71%, and the training accuracy is at roughly 65%. As for the validation loss, it is roughly 0.665 while the training loss is also similar at 0.665.



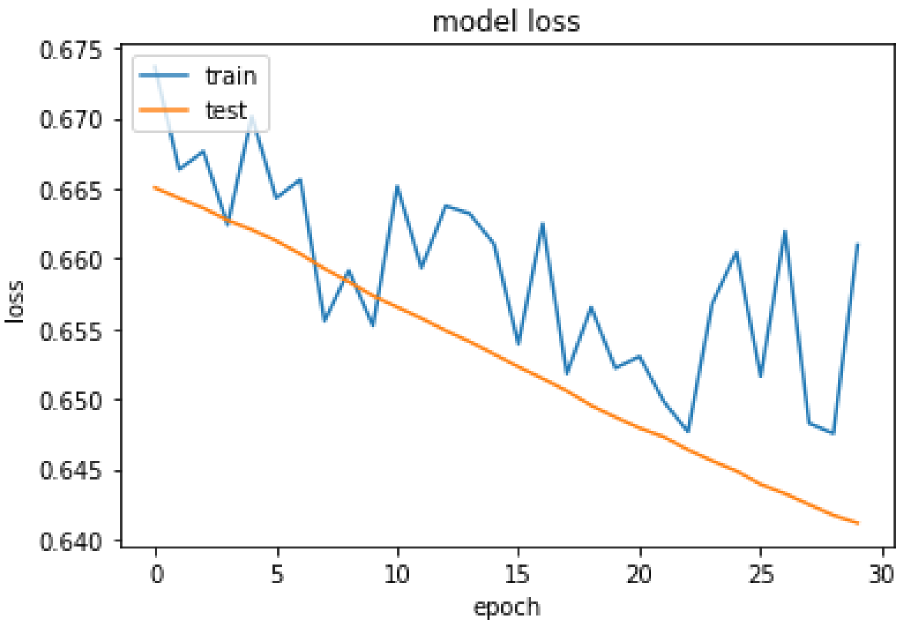


Figure 5.22: Train Test for the dataset with softmax

In fig 5.22 We can see that the test accuracy is roughly 75%, and the training accuracy is at roughly 66%. As for the test loss, it is roughly 0.64 while the training loss is fluctuating at 0.66.

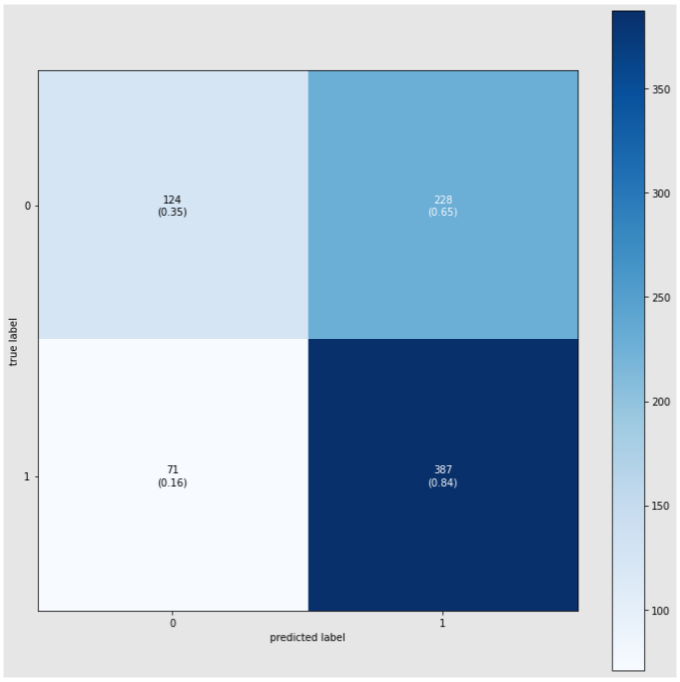


Figure 5.23: Confusion Matrix for Train Validation for the dataset with softmax

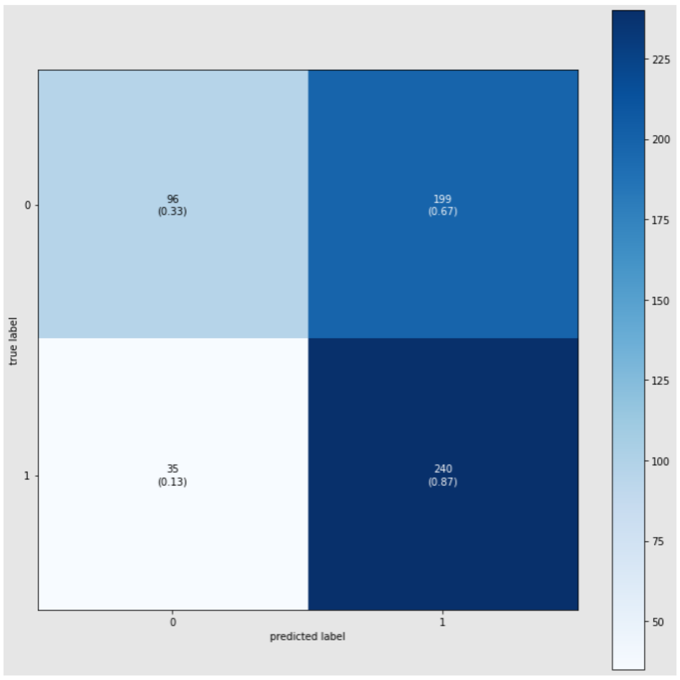
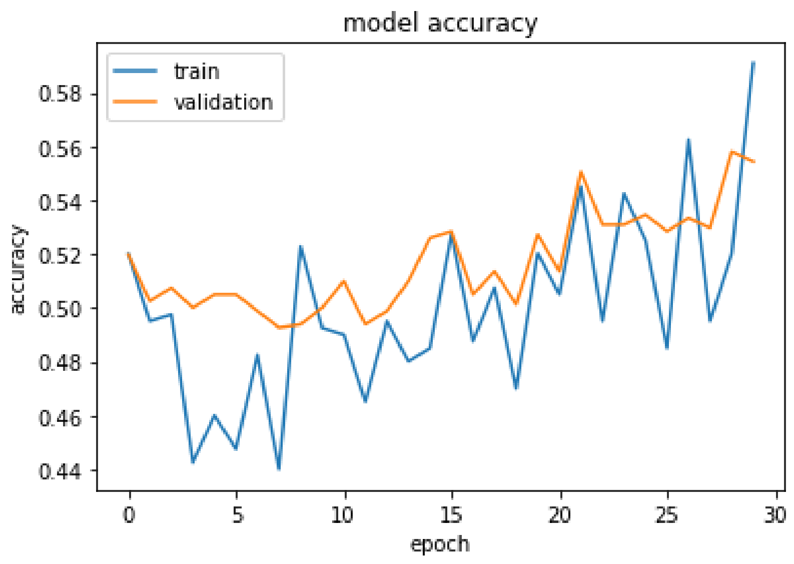


Figure 5.24: Confusion Matrix for Train Test for the dataset with softmax



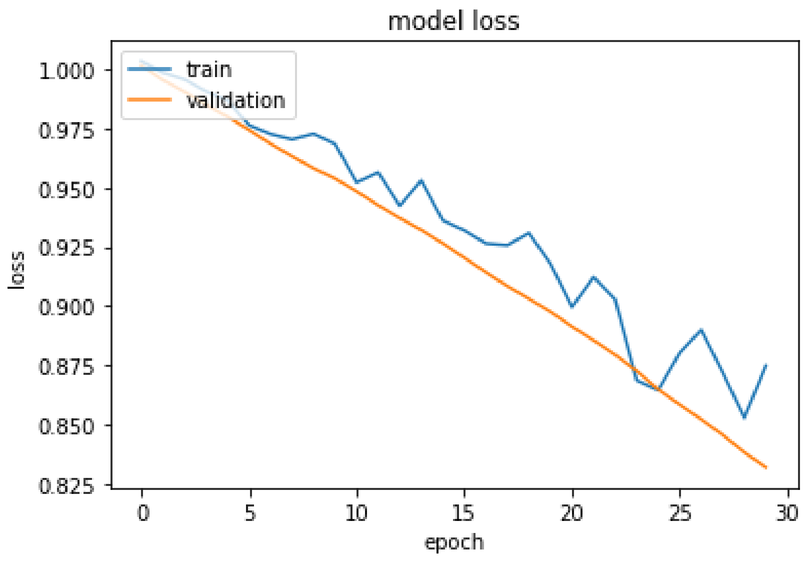
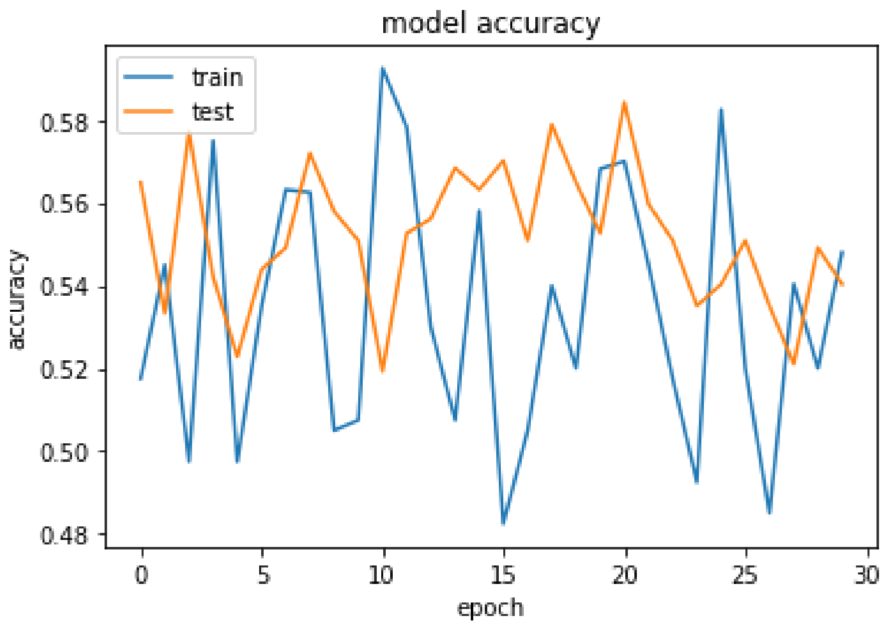


Figure 5.25: Train Validation for the dataset with SVM

In 5.25 We can see that the training accuracy is roughly 58%, and the validation accuracy is at roughly 54%. As for the test loss, it is roughly 0.85 while the training loss is also similar at 0.875.



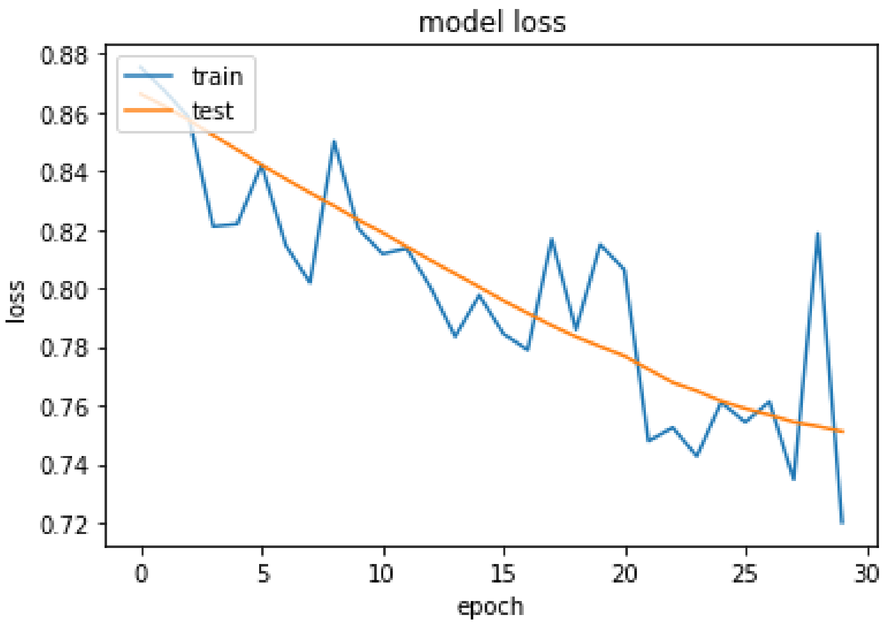


Figure 5.26: Train Test for the dataset with SVM

In 5.26 We can see that the training accuracy is roughly 54%, and the test accuracy is alos similar at roughly 54%. As for the test loss, it is roughly 0.76 while the training loss is also similar at 0.74.

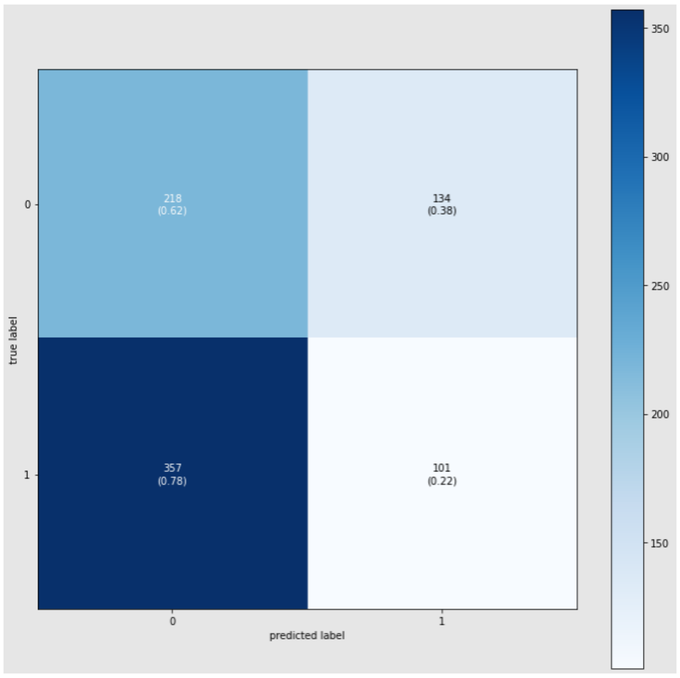


Figure 5.27: Confusion Matrix for Train Validation for the dataset with SVM

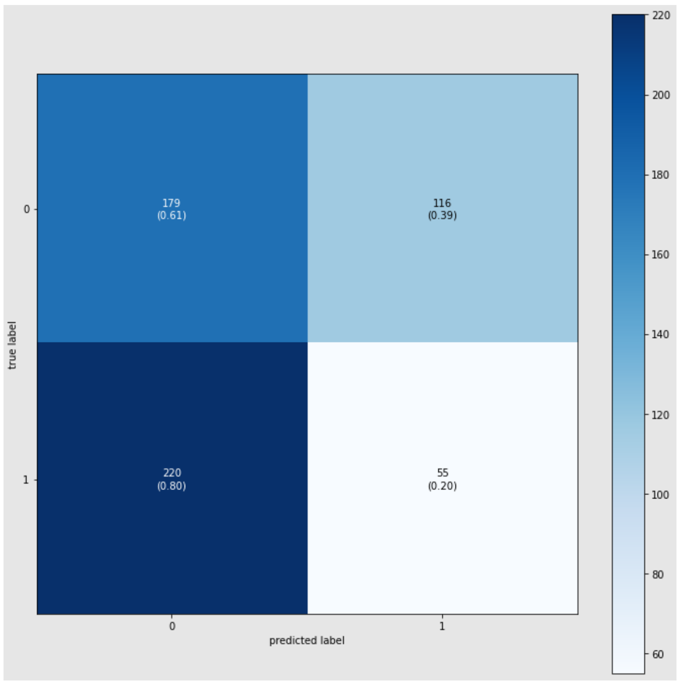


Figure 5.28: Confusion Matrix for Train Test for the dataset with SVM

5.4 Further Analysis

We decided to further analyze our results by simulating the accuracy graphs for when we used only the foreign dataset as both train and test 5.29. We also simulated results of when the foreign dataset is used as train dataset, and the entire dataset with mixed images were used as train in 5.30. We also used our own local dataset as train and the entire dataset as test in 5.31.

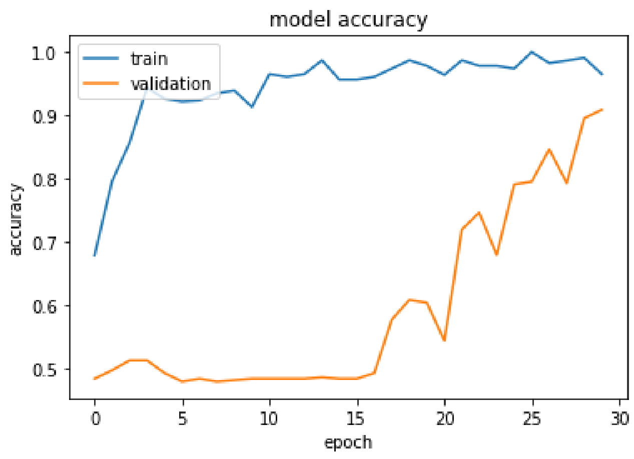


Figure 5.29: Foreign train with Foreign validation accuracy

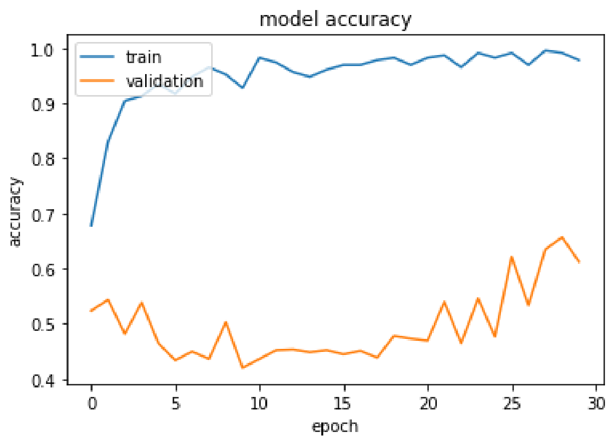


Figure 5.30: Foreign train with Mixed data validation accuracy

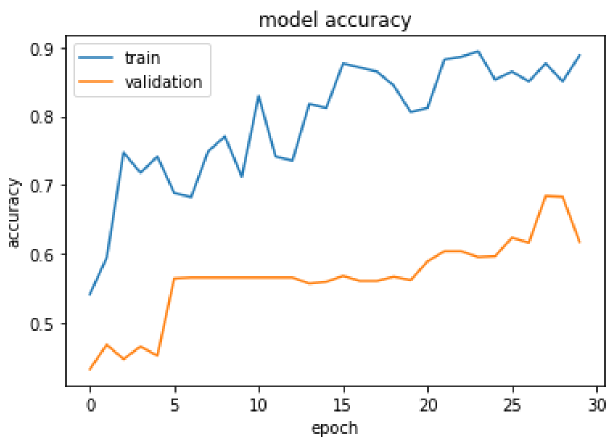


Figure 5.31: Local dataset train with Foreign dataset as validation

From the above graphs we can see that using only one of the type of data, be it foreign or our own local dataset, the accuracy of the validation data is not even close to the accuracy of when the entire dataset is combined. Furthermore, due to the foreign dataset containing most of the images as photoshopped, the accuracy is lower than it is when we used our local dataset which consisted of pictures which were all manually taken. This allows us to predict that the dataset performs much better to a random image if the training dataset contains more pictures which imitate real life rather than photoshopped pictures.

5.5 Summarizing the Results

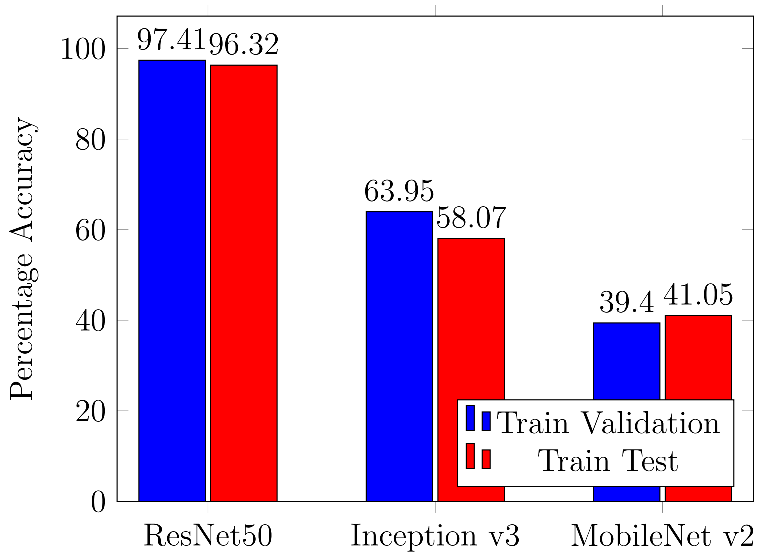


Figure 5.32: Percentage Accuracy for each Algorithm using SVM classifier

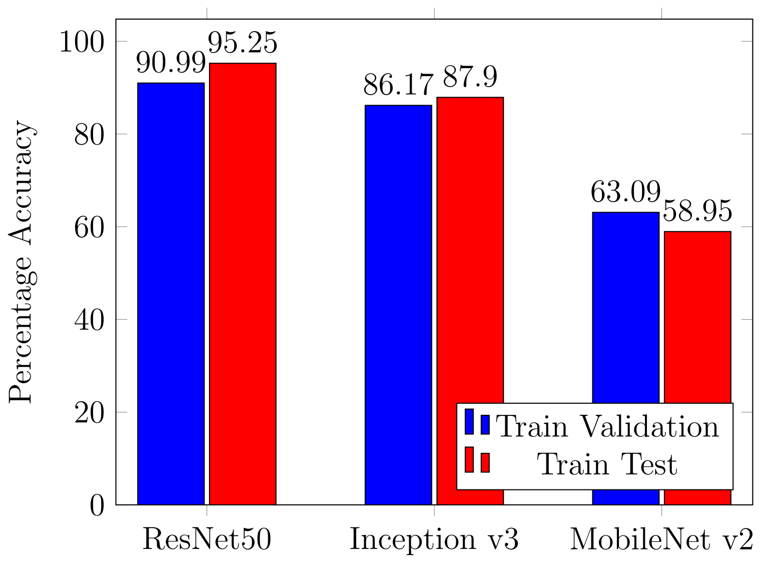


Figure 5.33: Percentage Accuracy for each Algorithm using Softmax classifier

Algorithms used	Model Accuracy				Model Loss			
	Train Validation		Train Test		Train Validation		Train Test	
	SVM	Softmax	SVM	Softmax	SVM	Softmax	SVM	Softmax
ResNet50	97.41	90.99	96.32	95.25	0.1	0.2	0.05	0.4
Inception v3	63.95	86.17	58.07	87.90	0.55	0.425	0.6	0.375
MobileNet v2	39.40	63.09	41.05	58.95	0.825	0.66	0.76	0.64

Table 5.1: Percentage Accuracy and Loss for each Algorithm

From the graphs above and the confusion matrix, we have tabulated the confusion matrix which predicted the data-set images and have considered this as the final accuracy and final loss. Here we can deduce that yet again Mobilenet v2 similar to Inception v3 runs better with softmax rather than SVM, achieving a better accuracy by roughly 10% than that of SVM. However, Mobilenet does seem to underperform when compared with both Inception v3 and Resnet50.
Resnet50 appears to be the algorithm which suits our data-set the best. Providing a constant accuracy of over 90% when Softmax is used, and an accuracy of over 95% when SVM is used as classifier, ResNet50 outperforms all the other algorithms by a fair margin. ResNet50 is then followed by Inception v3 and finally Mobilenet. SVM works best with Resnet, however underperforms when compared to Softmax for both Inception v3 and Mobilenet v2

Chapter 6

Conclusion

6.1 Conclusion

In countries with a high population density, airborne diseases spread rapidly and readily. Using face masks has become an unavoidable situation rather than an optional task. However, properly wearing a face mask is not a habit that everyone has grown to even at such dire situations of an ongoing pandemic such as the COVID-19. Bangladesh currently has a total number of cases of 782,000 due to the virus and is only increasing every day. Regardless of the virus being at work, Dhaka, the capital of Bangladesh, ranks 9th worst in the world in terms of Air Quality Index (AQI) from a recent study as of the morning of May 19, 2021. The mega city had an AQI score of 117 at 09:17 am. Furthermore, the air was classified as ”unhealthy for sensitive groups”. Considering all these factors, we can claim that the proper usage of masks is becoming more and more important by the day. Utilizing modern technology provided to our resources, we aim to combat these numbers as well as provide a safer environment to reside on.

6.2 Future Work

In this paper, we have focused only on if an individual is properly wearing a mask or not, which is a binary classification using different CNN architectures. In future, we would like to work on image segmentation, which would be able to tell us the exact position of a mask on a person's face. Moreover, The data-set images mostly are taken as close-up pictures, we can take more images from a further distance away from the camera to provide an even richer trained model. We also need to collect images taken under dim or bad lighting environment to identify images which were not taken under proper light intensities such as street lights. In addition to this, we also plan to implement the prediction model in a way as to be able to identify properly and improperly masked images directly from live feedback which will be fed from close circuit cameras using R-CNN architecture.

Bibliography

[1] P. Bahl and C. Doolan, "Charitha de silva, abrar ahmad chughtai, lydia bourouiba, and c raina macintyre. airborne or droplet precautions for health workers treating coronavirus disease 2019," *The Journal of infectious diseases*, 2020.

[2] I. M. Siegfried, "Comparative study of deep learning methods in detection face mask utilization," 2020.

[3] T. Irla, *Transfer learning using inception-v3 for image classification*, Oct. 2019. [Online]. Available: https://medium.com/analytics-vidhya/transfer-learning-using-inception-v3-for-image-classification-86700411251b.

[4] N. Kumar, *Batch normalization and dropout in neural networks explained with pytorch*, Dec. 2019. [Online]. Available: https://towardsdatascience.com/batch-normalization-and-dropout-in-neural-networks-explained-with-pytorch-47d7a8459bcd.

[5] Hassan, Feb. 2021. [Online]. Available: https://neurohive.io/en/popular-networks/vgg16/.

[6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, IEEE, vol. 1, 2001, pp. I–I.

[7] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, Ieee, vol. 1, 2005, pp. 886–893.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-shot multi-level face localisation in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5203–5212.

[10] A. Chavda, J. Dsouza, S. Badgujar, and A. Damani, "Multi-stage cnn architecture for face mask detection," *arXiv preprint arXiv:2009.07627*, 2020.

[11] W. Hariri, "Efficient masked face recognition method during the covid-19 pandemic," 2020.

[12] D. E. King, "Dlib-ml: A machine learning toolkit," *The Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the covid-19 pandemic," *Measurement*, vol. 167, p. 108 288, 2021.

[15] K. Hammoudi, A. Cabani, H. Benhabiles, and M. Melkemi, "Validating the correct wearing of protection mask by taking a selfie: Design of a mobile application" checkyourmask" to limit the spread of covid-19," 2020.

[16] B. Rehman, W. H. Ong, A. C. H. Tan, and T. D. Ngo, "Face detection and tracking using hybrid margin-based roi techniques," *The Visual Computer*, vol. 36, no. 3, pp. 633–647, 2020.

[17] A. Kumar, A. Kaur, and M. Kumar, "Face detection techniques: A review," *Artificial Intelligence Review*, vol. 52, no. 2, pp. 927–948, 2019.

[18] S. M. Bah and F. Ming, "An improved face recognition algorithm and its application in attendance management system," *Array*, vol. 5, p. 100 014, 2020.

[19] S. Pooja and S. Preeti, "Face mask detection using ai," in *Predictive and Preventive Measures for Covid-19 Pandemic*, Springer, 2021, pp. 293–305.

[20] S. Yadav, "Deep learning based safe social distancing and face mask detection in public areas for covid-19 safety guidelines adherence," *International Journal for Research in Applied Science and Engineering Technology*, vol. 8, no. 7, pp. 1368–1375, 2020.

[21] K. Suresh and V. Pattabiraman, "An improved utility itemsets mining with respect to positive and negative values using mathematical model," *International Journal of Pure and Applied Mathematics*, vol. 101, no. 5, pp. 763–772, 2015.

[22] Y. Chen, M. Hu, C. Hua, G. Zhai, J. Zhang, Q. Li, and S. X. Yang, "Face mask assistant: Detection of face mask service stage based on mobile phone," *IEEE Sensors Journal*, vol. 21, no. 9, pp. 11 084–11 093, 2021.

[23] M. S. Ejaz, M. R. Islam, M. Sifatullah, and A. Sarker, "Implementation of principal component analysis on masked and non-masked face recognition," in *2019 1st international conference on advances in science, engineering and robotics technology (ICASERT)*, IEEE, 2019, pp. 1–5.

[24] N. U. Din, K. Javed, S. Bae, and J. Yi, "A novel gan-based network for unmasking of masked face," *IEEE Access*, vol. 8, pp. 44 276–44 287, 2020.

[25] S. Ge, J. Li, Q. Ye, and Z. Luo, "Detecting masked faces in the wild with lle-cnns," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2682–2690.

[26] M. Jiang, X. Fan, and H. Yan, "Retinamask: A face mask detector," *arXiv preprint arXiv:2005.03950*, 2020.

[27] B. Qin and D. Li, "Identifying facemask-wearing condition using image super-resolution with classification network to prevent covid-19," *Sensors*, vol. 20, no. 18, p. 5236, 2020.

[28] A. Kabir, M. Islam, B. S. M. Rahman, M. S. Chowdhury, and M. J. Islam, “1st international conference on advances in science, engineering and robotics technology 2019 (icasert 2019),”

[29] J.-S. Park, Y. H. Oh, S. C. Ahn, and S.-W. Lee, “Glasses removal from facial image using recursive error compensation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 5, pp. 805–811, 2005.

[30] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[31] S. Yang, P. Luo, C.-C. Loy, and X. Tang, “Wider face: A face detection benchmark,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5525–5533.

[32] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, and A. K. Jain, “Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1931–1939.

[33] B. Yang, J. Yan, Z. Lei, and S. Z. Li, “Fine-grained evaluation on face detection in the wild,” in *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, IEEE, vol. 1, 2015, pp. 1–7.

[34] P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria, and J. Hemanth, “Ssdmnv2: A real time dnn-based face mask detection system using single shot multibox detector and mobilenetv2,” *Sustainable cities and society*, vol. 66, p. 102 692, 2021.

[35] M. Lin, Q. Chen, and S. Yan, “Network in network,” *arXiv preprint arXiv:1312.4400*, 2013.

[36] S. Sharma and S. Sharma, “Activation functions in neural networks,” *Towards Data Science*, vol. 6, no. 12, pp. 310–316, 2017.

[37] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation functions: Comparison of trends in practice and research for deep learning,” *arXiv preprint arXiv:1811.03378*, 2018.

[38] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[39] K. O’Shea and R. Nash, “An introduction to convolutional neural networks,” *arXiv preprint arXiv:1511.08458*, 2015.

[40] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[42] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, PMLR, 2015, pp. 448–456.

[43] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.