Face Mask Detection Using Convolution Neural Network Based Machine Learning Model

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE OF

MASTER OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

SUNNY

2K21/CSE/23

Under the supervision of

Dr. RAJNI JINDAL

(Professor)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

May 2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Sunny, Roll No. 2K21/CSE/23 student of MTech (Computer Science and Engineering), hereby declare that the Project Dissertation titled **"Face Mask Detection Using Convolution Neural Network Based Machine Learning Model"** which is being submitted by me to Delhi Technological University, Delhi, in partial fulfilment of requirements for the degree of Master of Technology in Computer Science and Engineering is original and is not copied from any source without proper citation. This work has not previously formed the basis for the award of any Dgree,Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi **Sunny**

Date: **2K21/CSE/23**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I, hereby certify that the Project titled **”Face Mask Detection Using Convolution Neural Network Based Machine Learning Model”**, which is submitted by Sunny, Roll No. 2K21/CSE/23, Department of Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of degree of MTech in Computer Science and Engineering is a genuine record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree to this University or elsewhere.

Place: Delhi **Dr. Rajni Jindal**

Date: Professor

ABSTRACT

In the year 2020 the world witnessed a global health emergency due to the outbreak of virus pandemic Coronavirus named COVID-19 also the increasing of the air pollution in the cities of developing countries due to emission of PM10 and PM2.5 pollutants from burning of Organic Fuels ,fast urbanization the usage of face masks in public has now become a way of life .The usage of face mask is not only mandated by the governments to control the spread of the virus but also a recommendation made by doctors to protect lungs of patients. Due to the usage of facial mask in public spaces with huge amount of footfalls namely markets, shopping malls, public transport like metro rails, sporting events ,music concerts manually check the proper usage of masks is not only a tedious and difficult task but also an impossible one for big country like India where population density is one of the highest in the world. The Public Monitoring systems widely used face many challenges to correctly monitor the usage of face mask due to difference in the mask types, low quality Cameras, Obsfuscation of faces etc. Also, majorly the lack of huge amount of data to be trained on is one of the main challenges. Therefore, this project is aimed at providing a comprehensive review of existing machine learning models that have been used to detect face masks and developing an ensemble approach for the same using a newer balanced dataset not widely used before.

ACKNOWLEDGEMENT

" It is not possible to prepare a project report without any assistance &

Encouragement of other people This one is certainly no exception."

On the very start of this project report, I would like to extend my sincere and heartfelt obligation to all the people who have helped me in this endeavor. Without their active guidance, help, cooperation encouragement, I would not have made headway in the project.

I am ineffably indebted to Prof. Rajni Jindal for her conscientious guidance, exceptional mentoring and constant monitoring to accomplish this assignment. Her valuable suggestions and ideas always proved helpful for me.

I am extremely thankful and would like to express my profound gratitude and deep regards to my faculty for encouraging me, clearing my doubts with all their zeal and constantly supporting me till the completion of this project.

I would also like to express my vote of thanks to the Computer Science Department (Computer Science and Engineering, DTU) for providing me with outstanding lab facility and LAN /Wi-Fi provision due to which I could Carry out this extensive project work without wasting my time.

I extend my gratitude to Delhi Technological University for giving me this opportunity.

I also acknowledge with a deep sense of reverence my indebtedness towards my parents and members of my family, who have always supported me morally as well as economically.

At last, but not least, a vote of thanks goes to all my friends who directly or indirectly helped me to complete this project.

Place: Delhi Date:

Sunny

2K21/CSE/23

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND NOMENCLATURE

1.CNN	Convolution Neural Network
2.ReLu	Rectified Linear Unit
3.SE-Block	Squeeze & Excitation Block
4.ML	Machine Learning
5.AI	Artificial Learning
6.conv	convolution
7.MaxPool	Maximum Pooling
8.AveragePool	Average Pooling

CHAPTER 1: INTRODUCTION

1.1) Overview:

The research area of image processing is one of the most widely pursued field due to its vast amount of application use in the day to day life. This has resulted in development of some of the very efficient and accurate algorithms using convolution neural networks [1]. The progress in the field of computation and memory technologies over the years has resulted in exponential growth in the ability to read, store and process vast amount of data very fast. The convergence of innovation in both the application side and hardware side has increased the use of machine learning. The automated traffic challan system is very good example of this. The arrival of COVID-19 comes with difficult challenges for health systems globally. The World Health Organization (WHO) declared a public health emergency In March 2020, as the covid continues to spread across the population, especially in vulnerable countries. The continuous mutations in the virus structure led to emergence of different variants and forced countries to take strict measures like travel restriction, national lockdown, isolation, and quarantine of individuals suspected to be exposed to virus and positive cases, advising the sanitization of hands at regular intervals, temperature monitoring, wearing of face masks, and social distancing. These restrictions presented huge challenges for developing nations having overburdened health systems, low funding and limited surveillance capabilities which impacted their potential efficacy. The vast restrictions are difficult to implement, due to different factors like public awareness, monitoring capabilities and policing. The process of monitoring a large number of people is an increasingly difficult task. The monitoring is the process to detect anyone who is not wearing a face mask.

Even after detection of improper behavior violating the mandated mask usage to ensure that it is corrected or to proceed with some punishment mechanism like issuing fines requires to identify the identity of the person. This is where the task becomes more difficult. Currently popular and used Face Identification Systems use Full Facial feature and Machine Learning to establish the identity of the person. But when there is a mask, it obfuscates the face, and it becomes more difficult to identify the person. In a low-resolution setting this task is much more difficult and the lack of datasets with face mask identities provides a challenge.

Due to the development of highly efficient and fast processor and memory devices the usage of Machine Learning for the purpose of solving artificial intelligence tasks is increasing. The Image processing is one the most famous field and there has been great developments of algorithms over the years. One issue for the public surveillance systems is the low quality images and low computational resources for processing the inputs.

1.1) Research Objectives:

This research project has following research objectives :

a): To study the different Convolution Neural Networks Models and compare their performance.

b): To Develop a Dataset for Face Mask Usage using Mask Superimposition.

c): To study the performance gain from using Ensemble of Machine Learning Models.

CHAPTER 2: THEORY

2.1) Machine Learning:

Machine learning is a field of computer science that seeks to develop algorithms that can enable computers to learn without explicit programming. It evolved from the study of pattern recognition and computational learning theory in artificial intelligence. Machine learning involves the construction of algorithms that can analyze data and make predictions or decisions based on the data patterns, rather than being solely controlled by pre-written code.

Machine learning is applied to various computing tasks, including spam filtering, network intrusion detection, optical character recognition, search engines, and computer vision. It has strong ties to mathematical optimization and computational statistics, which are also employed to make predictions through the use of computers.

Machine learning can be supervised or unsupervised, wherein supervised learning involves the inference of a function from labeled training data, while unsupervised learning entails inferring a function to describe hidden structures from unlabeled data. Reinforcement learning, on the other hand, is inspired by behaviorist psychology and is concerned with enabling software agents to take actions in an environment to maximize cumulative rewards.

Machine learning is critical in data analytics, as it enables the development of complex models and algorithms that aid prediction, also known as predictive analytics. It allows researchers and analysts to make reliable and repeatable decisions while uncovering hidden insights from historical relationships and trends in data.

Machine learning is divided into three main areas, namely supervised, unsupervised, and reinforcement learning. In supervised learning, the algorithm analyses the training data and produces an inferred function that can be used to map new examples. The goal is to enable the algorithm to generalize correctly from the training data to unseen situations. Unsupervised machine learning involves the inference of a function to describe hidden structures from unlabeled data. Reinforcement learning seeks to enable software agents to take actions in an environment to maximize cumulative rewards.

Machine learning is comprised of three main areas as seen in Figure 2.1:

a. **Supervised learning** :The machine learning task of inferring a function from labelled training data is known as supervised learning. A collection of training examples make up the training data. Each example in supervised learning is a pair made up of an input object, which is often a vector, and a desired output value, also known as the supervisory signal. An inferred function is generated by a supervised learning algorithm from the training data, which may then be used to map fresh samples. The algorithm will be able to accurately determine the class labels for instances that are not yet visible in an ideal environment. This necessitates that the learning algorithm generalise in a "reasonable" manner from the training data to hypothetical situations.

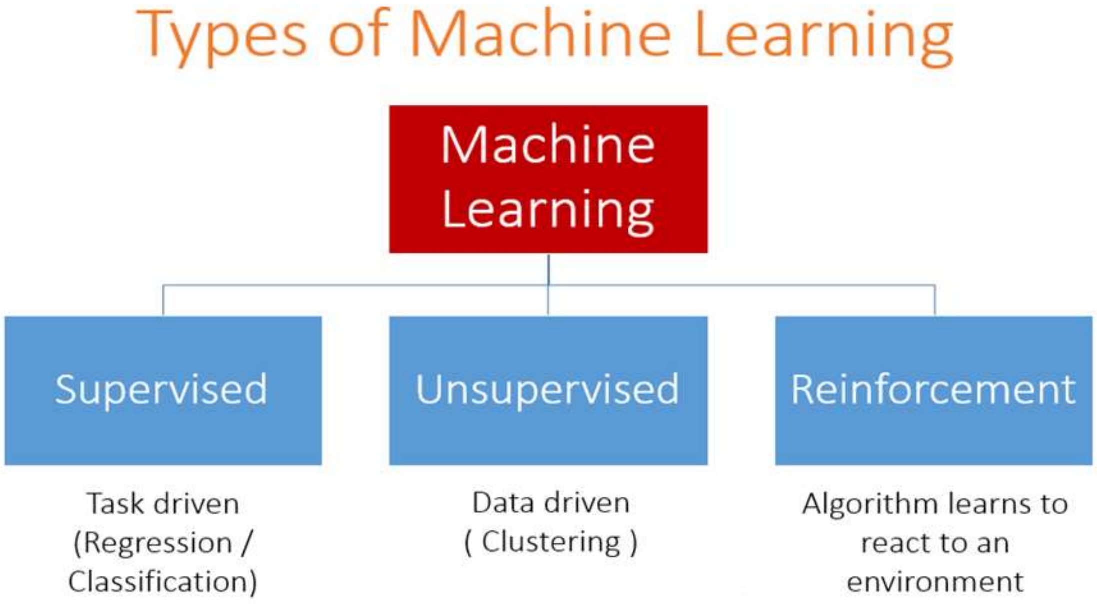

Types of Machine Learning
Machine Learning
Supervised
Unsupervised
Reinforcement
Task driven (Regression / Classification)
Data driven (Clustering)
Algorithm learns to react to an environment

Fig 2.1: Types of Machine Learning

b. **Unsupervised machine learning** :Unsupervised machine learning is the process of using unlabeled data to derive a function that describes a hidden structure. Unsupervised learning differs from supervised learning and reinforcement learning since the examples provided to the learner are unlabeled and do not contain any mistake or reward signals.

c. **Reinforcement learning** :A branch of machine learning called reinforcement learning, which was influenced by behaviourist psychology, is concerned with how software agents should behave in a given environment in order to maximise some sort of cumulative reward. Due to the problem's breadth, it is also investigated in a wide range of other fields, including statistics, genetic algorithms, multi-agent systems, game theory, control theory, operations research, information theory, and simulation-based optimisation. Although the majority of studies have focused on the existence of optimal solutions and their characterisation rather than the learning or approximation aspects, the subject has been investigated in the theory of optimal control. Reinforcement learning can be used to understand how equilibrium might develop under restricted rationality in economics and game theory.

Machine learning, which gives computers the ability to learn without explicit programming, is a crucial area in computer science. It has a wide range of uses since it may be applied to data analysis, forecasting, and decision-making. Its three main subfields are supervised, unsupervised, and reinforcement learning, each with distinct objectives and methods. Machine learning is poised to become increasingly important in our lives as hardware and software continue to advance, enhancing the intelligence and intuitiveness of how we interact with computers.

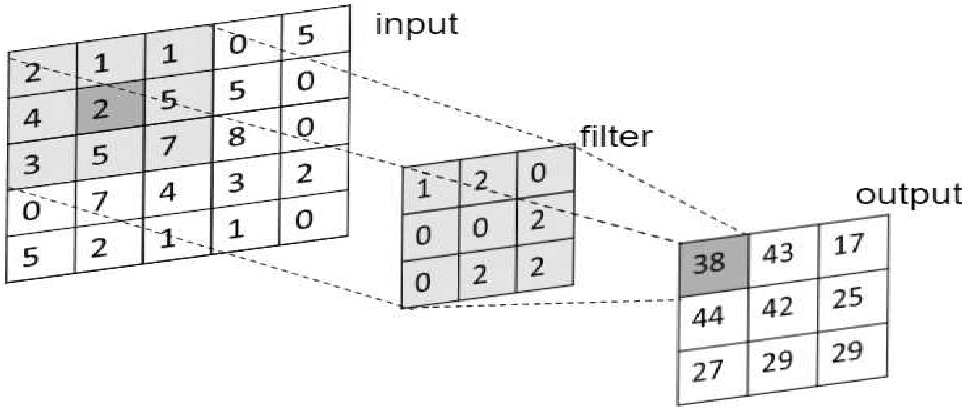
Another categorization of machine leaning tasks arises when one consider the desired output of a machine-learning system.

a. In classification, The learner must create a model that assigns unseen inputs to one or more (multi-label classification) of these classes when inputs are separated into two or more classes for classification. Usually, this is handled under supervision. When email (or other) messages are the inputs and the classifications are "spam" and "not spam," that is an example of categorization.
b. The outputs in regression, another supervised problem, are continuous as opposed to discrete.
c. In clustering, A set of inputs is to be separated into groups while performing clustering. Since the groupings are often unknown, unlike in classification, this is typically an unsupervised task.
d. Density estimation finds the distribution of inputs in some spaces.
e. Dimensionality reduction simplifies inputs by mapping them into lower dimensional space. Topic modelling is a related problem, where a Program is given cover similar human language documents and is tasked to find out which document and is tasked to find out which documents cover similar topics.

2.2)Image Processing Using Machine Learning:

The images are processed in machine learning algorithms using a operation called convolution operation where the image is transformed into a compressed form and using multiple filters of varying sizes the multiple regions of image are processed. The Layer used in Machine Learning Models utilizing this operation is called convolution layer and the newtworksare called Convolution Neural Networks(CNN) [1].They are special type of neural networks which are designed for designed to process vision inputs like images. This configuration was introduced by LeCun [1] in the year 1989. Essentially the image can be viewed as a matrix of numbers representing different spectrum of RGB.

An filter of different size, which is usually a square matrix made up of neurons with each neuron's weight describing the filter matrix is used to perform a dot product of input resulting in a vector representing image region in compressed information. Sliding this filter over whole image and combining all of the resultant vectors will output an matrix representing whole image in compressed form. This above operation is called an convolution operation depicted in Figure 2.2. The modern research in the design and framework of Convolution Neural Network [1] has resulted in breakthrough of the performances of CNN's. By sliding over the input an another feature map will be generated. Another layer is used in conjunction of convolution layers known as pooling layers. It is used to eliminate the repetitive information from the previous layer, hence it also helps in countering the problem of over-fitting.



2	1	1	0	5
4	2	5	5	0
3	5	7	8	0
0	7	4	3	2
5	2	1	1	0

1	2	0
0	0	2
0	2	2

38	43	17
44	42	25
27	29	29

Fig 2.2:Convolution Operation

There have been great improvements over the years in the different convolution operations used a convolution layer inside machine learning models. Some of them are:

a) Residual Connection: Residual Connection [5] or skip connection is an feed forward network connection where the output of a layer is used as an input in future layers.The input is aggregated together with the output of immediate previous layer. The main need of this connection aroused in deeply layered networks where the gradient value disappears as it travels further in the network as visible in Figure 2.3.

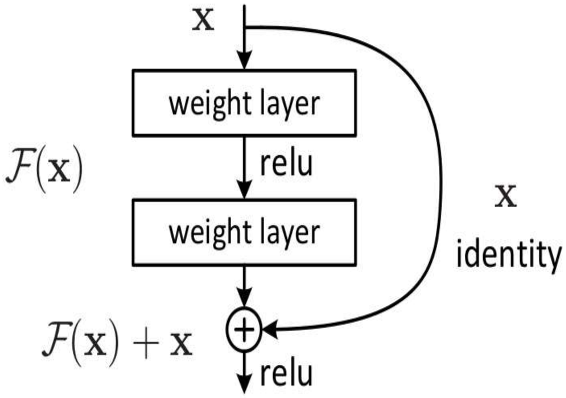

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Fig 2.3:Residual Connection

b) Dense Connection: A Dense Connection [6] is an operation in the neural network where a skip connection is made between all the previous preceding layer.hence each layer is connected with all future layers,hence the name Dense. The motivation for this connection comes from the gradient disappearing problem as discussed in above section in deeply layered networks.The Figure 2.4 shows a Dense Connection Block[6].

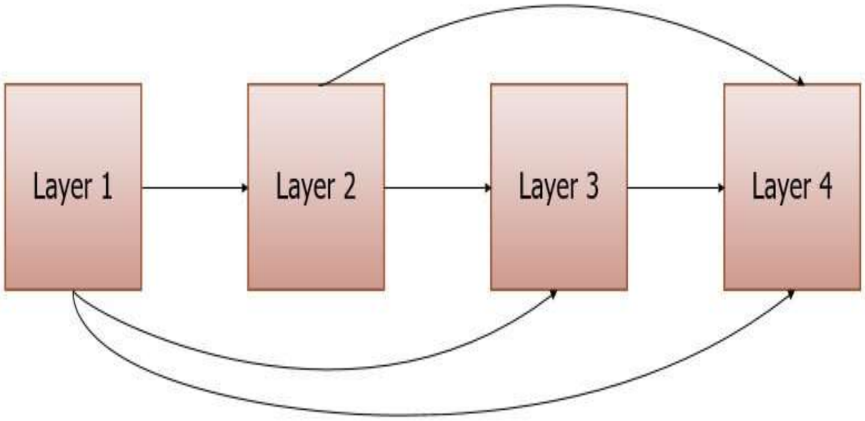


Fig 2.4:Dense Connection

c) Inverted Residual Block: An Legacy Residual Connection [5] the Convolution Operation is 1 x 1 Convolution followed by an 3 x 3 Convolution and again a 1 x 1 Convolution.But in Inverted Residual Connection introduced in [7], first a point-wise convolution [3] followed by a depth-wise convolution [3] and finally a 1 x 1 convolution is performed to reduce the size to input size as visible in Figure 2.5.This significantly the complexity of the Residual Block and there are less parameters in the model.

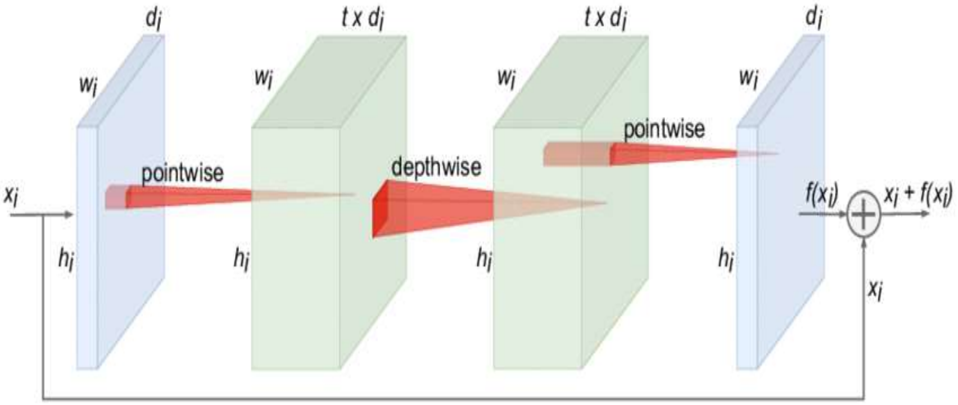


Fig 2.5:Inverted Residual Connection

d) Depth-wise Separable Convolution Layer: The Depthwise Convolution [3] operation in which a single filter per channel of the input is used.This operation is very cost efficient as compared to normal convolution operation.After this operation a Pointwise Convolution [3] Operation is applied to generate features equivalent to a normal convolution operation as seen in Figure 2.6. This combined operation is called Depthwise Separable operation [4].

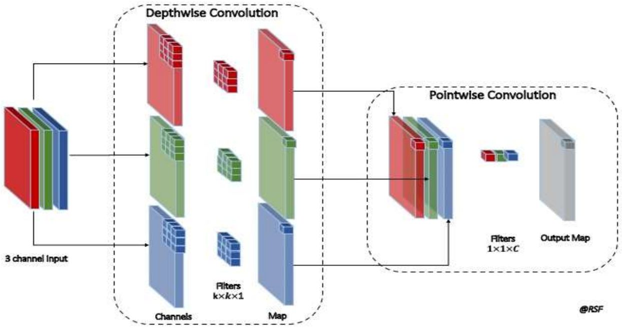


Fig 2.6:Depthwise Separable Convolution Operation

e) Inception Block: In Inception [9] block the higher convolutions operations are replaced with smaller convolution operations by using 1 x 1 convolution operation in start which reduces the depth of the input and hence help in reducing the complexity of convolution operation.The Figure 2.7 depicts the inception module [9].The input is passed through multiple stages in parallel where multiple convolutions are applied one after another and final results are concatenated.

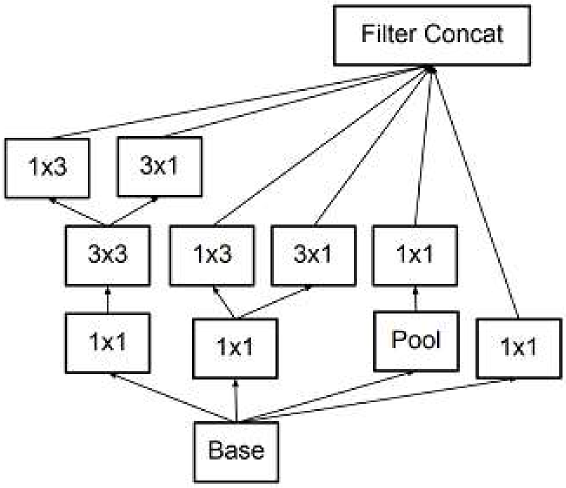


Fig 2.7:Inception Block

f) Squeeze & Excitation Block: During an Convolution Operation each channel is weighed equally important which will make all spatial features found in the input equally important.To improve the learning of more important features each channel is weighted differently which can be trained to determine the best weight-age for each channel. The SE-Block [8] has the Average Pooling operation first squeezes each channel to single unit and Further Fully Connected Layers with ReLu and Sigmoid activation performs the desired action.

Some of the Machine Learning Models which used and introduced the above convolution operations are:

a) Inception Convolution Neural Network: Inception [6] is a Deep Convolutions Neural Network which focuses on solving an higher dimension convolution operation using smaller convolutions. Consider the very basic convolution operation of 1 x 1, it reduces every pixel of the input image to a single value corresponding to the filter. This helps in reducing the size as well as depth of the input to the count of filter used in the current layer. The 1 x 1 convolutions are very useful to learn the information of small regions of the input. To get better understanding employing filters of varying sizes over the input image is a logical step, which will result in different regions of image being learned. Employing these operation in parallel and aggregating the output will result in very high usage of processing power. The Inception [6] module uses the concept of 1 x 1 convolutions to help in reducing the operational cost of higher convolution operation by employing an 1 x 1 convolution operation before an 3 x 3 or 5 x 5 convolution operation or higher. This results in reduction in depth of size of higher convolution and hence the efficient use of computational resources. The Figure 2.8 depicts the use of 1 x 1 operations before higher conv operation in an InceptionV3 [7] module.

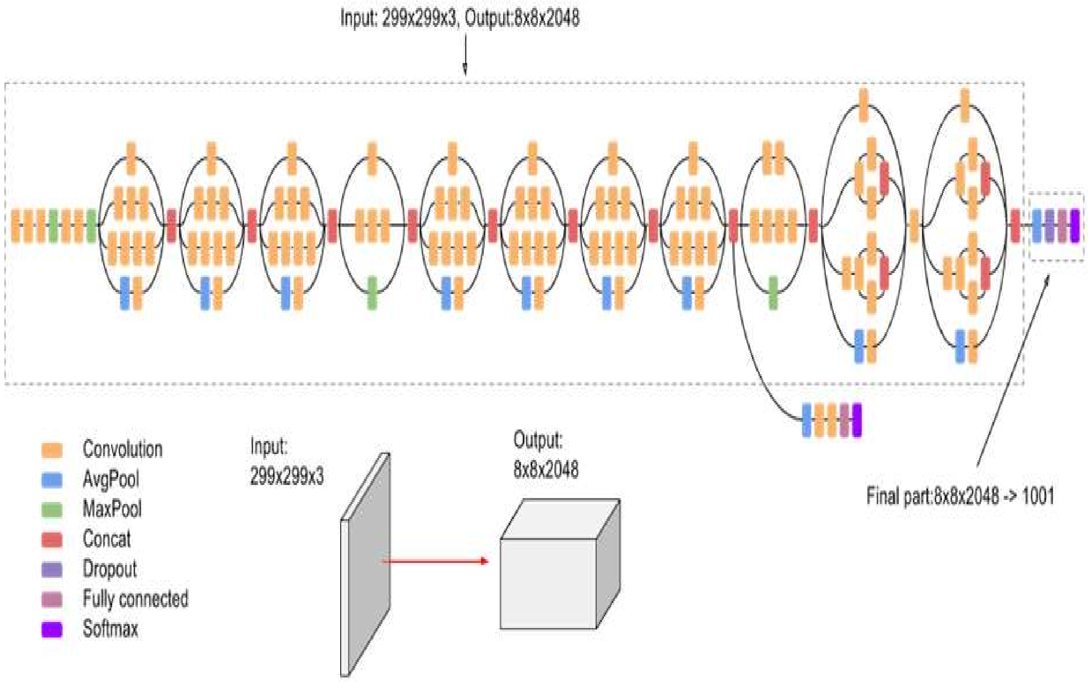


Fig 2.8:Inceptionv3 Architecture

b) MobileNet Convolution Neural Network:

MobileNet[8] is a deep Convolution Neural Network [1] framework which uses very low memory and computation designed primarily to be used on small devices likes Mobiles hence the name MobileNet [8]. MobileNet [8] model makes the use of depth-wise separable convolutions rather than utilizing the conventional convolution layer. MobileNet [8] framework introduces new hyper parameters i.e a width multiplier and resolution multiplier, which helps in managing the model design and can increase or decrease accuracy by increasing the latency or decrease it respectively. Each depth-wise separable convolution layer has two parts i.e a depth-wise convolution which is succeeded by a point-wise convolution. If we consider both of the depth-wise and point-wise convolutions as an unique layers, then this results in a total of twenty eight layers in the network model. The depth-wise convolution layer performs an individual convolution operation on every channel in input. After that point-wise convolutions layer combines the result of depth-wise convolution, linearly using 1x1 convolution operations.There is also an updated version MobileNetV2 [9] which uses inverted residual blocks. The Figure 2.9 depicts the architecture of the MobileNet [8] .

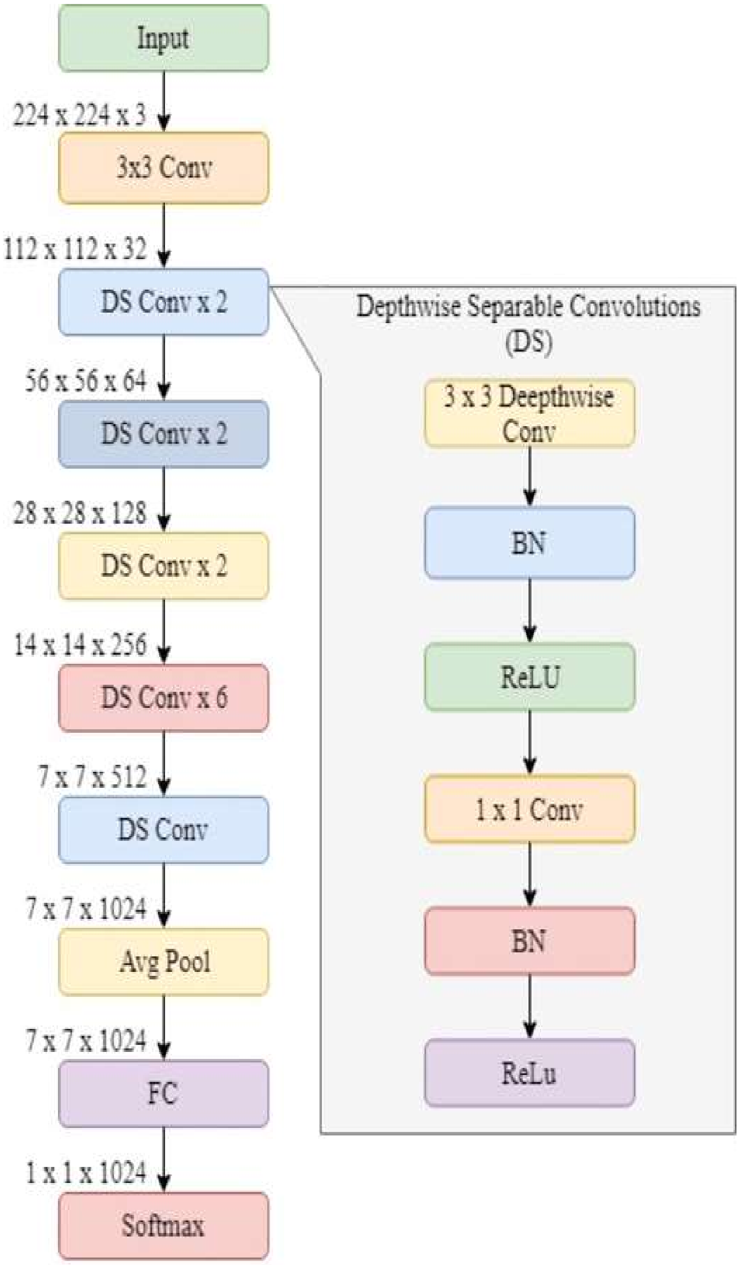


Fig 2.9:MobileNet Architecture

c) ResNet Convolution Neural Network: ResNet [11] is an artificial neural network that presented a so- called "identity skip connection" which allows the model to skip one or more layers. This way makes it feasible to train the network on thousands of layers without affecting performance. It is one of the most famous framework to tackle multiple computer vision tasks. Deep neural networks are very tough to train because of the infamous degrading gradient problem. As the gradient is back-propagated to earlier layers, continuous multiplication may make the gradient infinitely minute. As a result, the deeper the network goes, the further its performance becomes saturated or necessarily starts quickly degrading. ResNet [11] answered the problem using “residual connections”. It operates in two stages ResNet [11] creates multiple layers that are primarily not used, and skips them, reusing activation functions from preceding layers. At a second stage, the network re-trains over, and the “residual” convolution layers are amplified. This makes it possible to explore more region of the feature space which would have been skipped in a shallow convolution network configuration. ResNet50 [11] is a variant of [11] model comprising of total 50 layers more specifically forty eight convolution layers, one MaxPool and one AveragePool layer.

d) EfficientNet Convolutional Neural Network: Efficient-Net [12] is a convolution Neural Network [1] which uses optimised convolution operations and hence efficiently performs the desired task. A CNN has three different components regarding the input image which are its width, depth, and the resolution. All of these can be scaled as per requirements. The depth of an CNN represents the total number of different neural network layers used. The width of an CNN represents the count of neurons in a particular layer, it is different for each layer. The resolution of the network represents the width and height of current input image. By increasing the depth of the network, with the addition of convolution layer one over another can help in learning much complicated features of the input. Nevertheless the problem of gradient degradation is faced by doing this and the network becomes difficult to be trained. The effectiveness of the network performance depends heavily on the builduing blocks used to made it up. The basic block of this EfficientNet [12] framework is the mobile inverted bottleneck MBConv that’s also called inverted residual block with an added SE(Squeeze and Excitation) block. Residual blocks interlink the start and the ending of a convolution block with skip joins. In the Inverted Residual Block first channels will be widened by a point-wise convolution(conv 1 x 1) then uses a 3 x 3 depth-wise convolution that reduces significantly the number of parameters and finally it apply a 1 x 1 convolution operations which reduces the count of channels allowing the beginning and the end of the block to be added. Squeeze and Excitation (SE) Block improves the inter-dependencies between the multiple channels by assigning dynamic weights to more important channels.

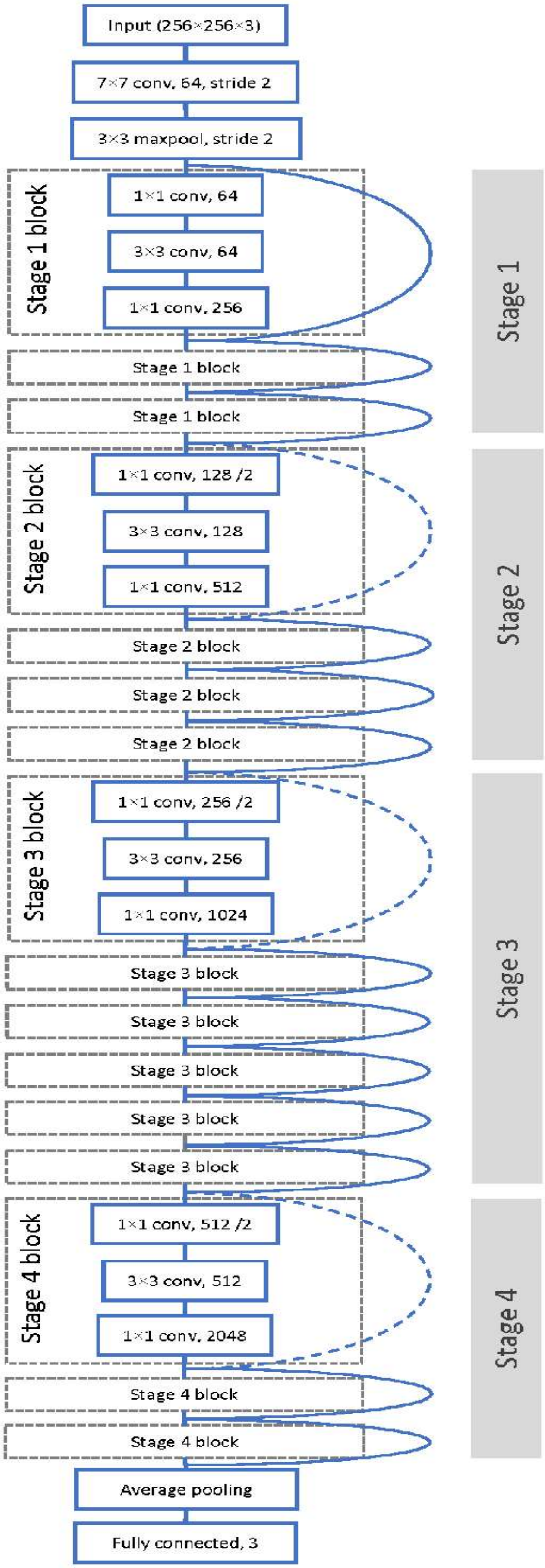


Fig 2.10:ResNet50 Architecture

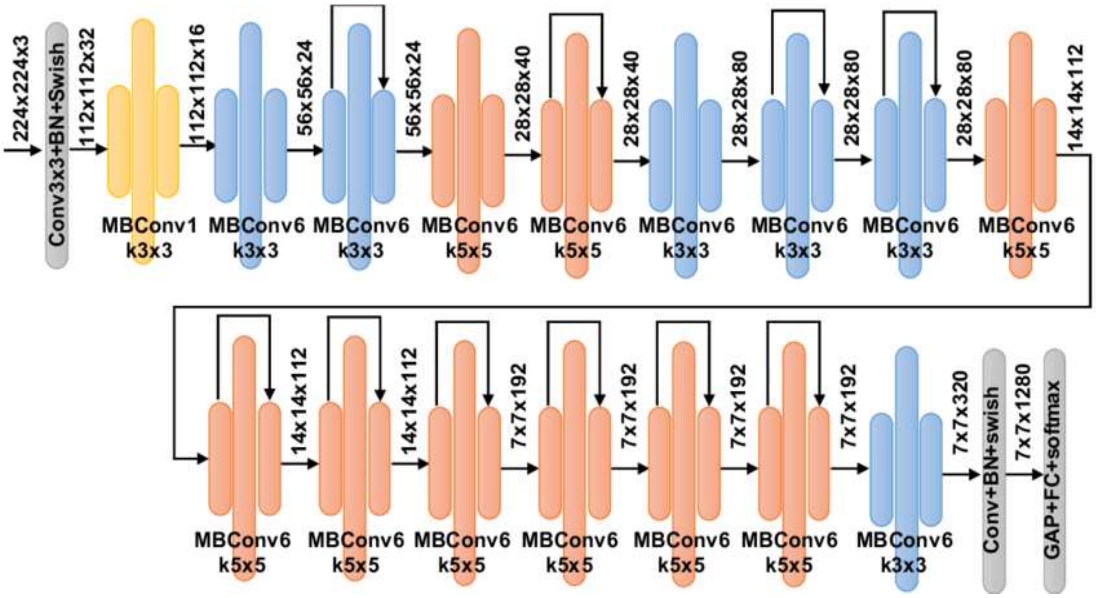


Fig 2.11:EfficientNetB0 Architecture

e) DenseNet Convolution Neural Network: Densely Connected Convolution Networks(DenseNet) [13] is a convolu- tional neural network(CNN) [1] framework which connects each layer with the previous layers. This allows the network to learn more effectively by reusing features, hence downgrading the number of parameters and improving the gradient flow during training. The framework of DenseNet [13] is made up of transition layers and dense blocks. Each convolutional layer inside a dense block is connected to every other layer within the block. This is achieved by linking the resultant of every layer to the input of the further following layers, making a "shortcut" link. The use of Dense Blocks in the framework is depicted in Figure 2.12. All the layers in these dense blocks are forward connected to all of the future layers inside block.

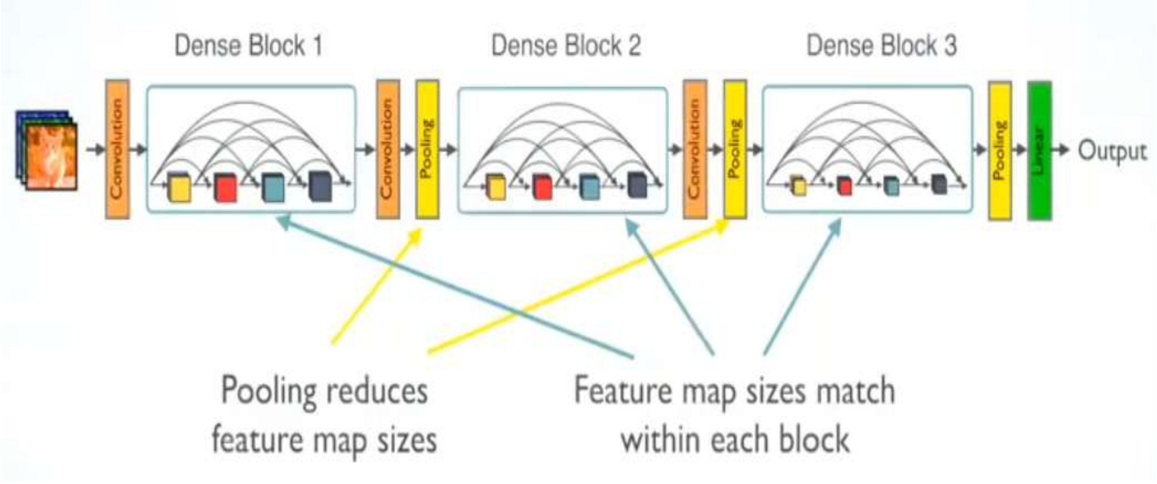


Fig 2.12:DenseNet Architecture

f) Xception Convolution Neural Network: Xception [14] Network is a very efficient convolution neuron network frame- work which comprises of Depth-wise Separable Convolution operations. Xception [14] is often called the “extreme” design of an Inception module. In Inception [6] network, the 1 x 1 convolution operations were applied to condense the initial input, and then different kind of filter maps are used on each of the depth of the input image. Xception [14] network is essentially just reverses the above step where the filters are first applied on the input data and then the input is compressed with 1X1 convolution by applying them on the depth of input. This strategy is nearly same to a depth-wise separable convolution. As seen in Figure 2.13 there are three stages of this model. The input travels from the first stage which is called entry flow, and then it passes from middle flow module and lastly from the exit flow. SeparableConv layer is the updated depthwise separable convolution. And there are residual connections, originally proposed by ResNet [11].

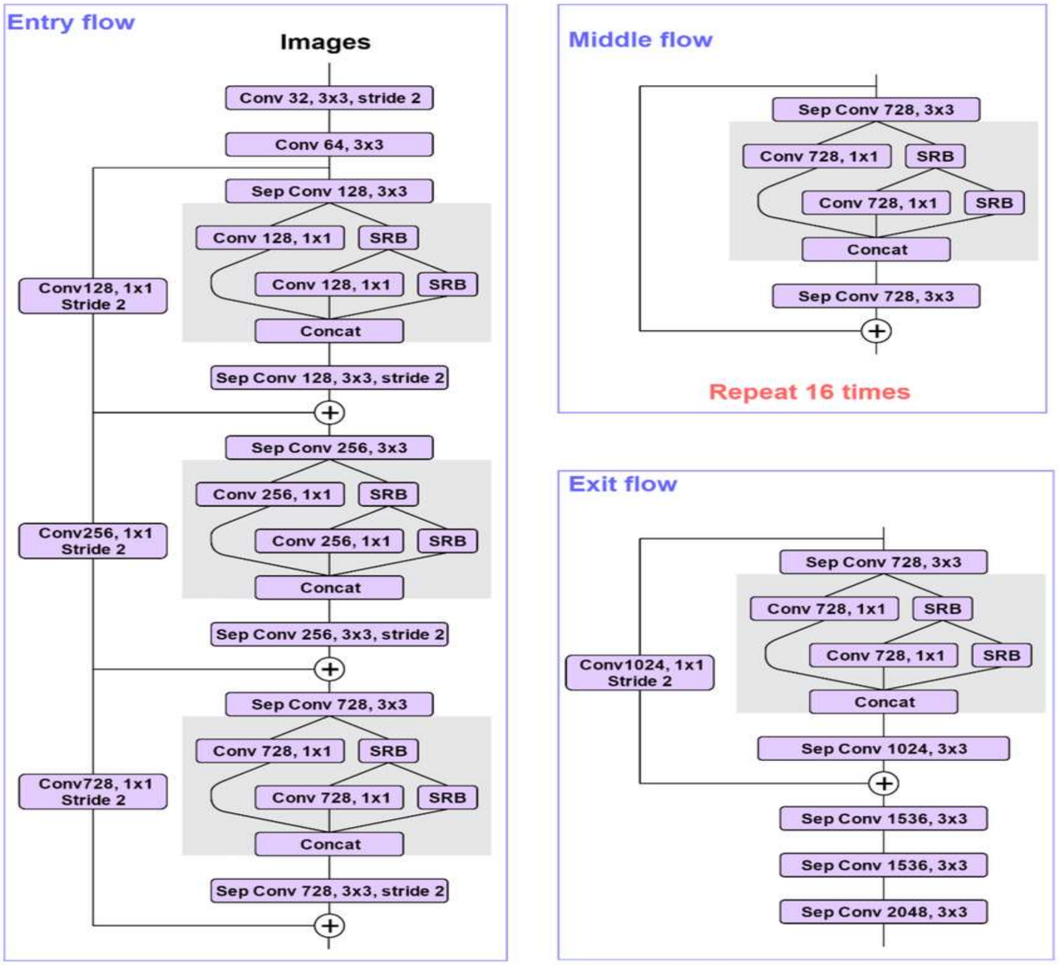


Fig 2.13:Xception Architecture

g) VGG Convolutional Neural Network: Visual Geom- etry Group or VGG [16] is a widely used deep convolution Neural Network [1] framework with many layers. It is called deep network because of the count of to the number of layers in the model with VGG16 [16] or VGG19 [16] have sixteen and nineteen convolution layers. The VGG16 [16] deep learning model is a fairly broad network. VGG16 [16] was popularised mainly due to the simple design framework. Starting from first layer the consecutive layer size is kept decreasing till a single layer of 1000 output nodes activated by softmax is produced. It contains some convolution layers which are then succeeded with a pooling layer which then reduces the size of the input data i.e height and the width. Due to no existing residual or skip connections it should perform less effectively than above discussed models. The VGG16 [16] model architecture is depicted in Figure 2.14 sourced from [17].

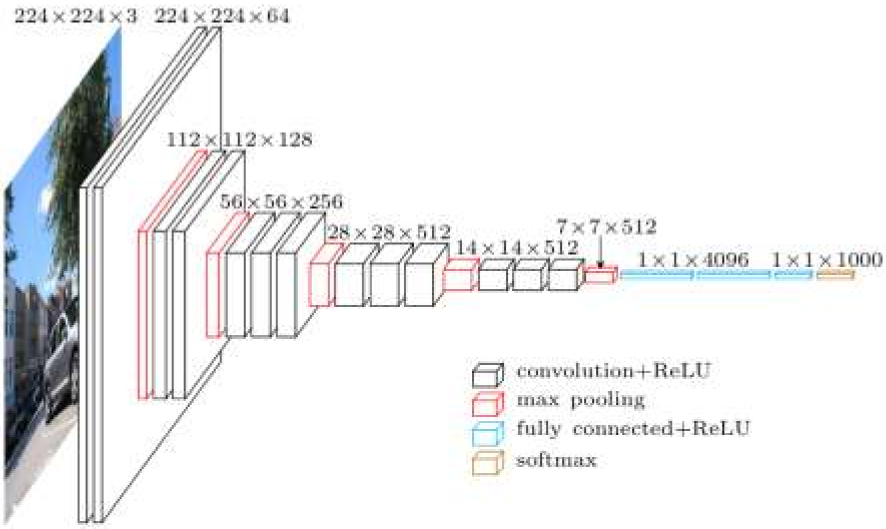


Fig 2.14:VGG16 Architecture

2.3) Performance Metrics

To compare different machine learning models I have used the following metrics:

1) Accuracy: The accuracy metric for a classification model is defined as the measure of total test data for which the prediction matches the actual value. The correct prediction is when the output class matches the actual class of the test data. The total predictions are the total cases of correct and incorrect predictions. The formula for measuring the accuracy is:

$$Accuracy = \frac{Correct\ Predictions}{Total\ Predictions} \tag{2.1}$$

2) Loss: The loss metric is used to measure the variation in the prediction to the actual value of the test data. In this project I used the Categorical Cross Entropy Loss, which is widely used for classification models. The formula for cross entropy loss is :

$$loss = -\sum_{c=0}^{N} y_{ic} \log\left(P_{ic}\right) \tag{2.2}$$

where c denotes class number and N is total number of class ,in this project it is 3,$y_{i,c}$ denotes the actual prediction in binary (0,1) and $p_{i,c}$ is the predicted value between the range [0,1].

3) F1-Score: The F1-Score is a metric that is widely used to compare the performance of multiple classifiers. The F1-Score is calculated as the harmonic mean of the two metrics Precision and Recall. In multi-class classification problem , such as thiss F1-Score is calculated as the average of scores for each class. In this project I use weighted average of these score to calculate the final score:

$$F1 = \sum_{c=0}^{N} Sample_c F1_c \tag{2.3}$$

where c denotes class number and N is total number of class ,in this project it is 3,$Sample_c$ denotes the number of items in class c and $F1_c$ is the F1-Score for a particular class c. The Value of score lies between [0,1] with 0 being the worst and 1 denotes perfect prediction.

4) The Kappa Score metric also known as Cohen Kappa Score [] was introduced by Cohen [] in 1960. It measures the agreement between two judges. In classification problem one judge being the dataset creator and another being the classifier. it ranges between [-1,1] with 0 and negative score signifying no agreement or inverse agreement and 1 signifies the perfect agreement. The formula to measure this score is :

$$Kappa = \frac{P_o - P_e}{1 - P_e} \tag{2.4}$$

Where P_o denotes the probability of agreement between predicted and actual class of image, P_e is the expected probability of agreement by chance.

2.4) Tools Used:

1) Python: Python is a widely used high-level programming language that is known for its simplicity, readability, and ease of use. It is ideal for data science, web development, artificial intelligence, machine learning, and other fields. Python comes with a large standard library that simplifies complex applications. It is also an interpreted language, which means it can be run on multiple platforms without needing to compile the code beforehand.
2) Jupyter Notebook: Jupyter Notebook is an open-source web application for creating and sharing documents containing live code, equations, visualizations, and explanatory text. It is a popular tool for data scientists, researchers, and engineers working on data analysis, data visualization, machine learning, and other tasks. Jupyter Notebook supports multiple programming languages, including Python, R, Julia, and Scala, and provides an interactive environment for explorative work.
3) Keras:Keras is an open-source neural network library written in Python. It simplifies building and training deep learning models using a simple and intuitive API. Keras allows users to create complex neural networks with just a few lines of code and supports various types of layers, including convolutional, recurrent, and dense layers. It can be used with different backends, such as TensorFlow, CNTK, and Theano. Keras is widely used in academic research and industrial applications.
4) Scikit-Learn:Scikit-Learn is a popular Python library for machine learning that offers a range of algorithms for classification, regression, clustering, dimensionality reduction and model selection. It features a consistent API for different algorithms and utilities for data preprocessing, cross-validation, and model evaluation. Scikit-Learn is easy to use and widely used in industry and academia for building predictive models.
5) Numpy:NumPy is a powerful Python library for numerical computing that provides a fast and efficient multi-dimensional array object and a large collection of mathematical functions for working with arrays. It is widely used for scientific computing, data analysis, and machine learning. NumPy provides a simple and intuitive API for array operations and supports various data types and indexing methods. It is a fundamental package for scientific computing with Python.

6) Matplotlib:Matplotlib is a Python library for creating visualizations. It offers a wide range of tools for data visualization, including line plots, scatter plots, bar charts, and histograms. Matplotlib is highly customizable and provides a rich set of options for controlling the appearance of visualizations. It is a popular tool for data analysts, scientists, and engineers for creating publication-quality plots and charts.

7) Visual Studio:Visual Studio is a widely used integrated development environment (IDE) developed by Microsoft. Programmers use it to develop software applications, web apps, and mobile apps using various programming languages, including C++, C#, Python, and others. Visual Studio provides a rich set of features, including code editing, debugging, testing, and deployment tools. It also includes integrated support for Git, Azure, and other cloud services. Visual Studio is a go-to tool for developers building high-quality software applications around the world.

CHAPTER 3: LITERATURE SURVEY

3.1) Related Works:

There have been many research works proposed for this problem statement some briefs about their works is presented below for better understanding.

1) SRCNET[19]: In this paper, a new algorithm known as the SRCNet [19] is discussed, which can identify the status of facemask-wearing in public through the use of convolutional neural networks. The SRCNet algorithm has the capability to detect three different categories of facemask-wearing conditions: no facemask-wearing, incorrect facemask-wearing, and correct facemask-wearing. The algorithm employs several techniques such as image pre-processing, facial detection and cropping, an SR network, and facemask-wearing condition detection methods to obtain the desired results. The study used large-scale facial image datasets and the Medical Masks Dataset to evaluate the algorithm's accuracy. The use of the SR network method in the SRCNet led to better performance in terms of detail enhancement and image restoration, ultimately improving the accuracy of identifying facemask-wearing conditions. SRCNet was capable of achieving a 98.70% accuracy in identifying three categories of facemask-wearing, outperforming traditional end-to-end image classification methods.This study addresses the challenges that are associated with identifying facemask-wearing conditions, including the limited availability of datasets and the various challenges of detecting facemasks that are being worn incorrectly. SRCNet's approach incorporated the use of an SR network and transfer learning prior to classification, which led to improved performance. The algorithm was designed with considerations for network complexity, providing an opportunity to apply it in public settings through IoT technologies, as a way of encouraging proper facemask-wearing practices for disease prevention. Although no previous studies have leveraged deep learning to identify facemask-wearing conditions, the study highlights the potential for automatic identification of facemask-wearing conditions using SRCNet [19].
2) Hybrid Deep Learning Model [20]: The paper discusses a mask face detection model that can be incorporated with surveillance cameras to identify individuals that are not wearing face masks and prevent the spread of the COVID-19 virus. The model is a combination of deep transfer learning and classical machine learning techniques. ResNet50 is utilized as a feature extractor, and classical machine learning algorithms such as decision trees, SVM, and ensemble are integrated to enhance the model's performance.The ResNet50 is preferred as

the feature extractor because of its higher accuracy rate compared with other models. This neural network type is based on residual learning and has 50 layers, with each containing 16 residual bottleneck blocks and three convolution layers. In the model, the last layer of ResNet50 is removed during classification and replaced with classical machine learning classifiers to ameliorate the model's effectiveness. Out of the various classifiers employed, SVM achieved the highest accuracy rate with the least training time.The proposed model exceeded previous works relative to testing accuracy. However, the model has one limitation, which is its failure to use most classical machine learning techniques to achieve the lowest consume time and highest accuracy rate.In conclusion, the integrated mask face detection model can aid in the identification of individuals not wearing masks and reduce the spread of COVID-19 via surveillance cameras. Further studies should investigate deeper transfer learning models for feature extraction and the neutrosophic area, which displays a promising capability in classification and detection issues.

3) FaceMaskNet [21]: Face mask detection is now achievable through the integration of Deep Learning techniques. In This paper we use the technology which has been divided into two phases: training face mask detection and implementing face mask detection. In the training phase, the model is serialized after loading data into the dataset, and the trained model is then applied to detect faces and mask regions in images and videos. Classifications of images and videos are determined as “with a mask”, “improperly worn mask” and “without a mask”. The FacemaskNet [21] model is used to support this process. The FacemaskNet[21] model architecture consists of an input layer containing images of the size 227 x 227 x 3. The 2D convolution layer is used as the input layer, followed by the ReLu layer that can activate functions. More efficient training and accurate results are guaranteed with the help of norm1 layer, where faster performance is achievable by using the maxpooling layer. Using the weight matrix, the input is multiplied, a bias vector is added using the fully connected layer, and classifications of the face and mask are obtained with the softmax layer. After training the FacemaskNet[21] model, it was implemented to images and live videos resulting in 98.6% accuracy in recognizing faces and masks. This face mask classifier was then applied to achieve the necessary classifications of faces and masks in images and videos.
4) Multi-Scale Face mask Detection [22]: The article discusses two different multi-scale face mask detection methods, namely Face Mask using YoloV3 (FMY3) and Face Mask Using NASNetMobile (FMNMobile). Both approaches rely on CNN-based transfer learning techniques, where pre-trained network models are

modified according to the particular problem requirement. The YoloV3 algorithm is a state-of-the-art single-stage object detection technique that uses Darknet-53 as the backbone architecture, whereas NASNetMobile was developed by the Google Brain team and uses both normal and reduction cells to achieve high mAP.In the FMY3 model, YoloV3 is used to detect the various states of face masks for a given input image. On the other hand, FMNMobile uses NASNetMobile as the backbone network to train face mask images. The authors gathered approximately 1,400 images of people with and without face masks and used a pre-trained Resnet-SSD100_caffee face detection model to detect faces in the input image.The paper concludes that both FMY3 and FMNMobile detection models achieved good accuracy. FMNMobile achieved higher accuracy with a recall rate of 98% compared to FMY3. It is highlighted that FMY3 depends on the annotations of the mask in the images, while FMNMobile detects human faces using state-of-the-art Resnet_SSD300 model and predicts face masks for the classification task.The second method uses Inceptionv3 which is a convolution neural networks based module provided by Google.It comprises of 22 layers of deep interconnected networks to increase the performance accuracy. It detects succesfully a person wearing a mask from the person not wearing any mask.But here the dataset usedin the study in very small image dataset. The model achieved 99.9% accuracy.

5) RCNN Model [23]: The proposed model is a Convolutional Neural Network (CNN) that utilizes region proposals and residual skip connections to improve the performance of object detection. The proposed architecture consists of four main components: a pre-trained CNN, a Region Proposal Network (RPN), RoI Pooling, and a Fully Connected Network. The pre-trained CNN extracts hierarchical features from the input image while the RPN produces bounding boxes for the region where the object is located. The RoI Pooling layer ensures that proposals of different sizes are of the same dimension as the input layer, and the Fully Connected Network outputs the class label and the bounding box coordinates. The proposed model outperforms the state-of-the-art models because of its use of residual blocks in the feature extraction stage, which in turn, prevents the problem of vanishing or exploding gradients. The efficacy of the proposed model was demonstrated by its ability to detect masked and non-masked faces.In conclusion, the proposed model is a sophisticated CNN that uses a range of techniques to improve object detection. The four main components of the proposed architecture, i.e., pre-trained CNN, RPN, RoI Pooling, and Fully Connected Network, work synergistically to produce accurate and reliable object detection results. Overall, the proposed model is an

important contribution to the field of object detection and could be useful for many applications, including face recognition and autonomous driving.

6) Yolov2 and ResNet50 Model [24]: The paper presents a new model for detecting medical masked faces to prevent the spread of COVID-19. The model comprises three main components: the number of anchor boxes, data augmentation, and the detector. The proposed detector uses the YOLOv2 with ResNet-50 [24] for feature extraction and detection during training, validation, and testing.To estimate the number of anchor boxes, the model uses mean Intersection over Union (IoU), which is a method for calculating the similarity between target bounding boxes and predicted outputs. The best number of anchors is determined to be 23, ensuring that the anchor boxes overlap with the boxes in the training data, hence improving the detector's performance.Data augmentation is used to improve the diversity of training datasets artificially, enhancing the detector's performance in training. The detector uses the YOLOv2 object detection deep network, which is composed of a feature extraction network and a detection network. The feature extraction network uses the ResNet-50 transfer learning model, which has 16 residual bottleneck blocks with feature maps ranging from 64 to 1024.The detection network, on the other hand, is a convolutional neural network containing few convolutional layers, a transform layer, and an output layer. The transform layer improves the stability of the deep neural network, while the output layer produces the locations of pure bounding boxes of the target.The model improves detection performance by adopting mean IoU to estimate the best number of anchor boxes. Additionally, a new dataset is designed based on two public masked face datasets to train and validate the detector in a supervised state. The proposed model is an effective tool for detecting medical masked faces, as shown by its high-performing outcomes. Performance metrics such as AP and log-average miss rates score had been studied for SGDM and Adam optimizer experiments.
7) M-CNN Model [25]: This paper presents a novel model that can detect individuals who are not wearing a facemask in public areas. The model uses facial detection technology to identify the person without a mask, and this data is combined with information from a public identification database to collect details about the individual, such as their name and address. The model will then send a fine amount to the individual's mobile number and address. To detect individuals without a mask, the paper proposes a new CNN architecture called M-CNN [25]. This architecture takes an input size of 150X150x3 and features a convolution layer for feature extraction using a 3X3 kernel of size 100. The second layer is a Max Pooling layer of size 2X2. The third layer takes input from

the previous layer and again convolves with a 3X3 kernel of size 100 followed by a max pooling layer of size 2X2. This is then followed by a flatten layer, a drop out layer, and finally, two dense layers. The first of these layers uses a Relu activation function, while the second uses a soft max activation function. Overall, the paper proposes a new model that can detect individuals in public areas who are not wearing a facemask. This model is based on a novel CNN architecture called M-CNN [25], which is designed to be highly effective in detecting individuals without a mask. The model can be used to enforce mask-wearing regulations and could help to prevent the spread of COVID-19 in public spaces.

8) Cascade Framework CNN [26]: This presents a face detector that uses a CNN cascade framework. It has three binary CNN classifiers called "Mask-12", "Mask-24-1", and "Mask-24-2". These classifiers can detect faces at different levels of classification ability. When an image is inputted into the detector, the three CNNs evaluate the image and eliminate any false detection windows. After each CNN, non-maximum suppression (NMS) is used to merge highly overlapped candidate windows. The final detection results are outputted after all evaluations are complete. The first stage CNN classifier is called Mask-1, which is shallow with five layers, and it's fully convolutional to adapt to any input image size. The second stage CNN classifier is Mask-2, which is deeper than Mask-1. The remaining detection windows from Mask-1 are cropped, resized to 24 x 24, and inputted into Mask-2. Like Mask-1, Mask-2 evaluates each detection window and eliminates any below the preset threshold. NMS then merges any remaining highly overlapped candidate windows. The last stage CNN classifier is Mask-3, which considers both classification ability and detection efficiency. The remaining detection windows from Mask-2 are cropped, resized to 24 x 24, and inputted into Mask-3. Mask-3 evaluates each detection window and eliminates any below its preset threshold. NMS merges any remaining highly overlapped candidate windows to produce the final detection result. The algorithm proposed in this paper is a deep learning-based approach to masked face detection, using a newly designed CNN cascade framework [26] that includes three CNNs.

9) RetinaFaceMask [27] : To handle the various scenarios in face mask detection, ResNet50 is used as a powerful feature extraction network. Intermediate feature maps from convolution layers with different receptive fields were generated from ResNet50 for multi-scale detection. However, shallow layers cannot provide sufficient semantic information, which affects detection accuracy. Therefore, the Feature Pyramid Network (FPN) was employed to address this

issue. The RetinaFaceMask [27] -Light model uses the MobileNetV1 backbone for efficient embedded device running. Face mask detection involves face localization and mask wear state discrimination. To improve feature extraction for mask wear states, a Convolutional Attention Module (CAM) was proposed, which uses parallel sub-branches with varying receptive fields. Channel and spatial attention are also implemented to focus on crucial face mask-related features. Due to the challenging and uncontrolled nature of in-the-wild scenes, obtaining more annotated data is one solution. However, RetinaFaceMask [27] uses knowledge transfer learning from face detection to aid in feature learning for face mask detection. RetinaFaceMask [27] achieved superior results on various face mask datasets, including a 4% increase in mAP compared to the baseline on the AIZOO dataset.

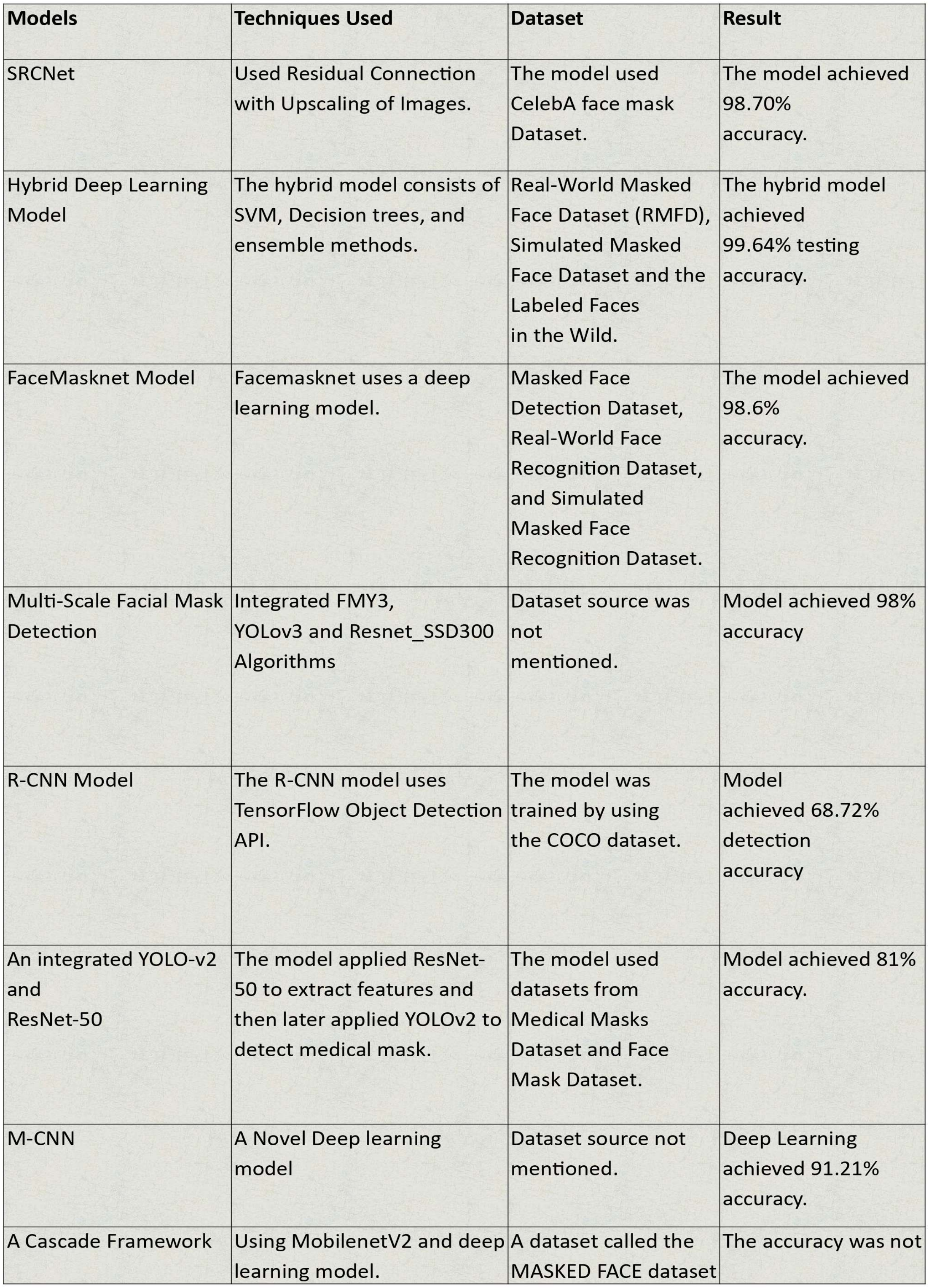
10) Yolov3 Model [28]: The YOLOv3 model [28] takes in an image and detects the object's coordinates by dividing the input into a grid and analyzing target object features from neighboring cells with high confidence rates. This study created a method to detect whether a person is wearing a mask or not using YOLOv3 architecture, performing well in images and real-time video with an impressive average fps of 17. Although this dataset wasn't very diverse, this custom object detection model showed promising accuracy when tested with real-world data. The focus of this research was on building a custom model rather than creating the entire architecture.

11) Multi-Stage CNN [29]: The architecture for detecting face masks has two main stages, with the first stage being the Face Detector and the second stage being the CNN based Face Mask Classifier. The Face Detector is responsible for identifying multiple faces in an image of varying sizes and situations, even when the faces overlap, and extracting the detected faces as regions of interest. These regions of interest are then processed and batched together for the second stage.The Face Detector uses an RGB image as input and outputs the detected faces with their bounding box coordinates. The accuracy of detecting the faces is crucial for the overall success of the architecture. The Intermediate Processing Block carries out further processing of the detected faces to prepare them for classification.The Intermediate Processing Block expands the bounding boxes of each face by 20% to cover the required Region of Interest with minimal overlap with other faces. Then, the faces are cropped out from the image to extract the ROI for each detected face. These faces are then resized and normalized for Stage 2.Stage 2 is the Face Mask Classifier, which takes the processed ROIs from the Intermediate Processing Block and classifies them as either Mask or No Mask. To train the classifier, an unbiased dataset of masked and unmasked faces

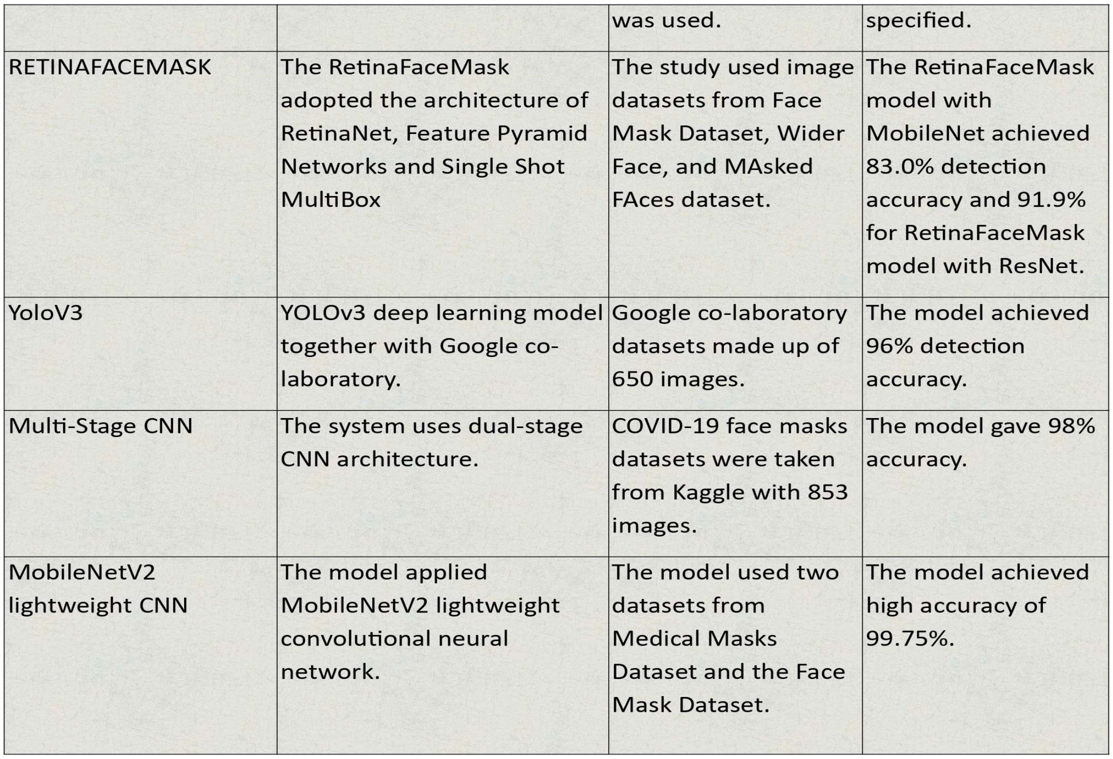
was created, and three lightweight models were trained on this dataset. Based on performance, the NASNetMobile model was selected as the best fit for classifying faces as masked or non-masked.Overall, the two-stage Face Mask Detector architecture has proven to be effective in accurately detecting and classifying faces as masked or unmasked. The use of a pretrained RetinaFace model for robust face detection and the NASNetMobile model for classification has ensured reliable results. This system has potential applications in public spaces to ensure compliance with mask mandates and reduce the spread of COVID-19.

12) MobileNetV2 Based Model [30]: The face mask detector discussed in this paper uses a two-stage detector framework. The first stage involves training the model using transfer learning with MobileNetV2 as a face mask classifier that can classify images into two categories: faces with masks and faces without masks. Once trained, the classifier is saved to disk. In the second stage, the saved classifier is loaded and used to detect real-time images by first using Haar feature-based cascade classifiers for face detection. These classifiers work on grayscale images and return the coordinates of detected faces which are then passed to the face mask classifier to determine if a mask is being worn.This approach was implemented to create a face mask detector for use during the COVID-19 pandemic, using a dataset of 11,800 images for training. Experimental results showed high accuracy, with a training accuracy of 99.9% and testing accuracy of 99.75%.

3.2)Comparison of Existing Works:



Models	Techniques Used	Dataset	Result
SRCNet	Used Residual Connection with Upscaling of Images.	The model used CelebA face mask Dataset.	The model achieved 98.70% accuracy.
Hybrid Deep Learning Model	The hybrid model consists of SVM, Decision trees, and ensemble methods.	Real-World Masked Face Dataset (RMFD), Simulated Masked Face Dataset and the Labeled Faces in the Wild.	The hybrid model achieved 99.64% testing accuracy.
FaceMasknet Model	Facemasknet uses a deep learning model.	Masked Face Detection Dataset, Real-World Face Recognition Dataset, and Simulated Masked Face Recognition Dataset.	The model achieved 98.6% accuracy.
Multi-Scale Facial Mask Detection	Integrated FMY3, YOLov3 and Resnet_SSD300 Algorithms	Dataset source was not mentioned.	Model achieved 98% accuracy
R-CNN Model	The R-CNN model uses TensorFlow Object Detection API.	The model was trained by using the COCO dataset.	Model achieved 68.72% detection accuracy
An integrated YOLO-v2 and ResNet-50	The model applied ResNet-50 to extract features and then later applied YOLOv2 to detect medical mask.	The model used datasets from Medical Masks Dataset and Face Mask Dataset.	Model achieved 81% accuracy.
M-CNN	A Novel Deep learning model	Dataset source not mentioned.	Deep Learning achieved 91.21% accuracy.
A Cascade Framework	Using MobilenetV2 and deep learning model.	A dataset called the MASKED FACE dataset	The accuracy was not



		was used.	specified.
RETINAFACEMASK	The RetinaFaceMask adopted the architecture of RetinaNet, Feature Pyramid Networks and Single Shot MultiBox	The study used image datasets from Face Mask Dataset, Wider Face, and MAsked FAces dataset.	The RetinaFaceMask model with MobileNet achieved 83.0% detection accuracy and 91.9% for RetinaFaceMask model with ResNet.
YoloV3	YOLOv3 deep learning model together with Google co-laboratory.	Google co-laboratory datasets made up of 650 images.	The model achieved 96% detection accuracy.
Multi-Stage CNN	The system uses dual-stage CNN architecture.	COVID-19 face masks datasets were taken from Kaggle with 853 images.	The model gave 98% accuracy.
MobileNetV2 lightweight CNN	The model applied MobileNetV2 lightweight convolutional neural network.	The model used two datasets from Medical Masks Dataset and the Face Mask Dataset.	The model achieved high accuracy of 99.75%.

Table.1: Comparison of Different Methods

CHAPTER 4: METHODOLOGY

4.1) Steps of The Experiment:

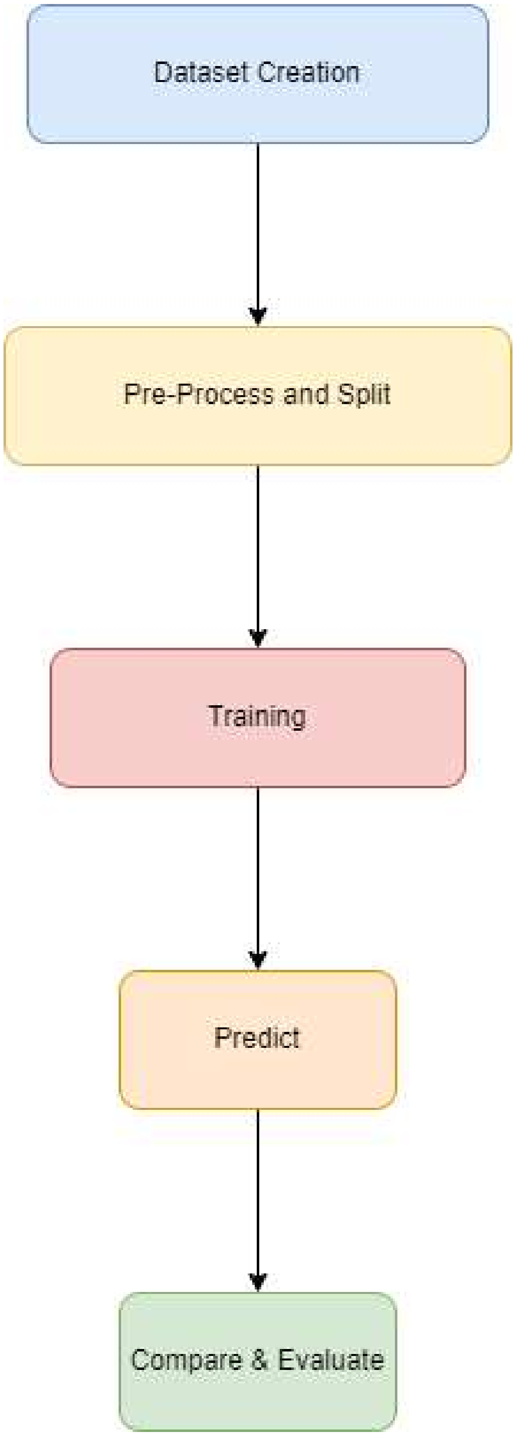


Fig 4.1:Steps of the Experiment

The following steps are followed in the experiment:

1. Dataset Creation: In this step the base image is taken and using the image superimposition technique mask is placed on the faces to create the dataset.
2. Pre-process & Split: In this step the Image are resized and converted to tensors which are understandable for the machine learning algorithms. Further the Dataset is split into different groups for training and testing stages.
3. Training: In this step the Machine Learning models are created and the trained using the dataset.
4. Predict: The test sample of the dataset is then used to predict the type of mask usage in the image.
5. Compare & Evaluation: In this step the different models are compared and evaluated on the performance metrics.

4.2) Dataset & Dataset Generation:

In this experiment two different datasets have been used .One is the publicly available dataset and another is generated using the Image Superimposition Technique.

1. **Face Mask Dataset:** The Face Mask Detection[31] dataset contains 3 different directories defining to which class each of the image belongs to. The 3 classes are with: a)mask,*b) without mask,*c) mask weared incorrect. Each folder holds 2994 images of people that belong to such a labeled class.

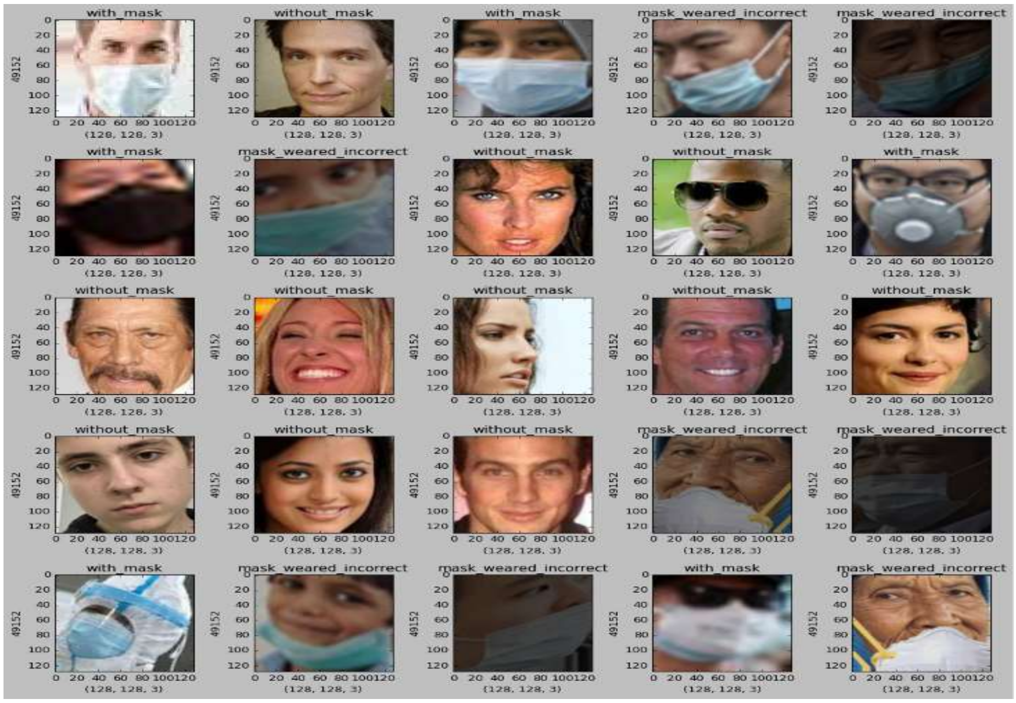


Fig 4.2:Face mask Dataset

2. **Artificial Dataset:**

To create the artificial dataset following steps were followed using the algorithm from [32]:

A) Identify Facial feature Points: to identify the Facial features in an image I use dlib's library to identify the 68 points in a face.

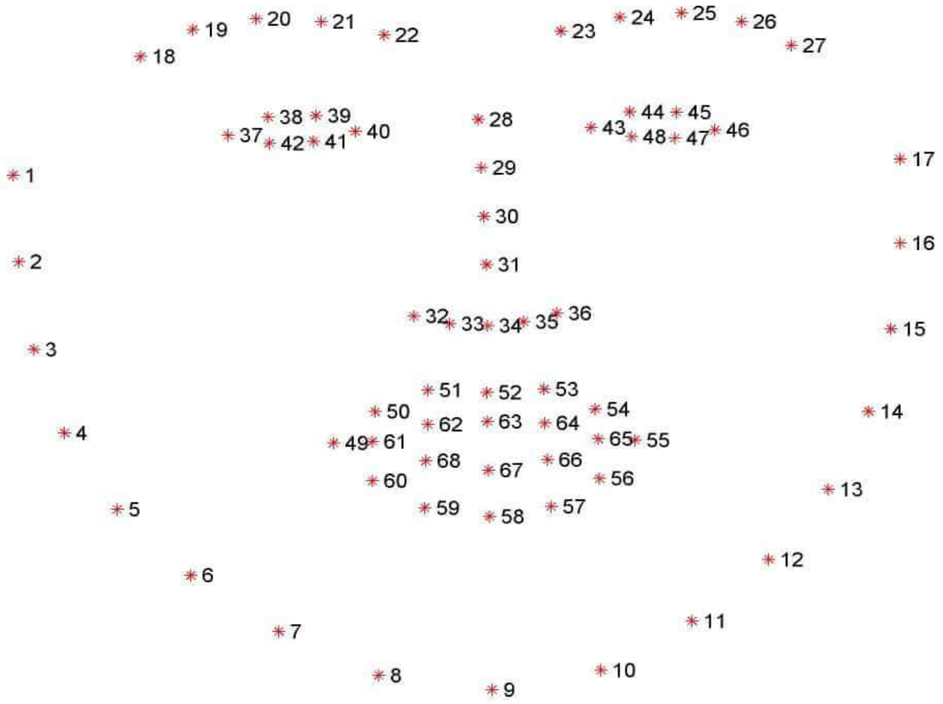


Fig 4.3:The Face Landmarks Mask.

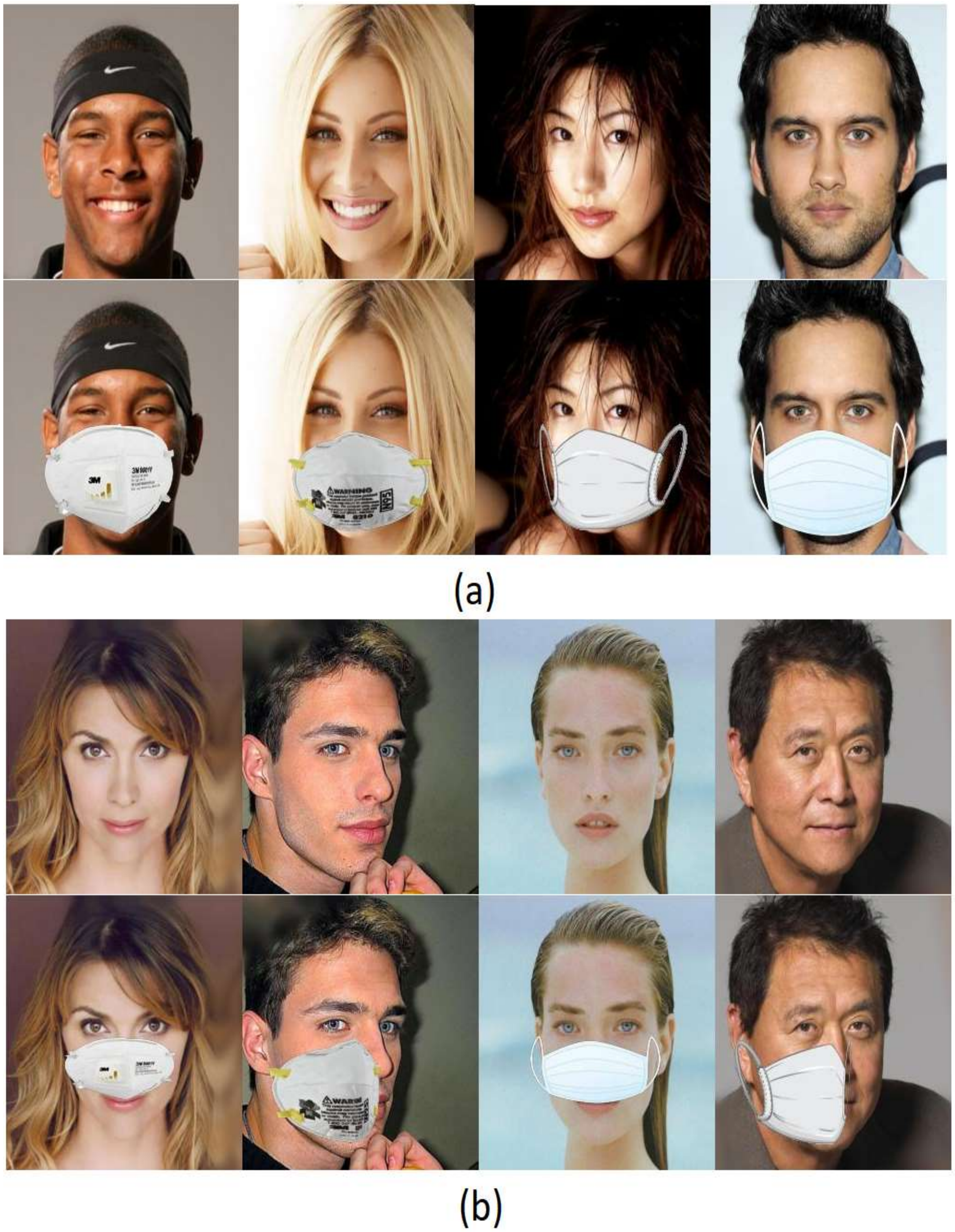
B)Then I identify the orientation of image by calculating the angle between chin end point and nose.

C) Further four different types of face masks were used to generate the masked faces for better distribution in the dataset.

Fig .4.4:The Different Face Masks Types Used in Dataset.

D)The Mask are then superimposed on the image according to points identifired in the face landmark.

Due to the limitations of the face landmark detection there were some ambiguities and some masks were not placed correctly.These images were segregated and place in the improper mask class for this dataset.The output samples of the dataset is given below.



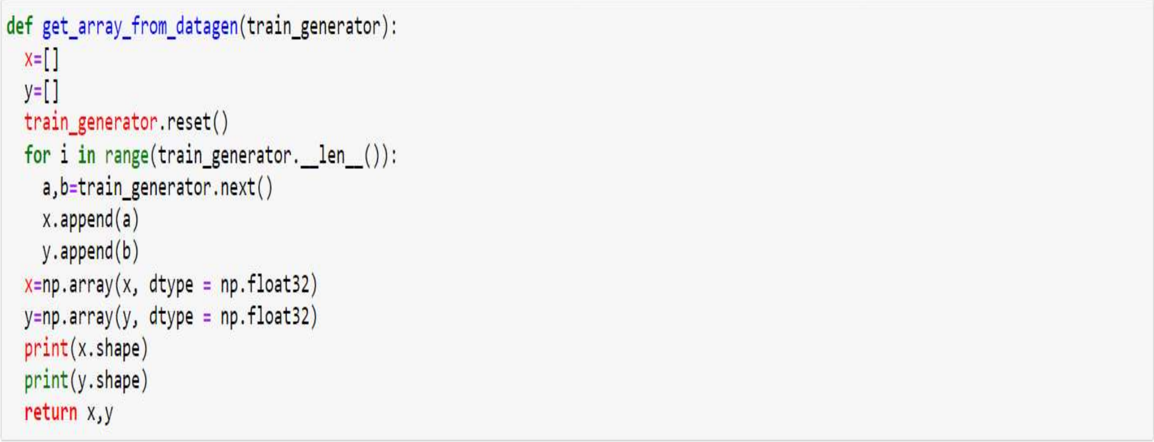
(a)

(b)

Fig 4.5:a) Correct Placement of Face Masks b) Incorrect Placement of Masks.

4.3) Preprocessing:

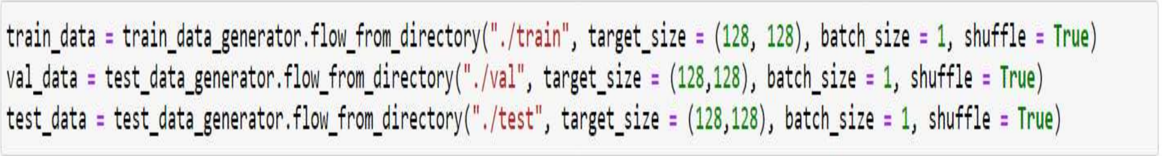
The images of the dataset are read into the memory and converted into a numpy array so that the models can process the data faster and efficiently.



```
def get_array_from_datagen(train_generator):
  x=[]
  y=[]
  train_generator.reset()
  for i in range(train_generator.__len__()):
    a,b=train_generator.next()
    x.append(a)
    y.append(b)
  x=np.array(x, dtype = np.float32)
  y=np.array(y, dtype = np.float32)
  print(x.shape)
  print(y.shape)
  return x,y
```

4.4) Data Split :

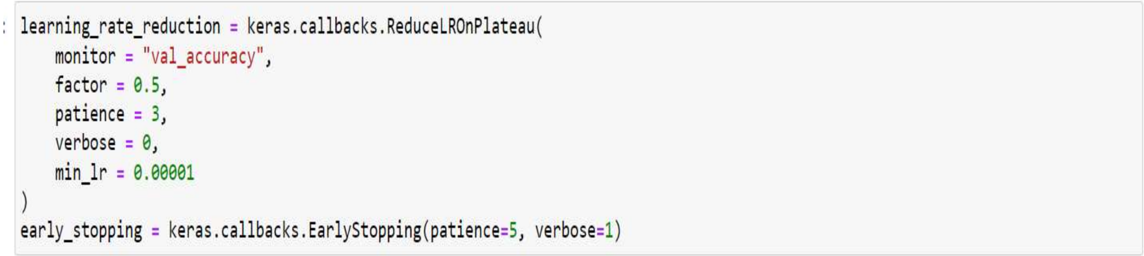
The Images are split into 3 sets namely train, validation and test data. The images are resized into 128 x 128 size.



```
train_data = train_data_generator.flow_from_directory("./train", target_size = (128, 128), batch_size = 1, shuffle = True)
val_data = test_data_generator.flow_from_directory("./val", target_size = (128,128), batch_size = 1, shuffle = True)
test_data = test_data_generator.flow_from_directory("./test", target_size = (128,128), batch_size = 1, shuffle = True)
```

4.5) Model Creation & Training:

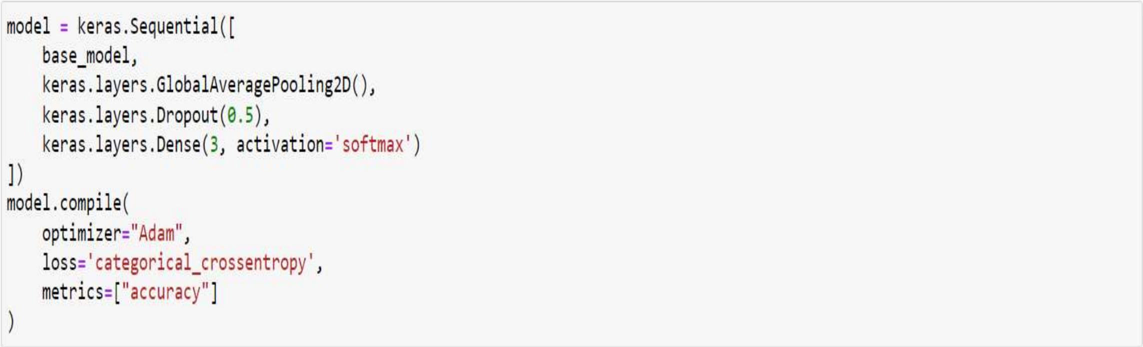
The models are trained for 10 epochs and To train the models i used some hyper parameters for training which are like below:



```
learning_rate_reduction = keras.callbacks.ReduceLROnPlateau(
    monitor = "val_accuracy",
    factor = 0.5,
    patience = 3,
    verbose = 0,
    min_lr = 0.00001
)
early_stopping = keras.callbacks.EarlyStopping(patience=5, verbose=1)
```

The early stopping mechanism is used when the model in training is not converging further after 5 epochs.

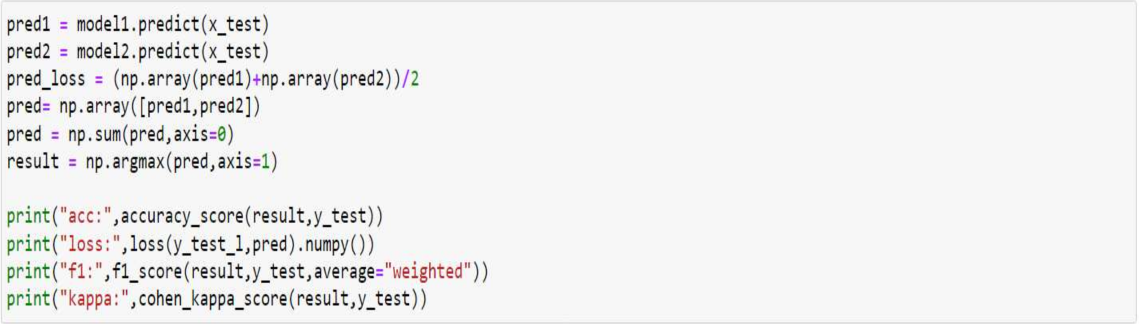
The Machine Learning Models are used as the base model in the pipeline of this model and further pooling , dropout layer are used to regularise the output from the models and finally a 3 output Dense Layer activated using Softmax is used for final prediction.



```
model = keras.Sequential([
    base_model,
    keras.layers.GlobalAveragePooling2D(),
    keras.layers.Dropout(0.5),
    keras.layers.Dense(3, activation='softmax')
])
model.compile(
    optimizer="Adam",
    loss='categorical_crossentropy',
    metrics=["accuracy"]
)
```

4.6) Testing and Calculating Metrics:

The Best Epoch Model Weights are saved and used for testing. The four different metrics are calculated for the each of the model. For Ensemble Models I use to predict from both Models of the Ensemble and combine the outputs for the final prediction of the class labels of the test data.



```
pred1 = model1.predict(x_test)
pred2 = model2.predict(x_test)
pred_loss = (np.array(pred1)+np.array(pred2))/2
pred= np.array([pred1,pred2])
pred = np.sum(pred,axis=0)
result = np.argmax(pred,axis=1)

print("acc:",accuracy_score(result,y_test))
print("loss:",loss(y_test_1,pred).numpy())
print("f1:",f1_score(result,y_test,average="weighted"))
print("kappa:",cohen_kappa_score(result,y_test))
```

CHAPTER 5: RESULT AND COMPARISON

5.1) Result:

Each machine Learning Model is trained and tested on both the datasets i.e namely Face Mask Dataset and Artificial Dataset. The Accuracy, loss, F1-Score and Kappa-Score of these models are as such:

1) Inception Model: The performance metrics for the inception model using Face Mask Dataset are Accuracy,Loss,F1-Score,Kappa-Score of 97.83, 0.062, 0.9783, 0.9675 respectively. Similarly, performance metrics for the inception model using Artificial Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 94.32, 0.181, 0.9434, 0.9073 respectively.

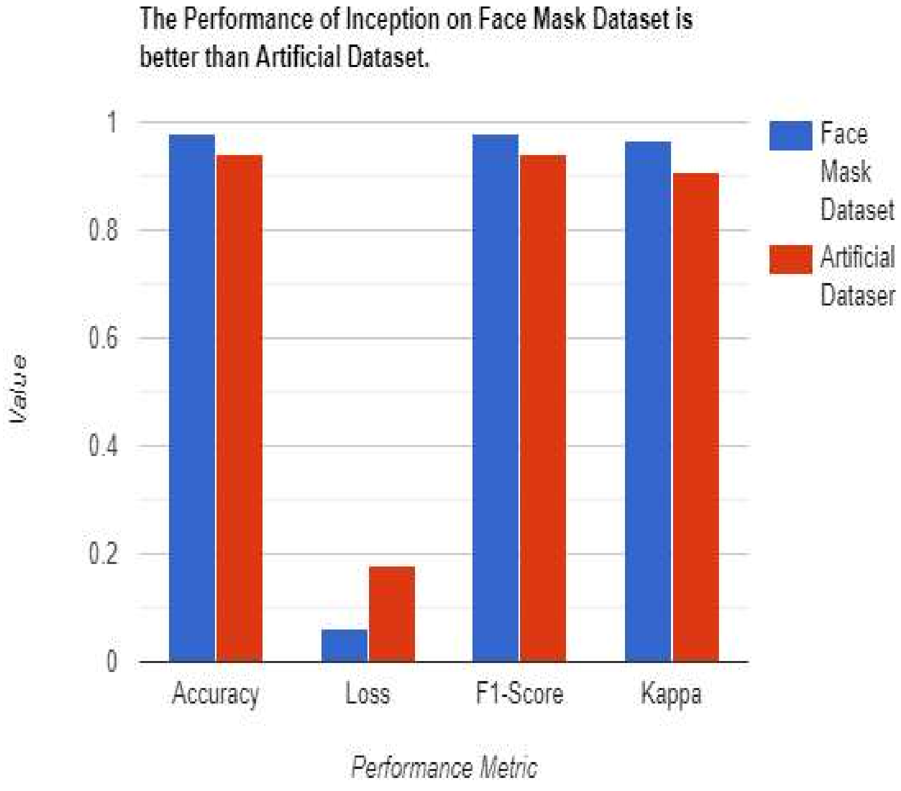


Fig 5.1:The performance Metric for InceptionModel.

2) MobileNet Model: The performance metrics for the MobileNet model using Face Mask Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 99.06, 0.038, 0.9905, 0.9858 respectively. Similarly, performance metrics for the MobileNet model

using Artificial Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 93.04, 0.276, 0.9309, 0.8866 respectively.

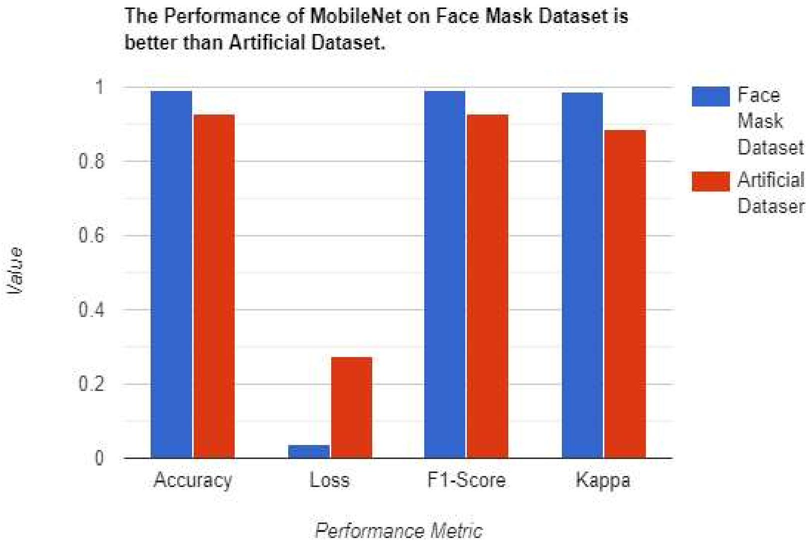


Fig 5.2:The performance Metric for MobileNetModel.

3) ResNet50 Model: The performance metrics for the ResNet50 modelusing Face Mask Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 94.94, 0.162, 0.9493, 0.9241 respectively. Similarly, performance metrics for the ResNet50 model usingArtificial Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 92.04, 0.324, 0.9252, 0.8679 respectively.

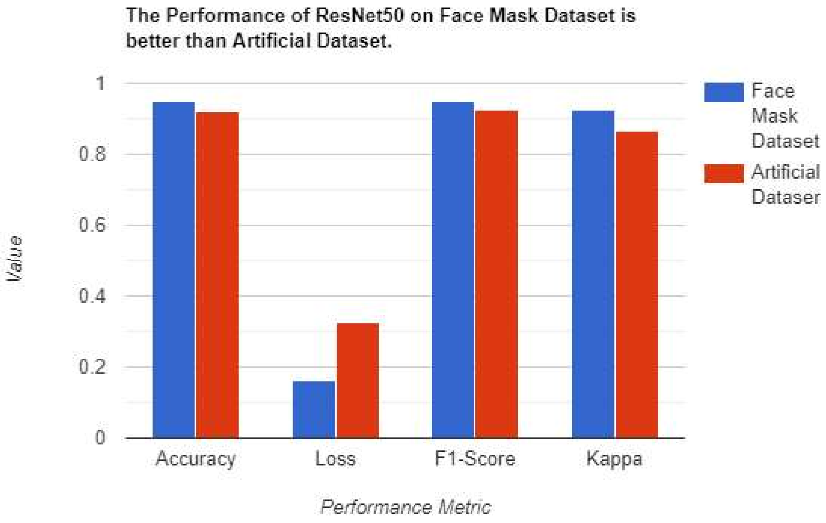


Fig 5.3:The performance Metric for ResNet50 Model.

4) EfficientNet Model: The performance metrics for the EfficientNet model using Face Mask Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 99.61, 0.024, 0.9961, 0.9941. Similarly, performance metrics for the EfficientNet model using Artificial Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 94.17, 0.22, 0.9413, 0.9057 respectively.

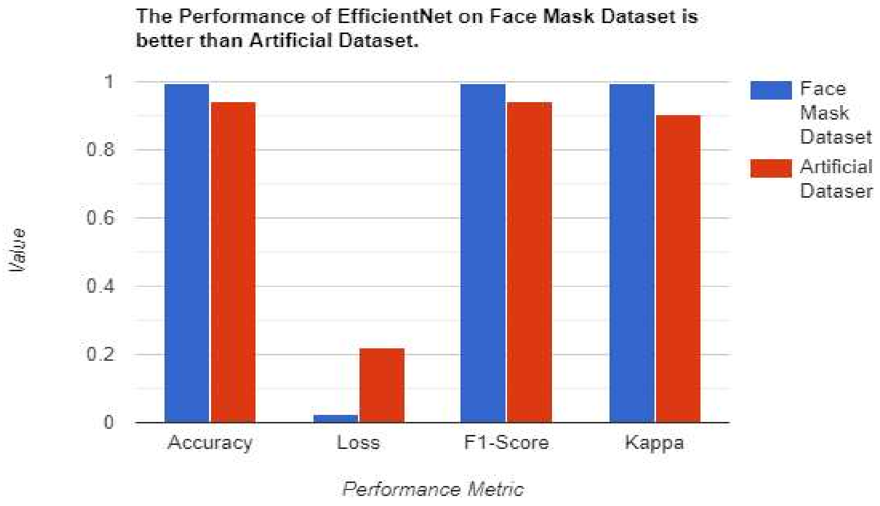


Fig 5.4:The performance Metric for EfficientNet Model.

5) DenseNet Model: The performance metrics for the DenseNet model using Face Mask Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 97.94, 0.062, 0.9794, 0.9691 respectively. Similarly, performance metrics for the DenseNet model using Artificial Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 93.32, 0.301, 0.937, 0.889 respectively.

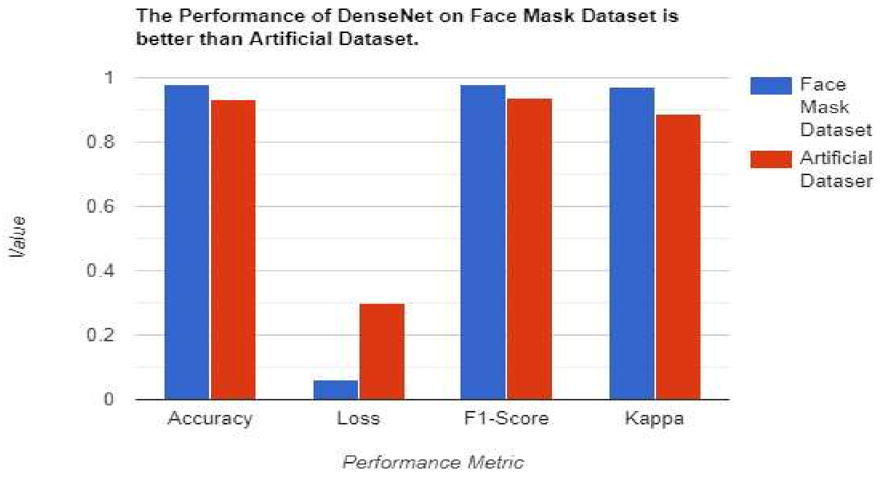


Fig 5.5:The performance Metric for DenseNet Model.

6) Xception Model: The performance metrics for the Xception model using Face Mask Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 99.02, 0.03, 0.9922, 0.9883. Similarly, performance metrics for the Xception model using Artificial Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 95.45, 0.176, 0.9548, 0.9258 respectively.

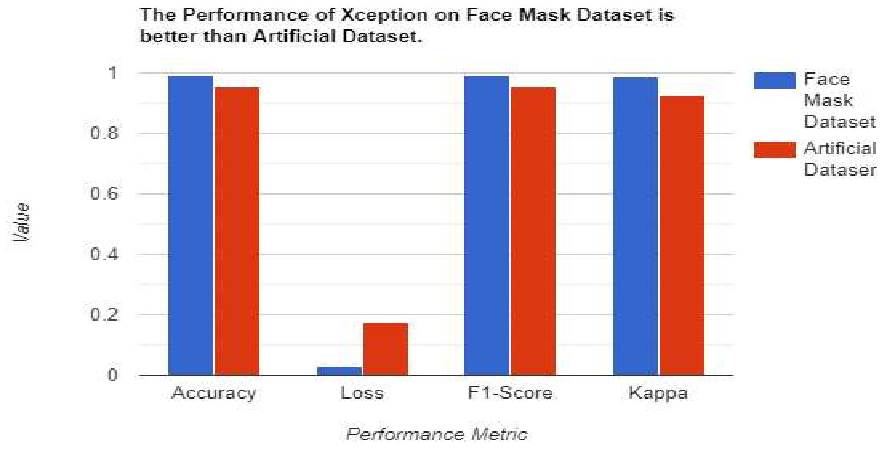


Fig 5.6:The performance Metric for Xception Model.

7) Ensemble Model (Xception + MobileNet): The performance metrics for the Ensemble model using Face Mask Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 99.44, 0.022, 0.9944, 0.9916 respectively. Similarly, performance metrics for the Ensemble model using Artificial Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 95.17, 0.115, 0.9527, 0.9209 respectively.

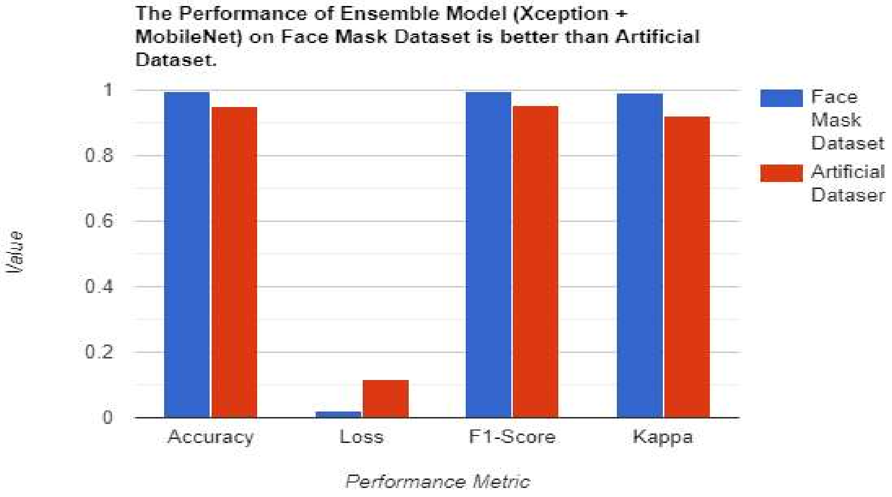


Fig 5.7:The performance Metric for Ensemble (Xception + MobileNet) Model.

8) Ensemble Model (DenseNet + MobileNet): The performance metrics for the Ensemble model using Face Mask Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 99.33, 0.035, 0.9933, 0.99 respectively. Similarly, performance metrics for the Ensemble model using Artificial Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 94.17, 0.132, 0.9457, 0.9033 respectively.

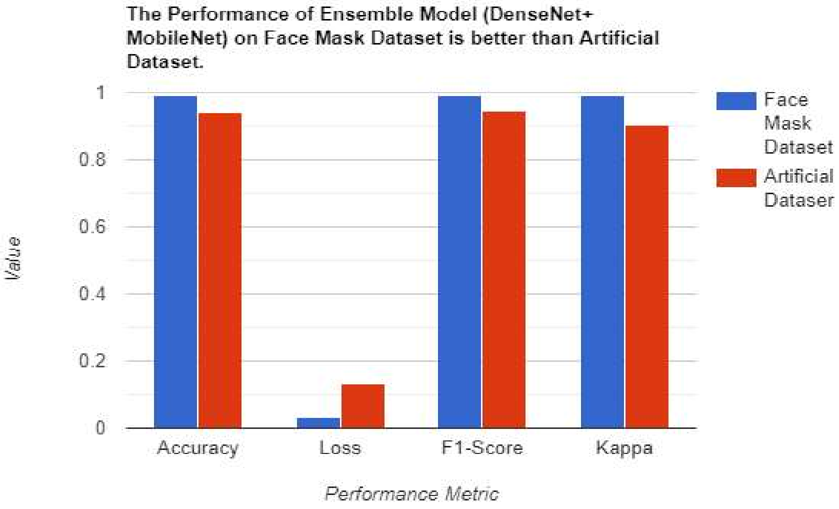


Fig 5.8:The performance Metric for Ensemble (DenseNet+ MobileNet) Model.

9) Ensemble Model (DenseNet + Xception): The performance metrics for the Ensemble model using Face Mask Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 99.5, 0.031, 0.995, 0.9925 respectively. Similarly, performance metrics for the Ensemble model using Artificial Dataset are Accuracy, Loss, F1-Score,Kappa-Score of 94.88, 0.171, 0.9509, 0.9156 respectively.

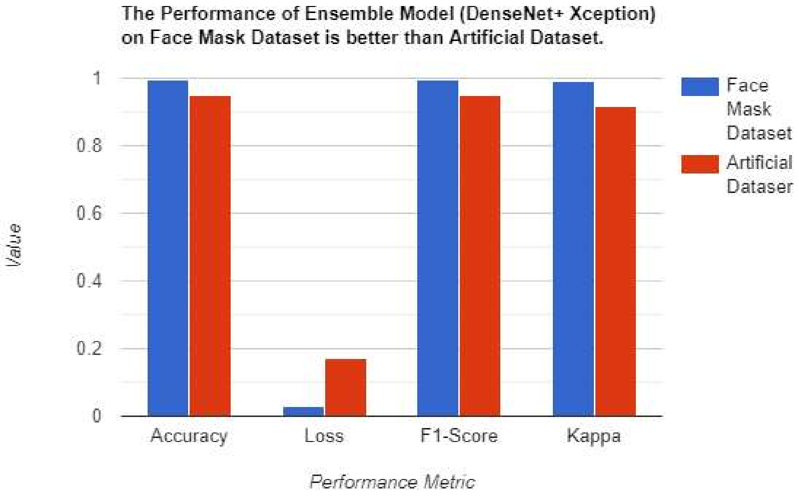


Fig 5.9:The performance Metric for Ensemble (DenseNet + Xception) Model.

10) Ensemble Model (EfficientNet + Xception): The performance metrics for the Ensemble model using Face Mask Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 99.72, 0.018, 0.9972, 0.9958 respectively. Similarly, performance metrics for the Ensemble model using Artificial Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 94.46, 0.168, 0.9478, 0.9082 respectively.

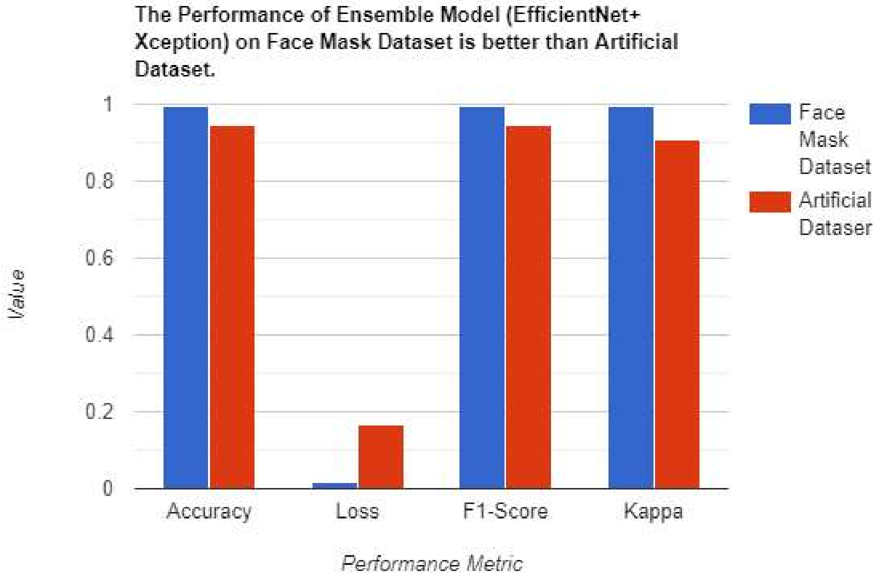


Fig 5.10:The performance Metric for Ensemble (EfficientNet + Xception) Model.

11) Ensemble Model (ResNet+ MobileNet): The performance metrics for the Ensemble model using Face Mask Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 99.27, 0.059, 0.9927, 0.9891 respectively. Similarly, performance metrics for the Ensemble model using Artificial Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 93.6, 0.157, 0.9394, 0.8943 respectively.

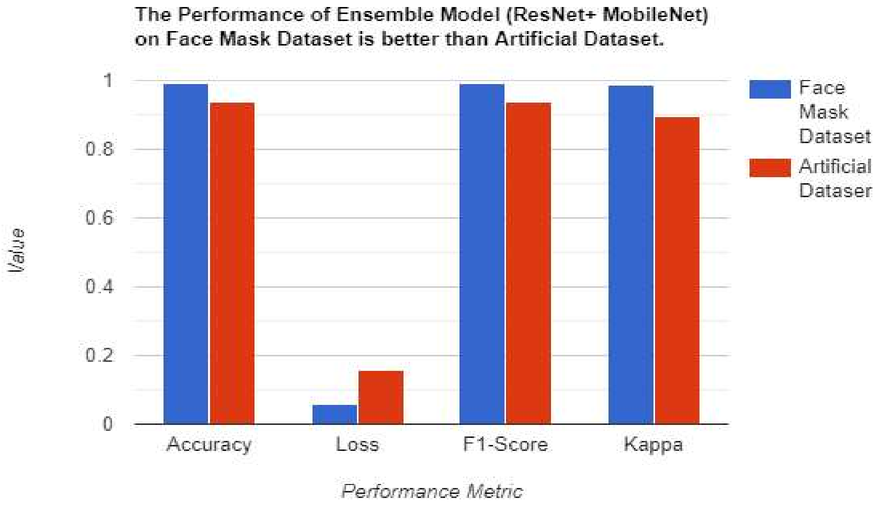


Fig 5.11:The performance Metric for Ensemble (ResNet+ MobileNet) Model.

12) Ensemble Model (ResNet + Xception): The performance metrics for the Ensemble model using Face Mask Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 99.33, 0.055, 0.9933, 0.99 respectively. Similarly, performance metrics for the Ensemble model using Artificial Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 94.74, 0.159, 0.9494, 0.9132.

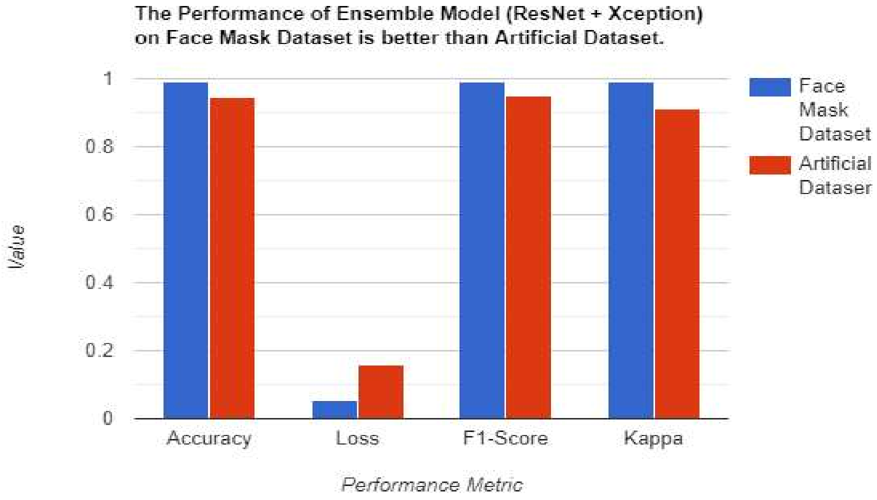


Fig 5.12:The performance Metric for Ensemble (ResNet + Xception) Model.

13) Ensemble Model (Inception + Xception): The performance metrics for the Ensemble model using Face Mask Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 99.33, 0.034, 0.9933, 0.99 respectively. Similarly, performance metrics for the Ensemble model using Artificial Dataset are Accuracy, Loss, F1-Score, Kappa-Score of 95.73, 0.122, 0.9577, 0.9304 respectively.

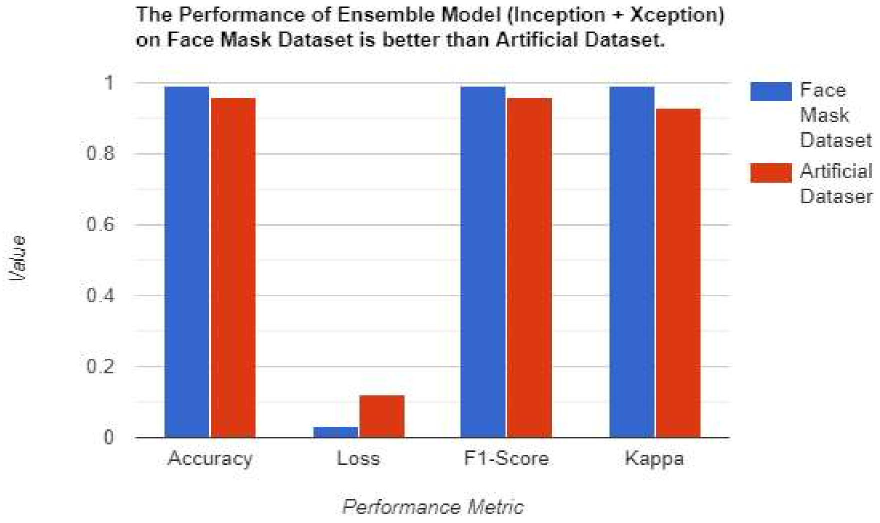


Fig 5.13:The performance Metric for Ensemble (Inception + Xception) Model.

5.2) Comparison:

Model	Accuracy	Loss	F1-Score	Kappa
ResNet [7]	94.94	0.162	0.9493	0.9241
DenseNet [8]	97.94	0.062	0.9794	0.9691
MobileNet [4]	99.06	0.038	0.9905	0.9858
EfficientNet [15]	**99.61**	**0.024**	**0.9961**	**0.9941**
Xception [5]	99.02	0.03	0.9922	0.9883
Inception [13]	97.83	0.062	0.9783	0.9675
Ens(Xception [5], MobileNet [4])	99.44	0.022	0.9944	0.9916
Ens(DenseNet [8], MobileNet [4])	99.33	0.035	0.9933	0.99
Ens(DenseNet [8], Xception [5])	99.5	0.031	0.995	0.9925
Ens(EfficientNet[15], Xception[5])	**99.72**	**0.018**	**0.9972**	**0.9958**
Ens(ResNet [7], MobileNet [4])	99.27	0.059	0.9927	0.9891
Ens(ResNet [7], Xception [5])	99.33	0.055	0.9933	0.99
Ens(Inception [13], Xception [5])	99.33	0.034	0.9933	0.99

Table 2: The Performance Metrics for Face Mask Dataset

The results of the different machine learning based convolution neural networks and ensemble of these models is presented in the Table 2 and Table 3. The Ens(Arg1, Arg2) is abbreviation is used to depict ensemble models with its arguments defining the models to be used in combination. I evaluated the models for each of the dataset i.e artificial dataset generated by us using the superimposition of masks over normal face images from CelebA-HQ Dataset and publicly available Face Mask Dataset. From analyzing the results for Face Mask Dataset which are listed in Table2, the EfficientNet Model performs the best with 99.61% accuracy, 0.024 loss, F1-Score of 0.9961 and Kappa Score of 0.9941, whereas the ensemble model of EfficeientNet and Xception perform best overall with 99.72% accuracy, 0.018 loss, F1-Score of 0.9972 and Kappa Score of 0.9958. For the artificial dataset the results are listed in Table 2, which reveals that Xception models performs best in individual models with 95.45% accuracy, 0.176 loss, F1-Score of 0.9548, and Kappa Score of 0.9258. The ensemble of Xception and Inception Model performs best than all other individual and ensemble models with

95.73% accuracy, 0.122 loss, F1-Score of 0.9577 and Kappa Score of 0.9304. The variation in the results of both datasets can be the result of different quality of images due to the fact that Face Mask Dataset is constructed from public footage from CCTV surveillance cameras. The high quality of this artificial dataset results in more feature in the input which may be the reason for lesser accuracy as compared to the Face Mask Dataset.

Model	**Accuracy**	**Loss**	**F1-Score**	**Kappa**
ResNet [7]	92.04	0.324	0.9252	0.8679
DenseNet [8]	93.32	0.301	0.937	0.889
MobileNet [4]	93.04	0.276	0.9309	0.8866
EfficientNet [15]	94.17	0.22	0.9413	0.9057
Xception [5]	**95.45**	**0.176**	**0.9548**	**0.9258**
Inception [13]	94.32	0.181	0.9434	0.9073
Ens(Xception [5],MobileNet [4])	95.17	0.115	0.9527	0.9209
Ens(DenseNet [8],MobileNet [4])	94.17	0.132	0.9457	0.9033
Ens(DenseNet [8],Xception [5])	94.88	0.171	0.9509	0.9156
Ens(EfficientNet[15],Xception[5])	94.46	0.168	0.9478	0.9082
Ens(ResNet [7],MobileNet [4])	93.6	0.157	0.9394	0.8943
Ens(ResNet [7],Xception [5])	94.74	0.159	0.9494	0.9132
Ens(Inception [13],Xception [5])	**95.73**	**0.122**	**0.9577**	**0.9304**

Table 3: The Performance metrics for Artificial Dataset.

CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1) Conclusion:

After the COVID-19 pandemic, to maintain the medical instructions in such time the governments and organization's need systems and machine learning systems will definitely perform better than manual systems. In this project I have studied one such application area where I have to identify the correct use of face masks to prevent the spread of diseases in public places. I have created a dataset for facial mask usage with high quality resolution of 256x256 pixels can be very helpful to create data where none exists, as governments cannot share the surveillance footage's due to privacy concerns , it can help in conducting future quality research. I have also presented thorough comparison and project of the different machine learning models based on different performance metrics such as accuracy, loss, F1-Score and Kappa Score [18]. The different types of convolution operations used as described in section II show the growth in the architecture of CNN [1] from the conventional operations. The residual blocks [7] have been one of the initial and very good innovation where the problem of degrading gradients has been tackled and the accuracy of models improved significantly. The obvious problem with this approach is the high amount of complexity which was tackled in development of inverted residual blocks [10]. The development of depth-wise separable convolution [5] is another great development which resulted in learning the different features of an image through less complex operation and to be used in small mobile devices without compromising on the accuracy. The inception module [13] is another efficient development which helped in realizing greater convolution operation through small convolution operations and hence the learning of the different features from the image becomes cheaper than the convolution operation. The Xception [5], Inception [13] and EfficentNet [15] models performed very highly accurately, and the ensemble of these models resulted in even better results which is indicative of the fact that all these models try to learn different features from the image.

6.2) Future Work:

The fact that images are static and in real time systems I need to analyse the video frames is one of the challenges which need to be tackled in future. The availability of video data for this field is challenging to obtains thus creating video dataset by using the similar concept of mask superimposition is also a possible task. The development of an combined model that leverages the different convolution operations in the same model is also another vertical to approach for future tasks.

REFERENCES

[1] 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.

[2] D. M. Altmann, D. C. Douek, and R. J. Boyton, “What policy makers need to know about COVID-19 protective immunity,” The Lancet, vol. 395, no. 10236, pp. 1527–1529, 2020.

[3] S. A. Bello, S. Yu, and C. Wang, “Review: Deep learning on 3D point clouds,” arXiv.org, 17-Jan-2020. [Online]. Available: https://arxiv.org/abs/2001.06280.

[4] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand,M. Andreetto, and H. Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” arXiv.org, 17-Apr- 2017. [Online]. Available: https://arxiv.org/abs/1704.04861.

[5] F. Chollet, “Xception: Deep learning with depthwise separable con- volutions,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[6] C. Cheuque, M. Querales, R. Leo´n, R. Salas, and R. Torres, “An efficient multi-level convolutional neural network approach for white blood cells classification,” Diagnostics, vol. 12, no. 2, p. 248, 2022.

[7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[8] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” 2017 IEEE Conference on Com- puter Vision and Pattern Recognition (CVPR), 2017.

[9] W. Yang, Y. Chen, C. Huang, and M. Gao, “Video-based human action recognition using spatial pyramid pooling and 3D densely Convolutional Networks,” Future Internet, vol. 10, no. 12, p. 115, 2018.

[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mo- bileNetV2: Inverted residuals and linear bottlenecks,” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

[11] S. Janarthan, S. Thuseethan, S. Rajasegarar, and J. Yearwood, “Family- based plant disease characterization using Deep Neural Networks,” 2022.

[12] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

[13] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov,

D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[15] M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for Convo- lutional Neural Networks,” PMLR, 24-May-2019. [Online]. Available: http://proceedings.mlr.press/v97/tan19a.html.

[16] B. S. Bayu Dewantara and D. Twinda Rhamadhaningrum, ”Detecting multi-pose masked face using adaptive boosting and cascade classifier”, Proc. Int. Electron. Symp. (IES), pp. 436-441, Sep. 2020.

[17] N. Petrovic and D. Kocic, ”Iot-based system for COVID-19 indoor safety monitoring”, Proc. IcETRAN, pp. 1-6, 2020.

[18] J. Cohen, “A coefficient of agreement for nominal scales,” Educational and Psychological Measurement, vol. 20, no. 1, pp. 37–46, 1960.

[19] B. Qin, D. Li, “Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19,” Sensors (Switzerland) (2020), doi:10.3390/s20185236.

[20] M. Loey, G. Manogaran, MHN. Taha, NEM. Khalifa, “A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic” , Meas J. Int. Meas. Confed. (2021), doi:10.1016/j.measurement.2020.108288.

[21] M. Inamdar, N. Mehendale, “Real-time face mask identification using facemasknet deep learning network”, SSRN Electron. J. (2020), doi:10.2139/ssrn.3663305.

[22] SK. Addagarla, G. Kalyan Chakravarthi, P. Anitha, “Real time multi-scale facial mask detection and classification using deep transfer learning techniques”, Int. J. Adv. Trends Comput. Sci. Eng. (2020), doi:10.30534/ijatcse/2020/33942020.

[23] J. Gathani, K. Shah, “Detecting masked faces using region-based convolutional neural network”, in: 2020 IEEE 15th Int. Conf. Ind. Inf. Syst., IEEE, 2020, pp. 156–161, doi:10.1109/ICIIS51140.2020.9342737.

[24] M. Loey, G. Manogaran, MHN. Taha, NEM. Khalifa, “Fighting against COVID-19: a novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection”, Sustain. Cities Soc. (2021), doi:10.1016/j.scs.2020.102600.

[25] Rao, T.S., Devi, S.A., Dileep, P. and Ram, M.S., 2020. A novel approach to detect face mask to control covid using deep learning. *European Journal of Molecular & Clinical Medicine*, *7*(6), pp.658-668.

[26] W. Bu, J. Xiao, C. Zhou, M. Yang, C. Peng, A cascade framework for masked face detection, in: 2017 IEEE Int. Conf. Cybern. Intell. Syst. CIS 2017 IEEE Conf. Robot. Autom. Mechatronics, RAM 2017 - Proc., 2017, doi:10.1109/ICCIS.2017.8274819.

[27] Fan, X. and Jiang, M., 2021, October. RetinaFaceMask: A single stage face mask detector for assisting control of the COVID-19 pandemic. In *2021 IEEE international conference on systems, man, and cybernetics (SMC)* (pp. 832-837). IEEE.

[28] MR. Bhuiyan, SA. Khushbu, MS. Islam, A deep learning based assistive system to classify COVID-19 face mask for human safety with YOLOv3, 2020 11th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2020, 2020, doi:10.1109/ICCCNT49239.2020.9225384.

[29] A. Chavda, J. Dsouza, S. Badgujar, A. Damani, Multi-stage CNN architecture for face mask detection. ArXiv 2020.

[30] Taneja, A. Nayyar, Vividha, P. Nagrath, in: Face Mask Detection Using Deep Learning During COVID-19, Springer, Singapore, 2021, pp. 39–51, doi:10.1007/978-981-16-0733-2_3.

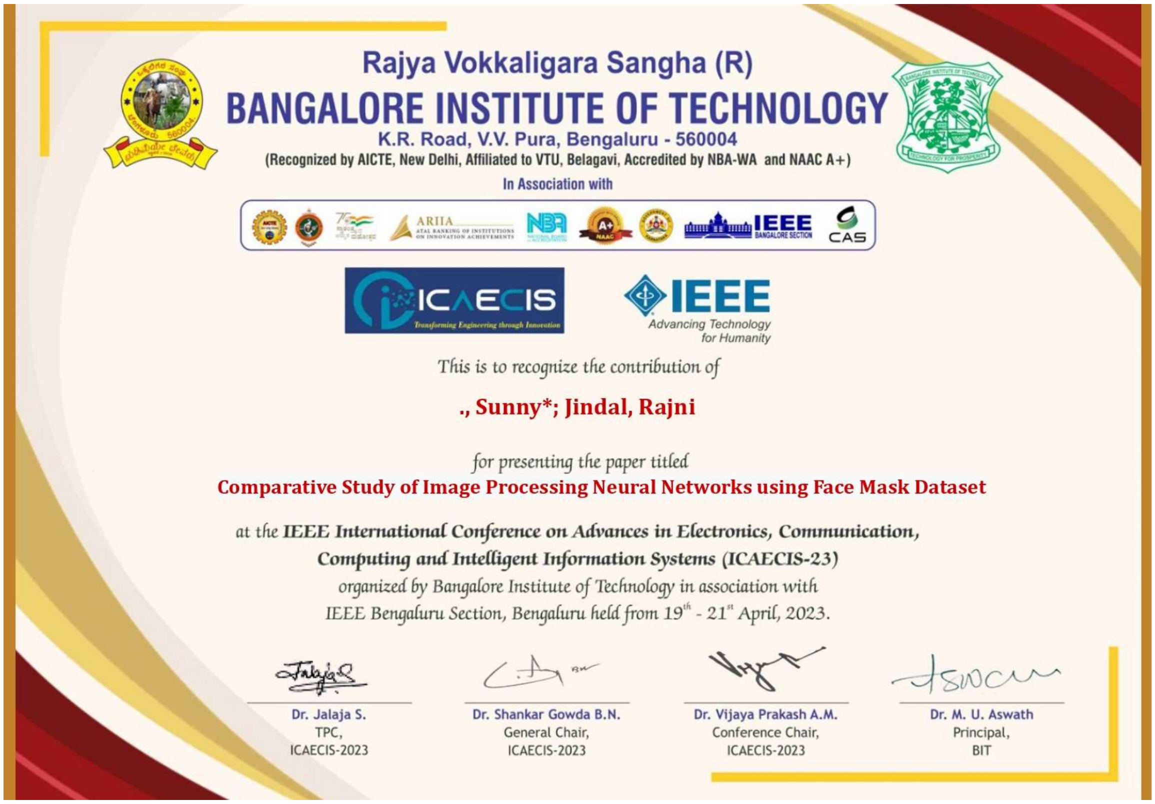
[31] V. Kumar, “Face mask detection,” Kaggle, 19-May-2021. [Online]. Available: https://www.kaggle.com/datasets/vijaykumar1799/face-mask- detection.

[32] Prajna Bhandary, ”Observations-Data Generator,” GitHub. [Online]. Available: https://github.com/prajnasb/observations

LIST OF PUBLICATIONS

[1] Sunny, R. Jindal , “Comparative Study of Image Processing Neural Networks using Face Mask Dataset” presented at IEEE International Conference on Advances in Electronics, Communication Information, and Intelligent Systems. (ICAECIS-2023)

[2] Sunny, R.Jindal , “Evaluating Performance of Convolution Neural Networks and Their Ensembles for Face Mask Classification ” presented at International Conference on Computational Intelligence and Sustainable Engineering (CISES-2023) .



Rajya Vokkaligara Sangha (R)

BANGALORE INSTITUTE OF TECHNOLOGY

K.R. Road, V.V. Pura, Bengaluru - 560004

(Recognized by AICTE, New Delhi, Affiliated to VTU, Belagavi, Accredited by NBA-WA and NAAC A+)

In Association with

ICAECIS

Transforming Engineering through Innovation

IEEE

Advancing Technology for Humanity

This is to recognize the contribution of

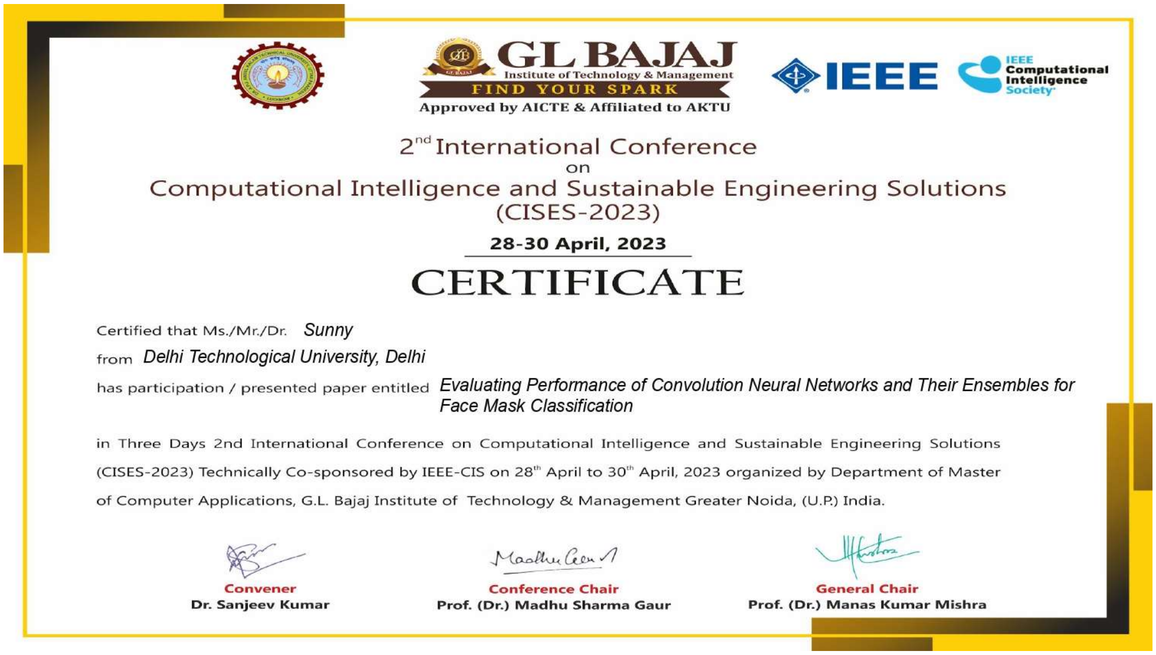
., Sunny*; Jindal, Rajni

for presenting the paper titled

Comparative Study of Image Processing Neural Networks using Face Mask Dataset

at the **IEEE International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems** (ICAECIS-23) organized by Bangalore Institute of Technology in association with IEEE Bengaluru Section, Bengaluru held from 19th - 21st April, 2023.

Dr. Jalaja S. TPC, ICAECIS-2023	Dr. Shankar Gowda B.N. General Chair, ICAECIS-2023	Dr. Vijaya Prakash A.M. Conference Chair, ICAECIS-2023	Dr. M. U. Aswath Principal, BIT



GL BAJAJ
Institute of Technology & Management
FIND YOUR SPARK
Approved by AICTE & Affiliated to AKTU

IEEE

IEEE Computational Intelligence Society

2nd International Conference
on
Computational Intelligence and Sustainable Engineering Solutions (CISES-2023)

28-30 April, 2023

CERTIFICATE

Certified that Ms./Mr./Dr. *Sunny*

from *Delhi Technological University, Delhi*

has participation / presented paper entitled *Evaluating Performance of Convolution Neural Networks and Their Ensembles for Face Mask Classification*

in Three Days 2nd International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES-2023) Technically Co-sponsored by IEEE-CIS on 28th April to 30th April, 2023 organized by Department of Master of Computer Applications, G.L. Bajaj Institute of Technology & Management Greater Noida, (U.P.) India.

	Madhu Gaur	
Convener	**Conference Chair**	**General Chair**
Dr. Sanjeev Kumar	**Prof. (Dr.) Madhu Sharma Gaur**	**Prof. (Dr.) Manas Kumar Mishra**